

# A REDUCED-COST MULTI-OBJECTIVE OPTIMIZATION METHOD BASED ON THE PARETO FRONT TECHNIQUE, NEURAL NETWORKS AND PVM

A.P. Giotis\*, K.C. Giannakoglou\* and J. Périaux†

\* National Technical University of Athens, P.O. Box 64069, Athens 157 10, GREECE  
Tel: (30)-1-772.16.36, Fax: (30)-1-772.16.58, kgianna@central.ntua.gr

† Dassault Aviation Direction de la Prospective, 78, quai Marcel Dassault, F92214  
Saint-Cloud, FRANCE, periaux@rascasse.inria.fr

**Key words:** Genetic Algorithms, Radial Basis Function Networks, PVM, Multi-objective optimization, Pareto front

**Abstract.** *Genetic Algorithms constitute a robust tool in aerodynamic shape optimization problems with one or more objectives. However, a known drawback is that they often require a large amount of evaluations based on time-consuming CFD analysis tools. The scope of this paper is to introduce the use of Radial Basis Function networks for the approximate pre-evaluation of candidate solutions in each generation. These networks are “locally” trained on a small number of previously examined solutions which are closer to each new solution; by building small-sized local networks, we avoid huge networks with time-consuming trainings. The Radial Basis Function networks are appropriate for multi-objective optimization problems, as will be explained in the text. Their effectiveness increases by assigning importance factors to each one of the design variables, so as to eliminate much of the noise that the less important design variables may cause during the networks’ training and use. The concurrent evaluation of a number of individuals on networked workstations, using the PVM message-passing model, is discussed in detail.*

---

## 1 INTRODUCTION - THE BASIC OPTIMIZATION TOOLS

CFD-aided optimization and inverse-design methods have gained particular attention in aeronautics. Without resorting to costly experiments, aerodynamic shapes can be designed or improved through either deterministic or stochastic approaches. Here, we stick with the latter and, more precisely, with Genetic Algorithms (GAs, [1]). Despite their robustness, genetic optimizers require a large number of evaluations. The evaluation phase becomes the time-consuming part of the relevant software, especially if the CFD-based evaluation uses sophisticated analysis tools. Parallelization and “cheap” knowledge-based pre-evaluations are tools that the authors have already used to accelerate GAs, in single- and multi- objective optimization problems. The aim of this paper is to extend the shape optimization tools presented by the same authors in [2] and [3] to a full-featured tool.

In [2] and [3], Artificial Neural Networks (ANNs) [4] are trained using “previous-seen” solution and used to pre-evaluate candidate solutions in each new generation of the GA, marking out a subset that “merits” exact re-evaluation. In this paper, the use of Radial Basis Function (RBF) networks, instead of the multi-layer perceptrons used in [2] and [3], is proposed for the aforesaid screening. Additionally, the proposed use of “local” networks is proposed, trained “locally” on the neighbours of the new solution in the database of previously examined ones seems to outperform the predictive capabilities of a single network for the entire search space. The advantages of using RBF networks in multi-objective optimization are discussed.

Either in single- or multi-objective optimizations, the concurrent evaluation of individuals allows the reduction of the elapsed time. This is achieved using a network of workstations and the PVM message-passing model. Some interesting parallelization aspects are discussed in a subsequent section.

For more information about genetic optimization the reader should refer to classical textbooks [1], [5]. In [2] and [3], the parameterization of aerodynamic shapes through Bezier curves (for the suction and pressure sides of an airfoil) along with circular arcs (for the leading and the trailing edges, if necessary) is presented and will not be repeated here. The ANN-based pre-evaluation will proved to be more efficient if “importance factors” are assigned to the design parameters and this will be demonstrated to this paper. The multi-objective optimization is based on the non-dominated sorting technique which provides a number of Pareto optimal solutions on the so-called Pareto front [6]. We shall, therefore, refrain from repeating here the details of the basic optimization tools as these could be found in the cited textbooks and papers. Only aspects related to the parallelization and the implementation of ANNs in the Pareto front technique will be discussed.

Apart from the aerodynamic optimization problem (modeled through either a time-marching Navier-Stokes method for unstructured grids [7] or a “cheap” surface source singularity method [8]), a numerical problem is also analyzed to support our conclusions.

## 2 CONCURRENT EVALUATION OF CANDIDATE SOLUTIONS

In optimization or inverse design problems relying upon CFD tools the computing cost per individual depends on the flow analysis tool used. Dealing with inviscid or viscous high-speed flows, this cost is high enough (since the Euler or Navier-Stokes equations need to be solved) and includes also the cost for meshing, pre- and post-processing. In order to reduce the elapsed time, the concurrent evaluations of the candidate solutions are used since the GA operates on a population of individuals which can be evaluated separately, within each generation.

The parallel implementation of the proposed optimizer uses the Parallel Virtual Machine (PVM) [9] software package. PVM allows a heterogeneous collection of workstations and/or supercomputers to function as a single parallel machine. It is portable, it runs on most modern platforms and is available in the form of source code at no cost.

The parallel optimizer is based on the master-slave model. All of the genetic operations, like parent selection and recombination are executed serially to keep the optimizer simple, without though damaging the final speed-up. The master process, which is started by the user, spawns as many slave processes as the number of available processors (one machine may have more than one processors) and maintains a queue of the currently available processing units and a table with the current evaluation assignments. The GA dispatches a set of design variables and an appropriate message identification to each slave process, for evaluation purposes. The slave returns one fitness value per objective. In general, the available processors are less than the population size and the evaluation requests are continuous during the evaluation phase. As soon as a processor becomes available, the next evaluation task is assigned to it. Note that the time required to complete each task may vary depending on the processor load, its power (in a heterogeneous environment) and some unforeseen circumstances. For instance, in a shape optimization problem an evaluation task may fail during the formation of the contour using the Bezier points (eg. crossing sides) or during the CFD evaluation. Even if a task does not fail, the evaluation times may differ if the convergence criterion depends on the final residual. At the end of each generation, all evaluation tasks must be completed before proceeding to the next generation, thus the master process should wait for each one of the slaves to return the fitness scores. Consequently, with a fixed number of processors, the overall speedup degrades as the population size decreases,

The communication cost is almost negligible since only the design variables are sent to the processors and one score per objective is returned to the master.

Each slave process saves the design variables in a file and runs a script which executes a series of programs that form the shape, define the computational domain, generate an unstructured mesh, call the Navier-Stokes flow solver and compute fitness(es) through post processing. Disadvantages of using script files is that copies of all executable files called should exist on each machine and that successive programs communicate by saving to and reading from the hard disk. However, this increases flexibility. Also, it is not an

easy task to incorporate different stand-alone programs in the same executable, even if source codes are available. The communication overhead between programs, activated by the script file, is small compared to the optimization cost as a whole (less than 0.05% in designs based on viscous flow CFD tools).

The GAs based optimizer is programmed in C++ whereas the CFD tools have been written in FORTRAN77.

### 3 RBF NETWORKS AND THEIR TRAINING

A RBF [4],[10] network maps the input space, i.e. the space of the  $M$  design variables onto the output space of objectives. Even for the multi-objective optimization problems, it is assumed here, that a single output unit is used, so the RBF mapping reads  $\mathcal{R}^M \rightarrow \mathcal{R}^1$ .

RBF networks involve three layers of processing units. The intermediate or hidden layer, between the  $M$ -unit inlet and the single output layers, consists of  $N$  units, the so-called RBF centers, as in fig. 1. Hidden units are associated with an equal number

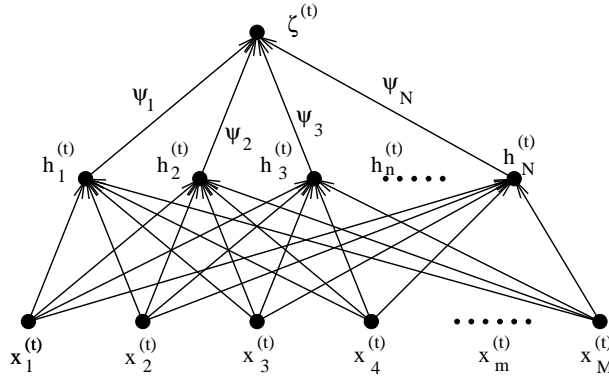


Figure 1: A typical RBF network with a single output unit

of arrays, with  $M$  components each, which will be denoted by  $\vec{c}^{(n)}$ ,  $n = 1, N$ . In a standard RBF network, which is presented with  $T$  paired input-output arrays ( $\vec{x}^{(t)}$  and  $\vec{y}^{(t)}$ , respectively,  $t = 1, T$ ), we select  $N = T$  hidden units as follows

$$c_m^{(t)} = x_m^{(t)}, \quad t = 1, T, \quad m = 1, M$$

Then, the  $n^{th}$  hidden unit value  $h_n^{(t)}$  which corresponds to  $\vec{x}^{(t)}$  is the outcome of a nonlinear filter, namely

$$h_n^{(t)} = \Phi \left( \left\| \vec{x}^{(t)} - \vec{c}^{(n)} \right\|_2, r_n \right) \quad (1)$$

Among various alternative expressions for the activation function [4], we have selected  $\Phi(u, r) = \exp(-u/r)$ , which performs an  $\mathcal{R}^+ \rightarrow \mathcal{R}$  mapping. Without loss of generality, we may assume that  $r_n = 1, n = 1, N$ . For the same input  $\vec{x}^{(t)}$ , the resulting output value  $\zeta^{(t)}$  is a linear function of all hidden unit values  $h_n^{(t)}$

$$\zeta^{(t)} = \psi_n h_n^{(t)} \quad (2)$$

where summation applies to the repeated index  $n$ . The weights  $\psi_n$  are to be calculated during the training phase, the criterion being the minimization of the norm  $E = \sum_{t=1}^T (\zeta^{(t)} - y^{(t)})^2$ , after presenting the network with the ensemble of available training patterns. Contrary to other multi-layer feedforward networks, the mapping of the hidden to the output space is linear, eq. 2, and consequently the computation of the  $N = T$  coefficients  $\psi_n$ , i.e. of the array  $\vec{\Psi}$  components, is the solution of the linear system

$$H\vec{\Psi} = \vec{Z} \tag{3}$$

The modified Gram-Schmidt method is used for the inversion of  $H$ . Note that, in multi-objective optimization problems, where more than one  $y_k^{(t)}, k = 1, K$  output units exist, the same training set leads to the same matrix  $H$ . The inversion of  $H$  is valid for any output other than the first and it can be safely stated that the cost for training the RBF network is approximately the same regardless the number of output units. This is the main advantage of the use of direct inversion compared to any iterative back-error propagation training method. In the latter, the training cost increases linearly with the number of outputs, i.e. the number of objectives.

In the form presented so far, a RBF network provides guesses for the objective values of any new candidate solution (provided that a training set which is representative of the search space is available) with affordable training cost. To ensure the latter, the size  $T$  is always kept low and the training is carried out through a small subset of the available database. This subset consists of the ‘‘neighbours’’ of the new individual in the solution space and is proved to be advantageous in terms of quality of predictions, over and above to the gain in CPU cost. However, the predictive capabilities of a RBF network can be further improved by considering a ‘‘weighted’’ norm in eq. 1. The modification proposed herein is based on the fact that, in an aerodynamic optimization problem, the involved design parameters are not all of equal importance. For instance, the leading edge circle radius is of primary importance since it affects considerably the pressure distribution further downstream. In this respect, user-defined importance factors  $I_m, m = 1, M$  are associated with each design variable. These are introduced in eq. 1, where the weighted norm

$$\|\vec{x}^{(t)} - \vec{c}^{(t)}\|_{wei} = \sum_{m=1}^M I_m (x_m^{(t)} - c_m^{(n)})^2 \tag{4}$$

is used instead of norm-2.

It is recommended that the user defines a small number (2–4) of classes of ‘‘importance’’ for the design variables. By doing so, the non-dimensionalization of eq. 1 requires that  $r_n = \sum_{m=1}^M I_m$ , at least for the activation function employed herein.

#### 4 GAs WITH APPROXIMATE PRE-EVALUATION

The main steps of the GA-based single- and multi-objective optimization method using approximate fitness pre-evaluations through RBF networks, are listed below:

#### 4.1 Single-Objective algorithm

**Phase 1:** The starting population evolves for a few generations, by applying genetic operations. To compute fitness scores, calls to the evaluation routine are used and entries are put into the database for subsequent trainings of the RBF network.

**Phase 2:** During the next generations, only a percentage of the candidate solutions is exactly evaluated. The entire population is first pre-evaluated using a local RBF network trained separately for each individual on the  $T$  closest entries in the database. The approximate fitness scores are used to sort out the actual individuals. The number of individuals to be exactly re-evaluated routine is the maximum of a predefined percentage of the population ( $\sigma L = L_\sigma, 0 < \sigma < 1$ , starting from the best one) and the number of individuals with higher predicted score than the best solution so far.

Thus, for each generation, the gain in computing time is  $(1 - \sigma)L$  evaluation routine calls. Of course, the  $L_\sigma$  new exact evaluations are used to enrich the database. The  $\sigma$ -parameter is the outcome of a parametric study that follows in the results section.

#### 4.2 Multi-Objective algorithm

**Phase 1:** The starting population evolves for a few generations, using non-dominated sorting and sharing [6]. At the end of each generation a new Pareto front is obtained, which consists of the nondominated (or Pareto-optimal) solutions, in the sense that none of them is absolutely superior to any other.

The reproduction task is based on dummy fitness values, calculated after sharing is separately applied along each front to spread the solutions all over the front. The sharing factor (with which the real score should be multiplied to give the dummy one)

$$m_i = \sum_{j=1}^{N_{pop}} \max \left( 0, 1 - \frac{d(i, j)}{\sigma_{share}} \right) \tag{5}$$

is computed for its  $i^{th}$  member where  $0 < \sigma_{share} < 1$ ,  $(i, j)$  lie along the same front and  $d(i, j)$  is their Euclidean distance.

After the end of the first generation the Pareto front is stored and used in successive generations for non-dominance checking and sharing. In each generation, this Pareto front is updated and generally it consists primarily of members found in previous generations and a few new entries. If an individual of the current generation belongs to the Pareto front, sharing includes also the stored Pareto front members .

**Phase 2:** In subsequent generations, the population is first pre-evaluated using trained RBF networks. The current population is sorted out in temporary fronts by also considering the actual Pareto front members. The number of individuals that merit

exact re-evaluation is the maximum of a predefined percentage of them ( $L_\sigma$ , starting from the Pareto front) and the population temporary Pareto front. Only accurately evaluated individuals are allowed to enter the actual Pareto front. Note that during the calculation of the dummy fitness values, exact and approximate scores are mixed up.

## 5 RESULTS AND DISCUSSION

Numerical and low-level CFD optimization problems which rely upon a “cheap” flow-analysis routine have been used for demonstration purposes and for some reduced CPU-cost parametric studies, before proceeding to the applications using the time-consuming Navier-Stokes routine.

### 5.1 Numerical optimization

In the first test problem, the Rastrigin function

$$f_{RA}(\vec{x}) = 10M + \sum_{m=1}^M (x_m^2 - 10\cos(2\pi x_m)) \quad (6)$$

with  $M = 20$  and  $-5.12 \leq x_m \leq 5.12$ ,  $m = 1, M$  should be minimized. This is a single-objective optimization problem, the sought for solution is  $x_m = 0$ , and we are using it in order to demonstrate the convergence acceleration of GAs with approximate pre-evaluation. Needless to say that in this case, the gain of the GAs-ANN algorithm is illusive since, it costs less to evaluate the function than to find the proper training set, train the ANN and then approximate  $f_{RA}$ .

Fig. 2 compares the convergence rate of the standard GA (with 100% exact evaluations in each generation) with its counterpart using ANN approximate pre-evaluations and only  $\sigma L$  exact evaluations per generation ( $\sigma=0.05, 0.10$  and  $0.20$ ). The best score evolution is plotted in terms of the number of exact evaluations. Fig. 2 shows the superiority of the proposed method compared to standard GAs and the important role of  $\sigma$ . An appropriate choice of  $\sigma$  may considerably improve the convergence rate. We can safely state that similar conclusions can be drawn for multi-objective optimization problems.

It is important to note that even if smaller values of  $\sigma$  seem to improve the convergence rate, a minimum value (minimum number of exact evaluations) is required since the GA may be driven to regions where insufficient information (scarce data) may exist.

### 5.2 Parametric studies

In this section, we are presenting studies that have been undertaken in order to demonstrate the role of several parameters involved in the present method. We have selected a low-level flow model and the corresponding non-time-consuming software, in order to increase the number of tests we had the possibility to carry out. Thus, we have defined and used two inverse-design exercises, namely the reconstruction of the NACA 0012 (the

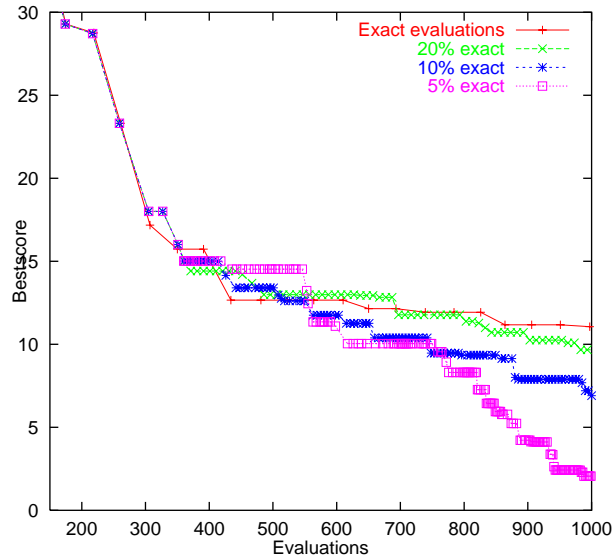


Figure 2: Convergence rate of the Rastrigin test function method, with and without ANN.

target being the pressure distribution at zero incidence) and the NACA 4412 (examined at 10 deg. incidence) profiles. In both cases, a panel method [8], for incompressible potential flows, was used. Both exercises used a single-objective, but the conclusions can be generalized to multi-objective optimization problems as well.

The first point to be investigated is the appropriate training set size for the RBF network. It is evident that small training sets (small values of  $T$ ) are preferred, since the corresponding training CPU-cost is low. Concerning predictive capabilities, the user should select  $T$  so as to bring about a reconciliation between the two extreme symptoms, namely insufficient learning (for very small  $T$ 's) and increased generalization of the network (for high  $T$  values; the cost to pay for the generalization is that faraway database entries may erroneously affect the network prediction). Since the value of  $T$  is closely related to the percentage ( $\sigma$ ) of the population that will be re-evaluated through the direct solver, a great number of  $T$  and  $\sigma$  values have been examined. The results summarized in fig. 3 correspond to five  $T$  values ( $T = 5, 10, 15, 25$  and  $50$ ) combined with ten  $\sigma$  values ( $\sigma = 0.1, 0.2, \dots, 1.0$ ). Each point in this figure is the best average score of ten runs for the same problem and the same parameters. Basic data for the GA are tabulated below:

The close similarity of the behaviors of the groups of curves shown in fig. 3, for both design exercises, is remarkable. Within the same number (600) of direct evaluations better final scores are obtained with the lower  $\sigma$  values. There is of course a lower limit for  $\sigma$  below which the role of the RBF network dominates, the number of direct evaluations per generation reduces dramatically and the GA completely fails to provide improved solutions. The lower limit is approximately  $\sigma = 0.1$  and we therefore refrain from presenting results for smaller  $\sigma$  values. From the same figure, definite conclusions about the opti-



Population size	50
Max number of exact evaluations	600
Two-point crossover probability	85%
Mutation probability	0.5%
Binary tournament probability	85%

Table 1: GA parameters

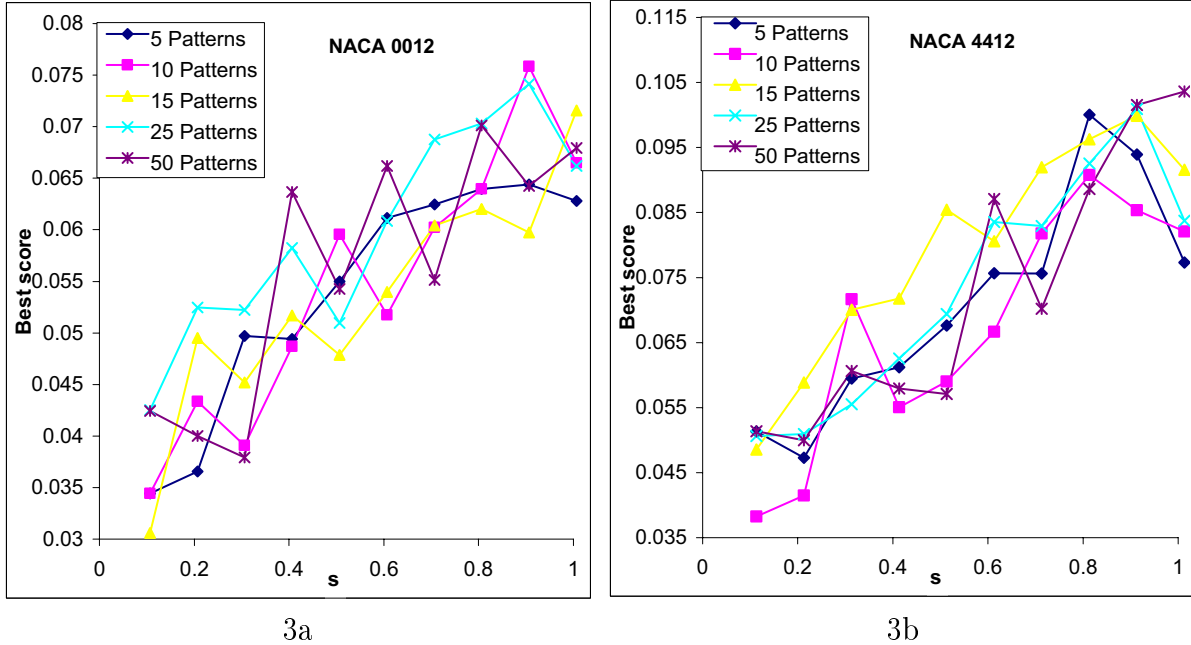


Figure 3: Re-design of NACA 0012 (left) and NACA 4412 (right) using the panel method. Average best score (minimization problem) for various  $\sigma$  and  $T$  values.

num  $T$  values cannot be drawn and seems that (considering also the stochastic nature of the model) this is not a crucial parameter. However,  $T$  values of the order of 15 (for NACA 0012) and 10 (for NACA 4412) seem to perform slightly better. Also, such small values yield low training cost per individual and should be definitely recommended. To complete this discussion, fig. 4 illustrates the predictive capabilities of the RBF network when trained with the 15 closest database entries. This figure corresponds to 50 individuals and is indicative of the errors that usually appear between RBF network guesses and exact fitness scores; it corresponds to the first exercise after the first 600 evaluations.

Another point that needs to be investigated is the use of importance factors for the design parameters. This is newly proposed in this paper and helps depreciating the role of some design parameters with “secondary” effect during the training of the RBF networks. Here, priority (highly valued importance factors  $I$ ) is given to the geometrical parameters that determine the shape of the airfoil leading edge. Fig. 5 summarizes results from five runs. In the captions, “circle priority” means that increased importance is given to the

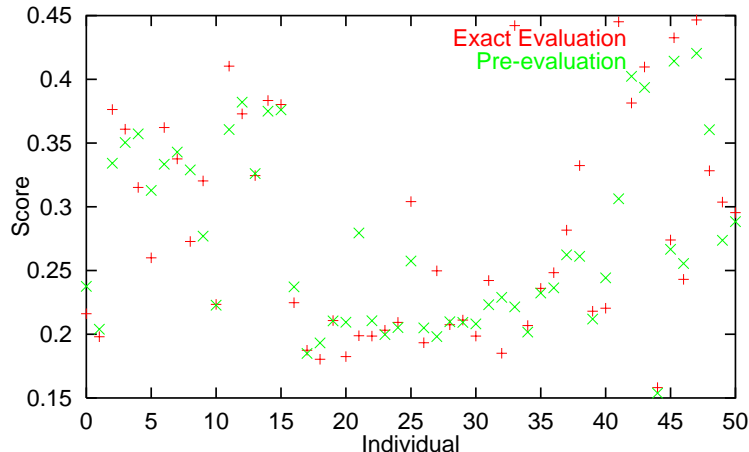


Figure 4: Redesign of the NACA 0012 profile. Comparison of RBF-based pre-evaluation and exact evaluation scores.

leading edge circle whereas in the “LE priority” the first Bezier points are also given high  $I$  values. Results with  $\sigma = 0.10$  and  $0.20$  are shown. Regardless the  $\sigma$  values, the importance factors considerably increase the performance of the genetic optimization method. They seem to be extremely useful during those generations where the database entries are really scarce; in order to train the network with local information,  $T$  should be kept low and with low  $T$  values we have to filter out the role of non-important parameters.

### 5.3 Multi-objective, viscous flow optimization

This case was first presented by the authors in [3], where a single multi-layer perceptron for the entire search space was built and used for the pre-evaluation of individuals. Here, we repeat the same study using local RBF networks that are trained separately for each new individual. Instead of the regular retraining of a huge network through an iterative method, we are relying upon the training of multiple networks of very small size and, consequently of almost negligible training cost per network.

The objective is to design a new airfoil that yields given pressure distributions at two operating points. The targets have been defined using, as “starting” profiles, a low-subsonic, high-lift airfoil ( $M_{inf} = 0.2, Re = 5 \cdot 10^6, a = 10.8^\circ$ ) and a transonic, low-drag one ( $M_{inf} = 0.77, Re = 10^7, a = 1.0^\circ$ ). Results from three computations, all of them with the same GA parameters, are presented. In the first, which was carried out without approximate pre-evaluations, 5000 exact evaluations were allowed at most. The other two runs stopped at 2000 exact evaluations. Between them, the second was without pre-evaluations whereas the third used RBF networks trained with the 10 closest neighbors of each individual and with  $\sigma = 8/50 = 0.16$ , 50 being the population size and 8 the number of processors used concurrently. The parallel system used was a cluster of Linux workstations (Intel Pentium III, 500MHz) connected on the same LAN using a fast

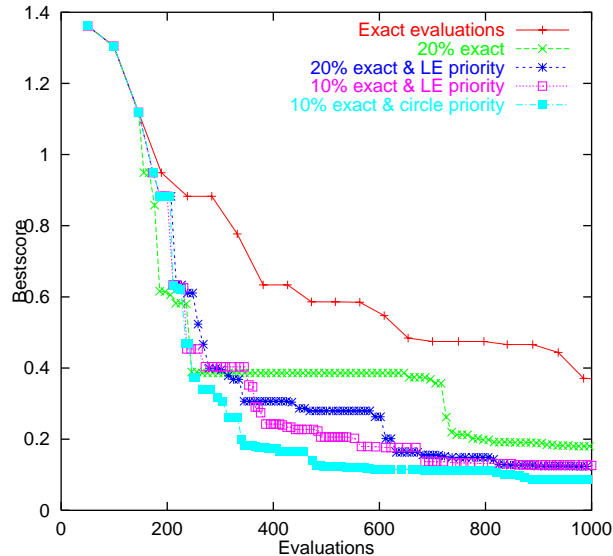


Figure 5: Re-design of the NACA 4412 profile. Convergence rates with and without ANN-based pre-evaluation and/or the use of importance factors.

Ethernet switch. The wall time required to evaluate an individual was about 4 minutes, mostly spent for two direct flow solver calls for each operating point. Unstructured grids with triangular elements were used to mesh the computational domain, with an average size of 6000 triangles / 3000 nodes. The airfoil contour was described using 200 nodes. The flow solver of [7], with the  $k - \epsilon$  model and the wall-function technique was used.

A profiling of the third run is given in fig. 6, for the first 40 generations of the GA, where the reader may find the number of exact evaluations required in each generation after the ANN-based screening. It is also noticeable that, during the first two generations, approximately 40 percent of the population failed during the CFD-routine call. This was due to the fact that the search space for each variable was large enough. Quite early (at the third generation) the number of CFD-routine fails reduces drastically thanks to the RBF network based pre-evaluation that discards most of the worse individuals.

Fig. 7 shows the computed Pareto fronts with the three aforesaid runs. The Pareto front which is the outcome of the first run should be considered as the reference front (as this costs as much as 2.5 times each one of the rests. However, as one may notice at a particular zone the third run (with pre-evaluation) gives locally better solutions. It is also interesting to compare the fronts resulted from the second and the third runs, as they both have similar CPU-costs. The ANN-aided run seems to provide a front that is less spread out than the second run, which was exclusively based on exact evaluations. However, the third front gives a higher number of entries in the “middle”, in a region which is practically of higher importance. We recall that only exactly evaluated individuals are allowed to join the Pareto front and that sharing (the spreading mechanism) applies separately along

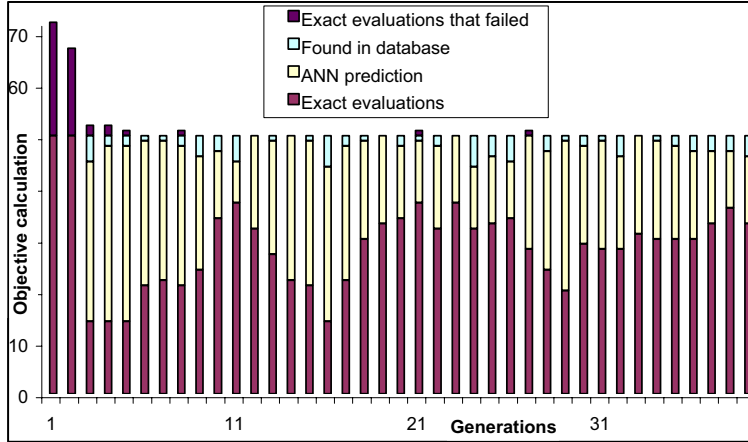


Figure 6: Two operating point design: Profiling of the third run.

each front. This explains clearly why the third front entries are in the “middle” of the reference front.

## 6 CONCLUSIONS

It is well known that Genetic Algorithms are an effective and robust tool for handling inverse-design and optimization problems in aeronautics. In this paper, two techniques are combined with GAs for acceleration purposes. These are: (a) the concurrent evaluation of multiple candidate solutions using networked workstations and (b) the use of a pre-evaluation phase in each generation, as a first screening that isolates the subset of the population that merits exact evaluation.

The pre-evaluation was based on Artificial Neural Networks and, particularly, on Radial Basis Function ones. They were trained on the previous seen solutions and then used to approximately evaluate individuals. Main conclusions are:

- (a) The pre-evaluation phase considerably reduces the overall cost of the optimization tool.
- (b) RBF networks trained in the “local” sense should be preferably used instead of a single huge network for the entire search space. The direct method used for their training bears almost negligible cost due to the small size of the training set. RBF networks should be preferred in multi-objective optimizations as the extra objectives does not practically increase the training set.
- (c) As introduced in this paper, it is recommended to use different importance factors for the various design parameters; their use improves the predictive capability of RBF networks, by artificially getting rid of the less important design parameters during its training with a small-sized training set.

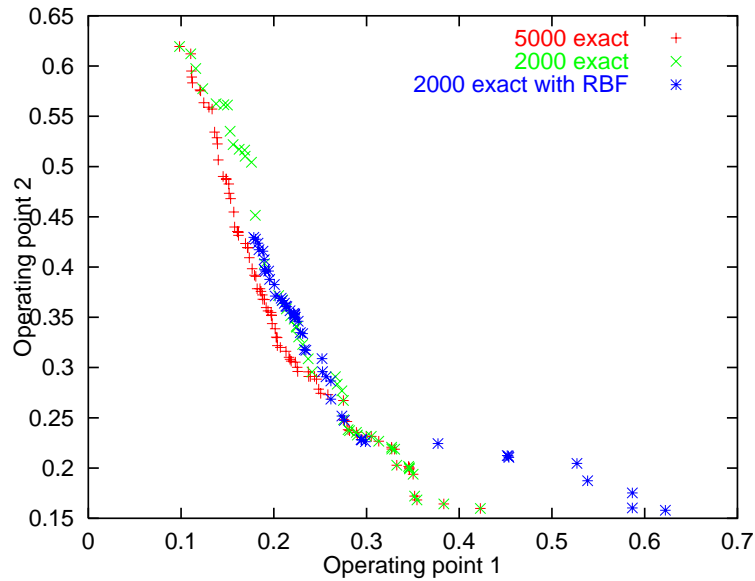


Figure 7: Two-operating point design: Pareto fronts computed with and without approximate pre-evaluation.

**Acknowledgment:** The first two authors are indebted to Dassault Aviation for funding this research.

## References

- [1] D.E. Goldberg. *Genetic Algorithms in search, optimization & machine learning*. Addison-Wesley, 1989.
- [2] K.C. Giannakoglou. Designing turbomachinery blades using evolutionary methods. ASME Paper 99-GT-181, 1999.
- [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. Cichocki and R. Unbenhauen. *Neural Networks For Optimization and Signal Processing*. John Wiley & Sons, 1993.
- [5] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin Heidelberg, 2nd edition, 1994.

- [6] J. Horn, N. Nafpliotis, and D.E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. Proceedings of the First IEEE Conference on Evolutionary Computation, 82-87, Volume I, ICEC '94.
- [7] 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.
- [8] V Dedoussis. *Calculation of Inviscid Flow through Aerofoil Cascade and between Wind Tunnel Wall*. M.Sc. Thesis, Dept. Aeronautics, Imperial College, London, 1983.
- [9] A. Geist, A. Beguelin, Jack Dongarra, W. Jiang, R. Manchek, and V Sunderam. *PVM: Parallel Virtual Machine. A user's guide and tutorial for networked Parallel Computing*. The MIT Press, 1994.
- [10] J. Hertz, A. Krogh, and Palmer R.G. *Introduction to the Theory of Neural Computation*. Addison-Wesley Publishing Company, 5th edition, 1992.