European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS 2000
Barcelona, 11-14 September 2000
©ECCOMAS

A PARALLEL INVERSE-DESIGN ALGORITHM IN AERONAUTICS BASED ON GENETIC ALGORITHMS AND THE ADJOINT METHOD

K. C. Giannakoglou[⋆], Th. I. Pappou [⋆], A. P. Giotis [⋆] and D. G. Koubogiannis[⋆]

[⋆] National Technical University of Athens (NTUA) P.O. Box 64069, 15710, Athens, Greece e-mail: kgianna@central.ntua.gr

Key words: Inverse design, optimization methods, genetic algorithms, adjoint method, parallel CFD.

Abstract. *This paper presents a hybrid (Genetic Algorithms, GA and Adjoint Method, AM) tool for the optimum design of aerodynamic shapes. CFD techniques for unstructured grids are used during both optimization phases. The best solution computed after a few generations through the GA costitutes a good starting shape for the second, AM-based optimization phase. This is enriched in design variables through a newly proposed technique before proceeding to the AM phase. Different parallelization techniques are proposed for each phase, using the PVM message passing protocol. During the genetic evolution the concurrent evaluations of candidate solutions is used whereas the domain decomposition technique is employed during the AM-based optimization.*

1 INTRODUCTION

In the literature, the available aerodynamic shape optimization methods can be discerned in either calculus-based or stochastic techniques. In the first class of methods, after defining a cost function, its gradient with respect to design the variables has to be approximated. The so-called gradient-based optimization methods make use of approximated sensitivity derivatives to iteratively drive the design variables to their optimum values, through steepest descent or similar techniques. A possible way to calculate sensitivity derivatives is through the solution of an adjoint equation in terms of co-state variables. Governing flow equations are treated as constraints by introducing them to the cost function through Lagrange multipliers. It is appropriate to mention earlier works based on control-theory by Pironneau [1], and Jameson [2]; in the last decade several applications of the adjoint method (AM), in either continuous or discrete mode, brought to light (for example [3], [4], [5], [6]). Their main advantages are the fast convergence properties, and the independency of the computational cost from the number of design variables However, through the adjoint technique new development efforts are due each time a new optimization problem (with different objective or an updated flow model) is elaborated.

On the contrary, stochastic optimizers are ready-to-use supplements to any available flow analysis software. Among the various Evolutionary Algorithms in use, Genetic Algorithms (GA) [7] have found widespread use. Their main advantage is that their solution is not trapped to local minima, regardless the starting solution. However, they are costly procedures since they require a great number of evaluations (CFD routine calls) for the individuals during their genetic evolution. This cost can be reduced using "cheap" neural network-based pre-evaluations and/or concurrency [9], [10].

Another possibility for accelerating GA-based optimization methods is through their hybridization with deterministic methods [11]. This paper presents a hybrid GA-AM optimization method for the design of aerodynamic shapes, based on CFD techniques for unstructured grids. The role of GAs is to perform a randomized search and locate, with affordable cost, a good starting solution for the adjoint method. The starting solution, being close to the optimum one, allows for the adjoint method to locate the final solution, within a reduced number of iterations. A new technique, based also on GAs, is proposed for the enrichment of the after-GAs control-point set with new points, as required for the continuation with the AM. In this paper particular emphasis is given to the parallelization of the hydrid optimization tool on distributed memory computing platforms. At each phase (GA, AM), parallelization is employed in a different way. During the genetic evolution the concurrent evaluations of candidate solutions is used, its advantage being the extremely low communication between processors. On the contrary, the domain decomposition technique is used for the parallel solution of the adjoint equations. In each new iteration step, the generated unstructured grid is first partitioned in as many subdomains as the number of processors out using an in-house partitioning tool based on genetic algorithms [13]. The successive direct-adjoint solutions are then executed using the partitiomed grid, with regular communication of data among subdomains. For the parallelization, a cluster of heterogeneous workstations is used under the PVM message passing protocol.

The inverse design of ducts, isolated airfoils and 2D cascades, with inviscid flow considerations, is demonstrated in the results section.

2 OPTIMIZATION TOOLS

In the context of the present method, all aerodynamic shapes are parameterized using Bezier-Bernstein polynomials. In the inverse design of an airfoil, two separate Bezier curves that share common starting and ending points are used for the description of its pressure and suction sides. The reason for abandoning the use of circular arcs in the front and rear part of the airfoil is that they increase the model complexity as far as the adjoint method is of concern. With pure Bezier curves, a single mathematical form models stands for any part of the shape. Consequently, sensitivity derivatives can be readily computed. On the other hand, the reason for fixing both the leading- and trailing-edges is to be able to also fix the airfoil chord. The next to the leading-edge control point in each of the two airfoil sides defines the local tangent direction of the airfoil shape; limits are often imposed on the allowed values of its coordinates depending upon the design problem at hand. In general, both x- and y-coordinates of the control points can be defined as the design parameters. However, it should be mentioned that, such a definition does not affect the computing cost of the adjoint technique, but in the genetic optimization the number of control parameters has to be kept as low as possible. For the same reason, during the genetic optimization phase, control-point abscissas are often fixed and can be re-adjusted only during the adjoint technique.

2.1 The Genetic Optimization Method

GAs are robust stochastic optimizers which operate on a population of individuals. Each individual corresponds to a candidate solution and is represented by a chromosome containing the concatenated design parameters. This can be realized by using either binary or real coding. Using various selection criteria, reproductive trials are allocated with some bias towards their fittest members. The offspring produced by applying the parent recombination operators form the next generation with possibly improved mean fitness.

For the design of an aerodynamic shape with prescribed pressure distribution $\overline{p}(s)$ along its contour, the cost function I is defined as

$$I = \frac{1}{2}\int_{wall} dp(s)^2 ds \qquad , \qquad dp(s) = p(s) - \overline{p}(s) \tag{1}$$

and this should be minimized.

2.2 The Adjoint-based optimization Method

In this section, the adjoint technique for the design of aerodynamic shapes, based on inviscid flow considerations, will be briefly presented. For detailed information, the reader should turn to the cited papers. The cost function is defined as in eq. 1. Through the introduction of the co-state variables $\mathbf{\Lambda} = (\Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4)^T$ the flow equations are enforced as a constraint into the cost function giving rise to the following augmented cost function

$$I^* = I - \iint_\Omega \mathbf{\Lambda}^T \cdot (\frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y}) d\Omega \tag{2}$$

It should be pointed out that the space integral is taken over the current computational space defined by fixed inlet, outlet or infinite boundaries and the sought for aerodynamic shape. The latter undergoes continuous modifications up to its final, converged shape. Consequently, variations of the augmented cost function have to properly account for volume changes, as well as changes in the flow field. Bezier-Bernstein polynomials are used to describe these shapes as $x(t) = C_i(t)X_i$ and $y(t) = C_i(t)Y_i$, (X_i, Y_i) being the i-control point coordinates, $C_i(t)$ the Bezier coefficients and t the Bezier parameter. Expressions for the variation of the arc-length ds or the outward normal vector $\mathbf{n} = (n_x, n_y)$ can be readily derived [6]. Using the assumption that any point on the current shape moves to a new position with the same t value, and by taking variations in eq. 2 it is a matter of mathematical rearrangement to get

(a) The governing adjoint equation,

$$[A]^T \frac{\partial \mathbf{\Lambda}}{\partial x} + [B]^T \frac{\partial \mathbf{\Lambda}}{\partial y} = 0 \tag{3}$$

By satisfying this equation, the volume integral in eq. 2, which contains variations of flow variables, is automatically eliminated; this causes cost function variations δI^* to become independent from any flow field variation.

(b) A set of boundary conditions to be satisfied by the $\mathbf{\Lambda}$ distribution along the contour of the domain. These aim at eliminating the dependence of δI^* upon boundary flow variations. Along the solid walls, they include the constraint

$$dp - \Lambda_2 n_x - \Lambda_3 n_y = 0$$

which is a direct outcome of the velocity slip-condition.

(c) The expressions of the sensitivity derivatives with respect to the design parameters, used to correct the Bezier point coordinates. For the y-coordinates, these expressions are derived from the following equation

$$\begin{aligned} \delta I^* &= \sum \frac{\partial I^*}{\partial Y_i}\delta Y_i = \sum \int (\frac{1}{2}dp^2 - p\ dp)\frac{\dot{x}}{\sqrt{(\dot{x}^2+\dot{y}^2)}} \\ -\ p(\Lambda_2 \frac{1-n_x^2}{\sqrt{\dot{x}^2+\dot{y}^2}} &- \Lambda_3 \frac{n_x n_y}{\sqrt{\dot{x}^2+\dot{y}^2}})\dot{C}_i \ +\ (\mathbf{F}^T \frac{\partial \mathbf{\Lambda}}{\partial x} + \mathbf{G}^T \frac{\partial \mathbf{\Lambda}}{\partial y})C_i n_y)ds\ \delta Y_i \end{aligned} \quad (4)$$

where the summation symbol is taken over the entire set of control points and dotted quantities stand for derivatives with respect to t. A similar expression for the variation of the x-coordinates of the Bezier points is similarly obtained. The update of the control points location $Y_i = (X, Y)$ is carried out through a steepest descent scheme as $Y_i^{new} = Y_i^{old} - \eta \partial I^*/\partial Y_i$ with positive η values. Limits can be selectively imposed to restrict the space of possible solution values.

Eq. 3 is similar to the linearized system of the Euler equations. A time-derivative is added to them and its discretization and solution are performed by means of the same technique used by the direct solver. For inviscid flows, this is based on the node-centered, finite-volume method and a time-marching scheme. The computational domain is discretized through unstructured grids with triangular elements. The control-volume around any grid node is defined by successively connecting the midnodes of the edges incident upon the node with the barycenters of the surrounding triangles. The inviscid fluxes are computed via the flux vector splitting scheme with MUSCL extrapolation for higher order accuracy. A pointwise-implicit Jacobi procedure is used to update the solution variables at each time-step.

3 ACCELERATION THROUGH MULTI-PROCESSING

Multi-processing on distributed computing platforms is used as the means to reduce the computing cost. This is implemented in a different way in the two optimizing phases (*GA* and *AM*).

(a) During the *GA*-based optimization, multiple candidate solutions are simultaneously evaluated on different processors. Due to the coarse-grain parallelization, the computing cost is divided approximately by the number of available processors.

(b) During the *AM*-based optimization, the domain-decomposition technique is applied. The unstructured grid is decomposed (using a fast *GA*-based partitioner described below) into a number of subdomains equal to the number of available processors. All subdomains contain the same number of grid entities. Consequently, during the solution of the adjoint equation, all processors are evenly loaded. Communication at the interfacial nodes is regularly required; this practically defines the efficiency of the parallelization [12].

All computations are carried out on a cluster of networked *Intel* processors, under *PVM*.

3.1 Unstructured Grid Partitioning

The partitioning of an unstructured grid is viewed as a minimization problem formulated on its equivalent graph. A fast and effective *GA*-based partitioning technique, developed by the same authors and described in detail in [13], [14] is used. Here, only a brief outline of the method will be presented. Implicit to the partitioner are a recursive bisection scheme (so that only 2^n partitions can be created) and a multilevel scheme for each bisection. In what follows, the terms "nodes" and "edges" will refer to graph nodes (i.e. triangular elements) and graph edges (i.e. grid segments), respectively.

A multilevel partitioner starts by creating a sequence of graphs, each of which is derived from the previous one through coarsening. During each coarsening step, graph nodes are clustered into groups and weights are assigned to newly formed nodes and edges. For graph nodes the wights stand for the sum of the weights of the constituent nodes whereas edge weights stand for the number of edges shared between groups of clustered nodes.

Any graph is then mapped onto a square parametric space (Φ, Ψ). created through a Laplacian filter with $0 \leq \Phi, \Psi \leq 1$. In this space, two point-charges A and B are allowed to float, creating potential fields $F(\Phi, \Psi)$ around them. At any point P in the (Φ, Ψ) space, F_P is computed as follows

$$F_P = F(\Phi_P, \Psi_P) = e^{-r_{PA}} - e^{k_B r_{PB}} \tag{5}$$

where

$$r_{PM} = \sqrt{(\Phi_P - \Phi_M)^2 + (\Psi_P - \Psi_M)^2}, \quad M = A, B \tag{6}$$

For each bisection, the free-parameters are $(\Phi_A, \Psi_A, \Phi_B, \Psi_B, k_B)$, $k_B < 0$ and these are controlled by the *GA*. For each pair of point charges, potential values are computed over the graph nodes. These are sorted and the graph nodes above and below the median are assigned to the first and second subdomain, respectively. Thus, the only objective left is the minimization of the interface between the two subdomains.

The partitioning method described above is applied to the coarser graph at each bisection. The bisection of the coarsest graph is then ""injected"" to the finer one, up to the starting graph. This is carried out using heuristics that locally improve the interface through a limited number of migrations in a narrow zone close to the current interface.

4 THE OVERALL OPTIMIZATION ALGORITHM

The *GA*-based optimization precedes the *AM*-based one, so that the overall parallel optimization algorithm is described by the following steps:

Step 1 Create the starting population, by randomly selecting the free-parameter values, within predefined search-space.

Step 2 Evolve this population for N_{GA} generations by applying selection, crossover and mutation operators. The required evaluations in each generation are carried out concurrently using all the available processors (N_{PROC}). Each candidate solution is associated with a processor; on this processor an unstructured grid is generated and the flow equations are solved. Fitness scores from the different processors are communicated to a master processor that undertakes all genetic operations.

Step 3 The starting point for the AM-based optimization is the best solution found from the GA. The corresponding unstructured grid is decomposed in N_{PROC} equally-loaded subdomains using the GA-based partitioning method. Repetitive calls of the direct and adjoint equation solvers are performed until the cost function is minimized. Between these calls, the geometry is corrected according to the steepest descent technique and the unstructured grid is modified, as described in the previous section.

Below, we will keep track of the basic steps, either parallel or sequential, of the parallel solution algorithm for the AM-based optimization:

(1) The starting aerodynamic shape is formed using the best solution found by the GA-based optimization, after its enrichment in control points. (sequential).

(2) An unstructured mesh is generated and its equivalent graph is defined (sequential).

(3) This grid is partitioned into 2^n subdomains using the GA-based partitioner (sequential).

(4) Each subdomain data is communicated to the corresponding processor.

(5) The direct flow problem is solved using the parallel Euler solver. Communication is required during the numerical solution for the exchange of data along the interfaces (parallel).

(6) The adjoint equations are solved on the existing grid partition using flow data residing on each processor. Communication is required at each iteration as in the direct solver (parallel).

(7) The master processor gathers pressure and co-state variable values as well as their spatial derivatives along the solid walls described through Bezier curves.

(8) The Bezier control points are corrected using the corresponding sensitivity derivatives and the aerodynamic shape contour is updated. Return to step (2).

In the above algorithm, the sequential tasks are handled by the master, whereas the parallel tasks are executed by all the participating processors.

5 RESULTS AND DISCUSSION

Three inverse design problems are used for the validation of the proposed algorithm, namely the re-designs of (a) a two-dimensional duct, the target being the pressure distribution along its sidewalls at $M_{2,\mathrm{is}} = 0.4$, (b) the isolated NACA 0012 airfoil so as to reproduce its pressure distribution at zero incidence and $M_\infty = 0.4$ and (c) a planar compressor cascade with known pressure distribution at $a_1 = 47deg.$ and $M_{2,\mathrm{is}} = 0.4$.

In all these problems the flow was inviscid and the design variables consist of the y-coordinates of the Bezier control-points used to model the unknown shapes, with fixed x-coordinates.

Figs. 1, 2 and 3 illustrate the reference and the optimum computed shapes as well as the target and the final pressure distributions. Both geometries and pressure distributions match each other satisfactory. The profiles shown in the above figures have all been computed using *GA* for approximately 20 generations with a small population (30 individuals) and the *AM* afterwards.

A practical problem that had to be faced in all these runs and the remedy proposed herein are discussed below. In general, the adjoint method requires more control points, to be distributed along the modeled solid wall, than that previously used by the *GA*. The *GA*-based optimization starts with a reduced number of design variables, for both efficiency and effectiveness. This number is not capable of ensuring a deep convergence of the *AM*, so that at the end of the genetic evolution and before proceeding to the adjoint method, the actual set of control points should be enriched.

A fast and flexible procedure to accomplish this need makes also use of the capabilities of *GAs*. The aim is to create additional control points which, along with those computed by the *GA*, yield a contour that matches, as close as possible, the best airfoil shape computed at the end of the first optimization phase. It should be clarified that the target in this *GA*-based optimization is a geometrical contour (the cost-function expresses the deviation between two geometrical shapes) and does resort upon pressure distributions. Thus, the evaluation task bears almost negligible computing cost and the so-called ”enrichment procedure” is very fast. The user defines the number of additional Bezier points to be generated in the interval formed by pairs of successive Bezier points, computed in the first optimization phase. The search area for each new control-point is automatically defined by splitting quadrilaterals formed by existing control points in a number of sub-areas, as in fig. 4 (left). By doing so, the new control-polygon maintains the general characteristics of the coarser one. Two alternative techniques have been successfully used in our tests. In the first of them, the x-coordinates of the new control-points are fixed whereas their y-coordinates are allowed to vary within the upper and lower borderlines of the quadrilateral, as in fig. 4 (right). The second does not use any of the existing control-points and searches for new points that match the ”target” contour. All three options perform equally well. One of the side-effects of the enrichment in control points is that the starting contour of the *AM*-based optimization could be locally wavy, due to the addition of new control points. As a consequence, the starting fitness value for the AM-based optimization is expected to be different (usually higher) than the ending one of the GA.

We will finally demonstrate the application of the aforementioned enrichment technique in the second test case (NACA 0012). In fig. 5 one may see the optimum airfoil shape computed at the end of the 21^{nth} generation of the GA. The cost of the GA-based optimization, up to this point, was about 620 flow evaluations, with a polulation size of 30. The convergence of the GA is shown in fig. 6, where this is extended beyond the 21^{nth} generation, for the purpose of comparison. The starting profile for the AM-based optimization, after the enrichment in control- points is also shown in fig. 5. In the right part of the same figure, one may notice the differences in the pressure distribution the new blade yields compared to the final profile from the GA. This explains the higher starting cost of the AM-based optimization in fig. 6. It should be noticed that, in order to include the AM convergence in the same figure, each cycle of the AM was given the cost of two equivalent evaluations (one for the direct method and the other for the adjoint equations).

6 CONCLUSIONS

A method was presented for aerodynamic shape optimization problems. It is based on a hybrid scheme, where the first optimization phase is based on Genetic Algorithms and the second on the Adjoint Method. The efficient use of a multiprocessor system required completely different parallelization techniques during the two phases. The GA-based method assigns the evaluation of each individual to a single processor. The speed-up is very high since the communicated data between processors are limited to the design parameters and the outcome of the evaluation, provided that the relation between population and number of processors is the proper one. The AM-based method is parallelized using the subdomain technique and a fast unstructured grid partitioning tool. The AM-based optimization requires more design variables than the GA-based one, so an enrichment technique is newly proposed. The coupling of GAs with the AM gives a deeper and faster convergence, measured in terms of equivalent evaluations.

To assess the gain from the parallelization, fig. 7 presents the number of evaluations per processor in the third case, during the GA-based optimization phase. Eight processors are used and the loading of each processor is shown as percentage of the total number of evaluations. With eight processors, the speed-up during the second, AM-based optimization phase is quite low (2.7) due to coarseness of the mesh used for inviscid flow computations. The speedup values for four and two processors were 2.8 and 1.8, respectively; so, this kind of problems, it is recommended to use a smaller number of processors during the second optimization phase.

Acknowledgment: The development of the GA-based optimization and of the unstructured grid partitioning method have been funded by Dassault Aviation.

References

[1] O. Pironneau, On optimum design in fluid mechanics, *J. Fluid Mech*, **64**, 97-110 (1974).

[2] A. Jameson, Aerodynamic design via control theory, *J. Scientific Computing*, **3**, 23-260 (1988).

[3] A. Jameson and J. Reuther, Control theory based airfoil design using the Euler equations, *AIAA Paper 94-4272* (1994).

[4] A. Iollo, G. Kuruvila and S. Ta'asan, Pseudo-time method for optimal shape design using the Euler equations, *AIAA Journal*, **34**, 1807-1813 (1996).

[5] G. W. Burgreen and O. Baysal, Three-Dimensional aerodynamic shape optimization of wings using discrete sensitivity analysis, *AIAA Journal*, **34**, 1761-1770 (1996).

[6] W. K. Anderson and V. Venkatakrishnan, Aerodynamic design optimization on unstructured Grids with a continuous adjoint formulation, *AIAA Paper 78-257* (1997).

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

[8] K. C. Giannakoglou, Designing turbomachinery blades using evolutionary methods, *ASME Paper, 99-GT-181* (1999).

[9] K. C. Giannakoglou, A design method for turbine blades using genetic algorithms on parallel computers, *ECCOMAS 98*, John Wiley & Sons (1998).

[10] 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* (1999).

[11] T. J. Martin and G. S. Dulikravich, Aero-Thermal Analysis and Optimization of Internally Cooled Turbine Airfoils, in *ISABE 97-7165*, 1232-1341, (1997).

[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.*, **160**, 89-110 (1998).

[13] A. P. Giotis and K. C. Giannakoglou, An unstructured grid partitioning method based on genetic algorithms, *Advances in Engineering Software* (1998).

[14] A. P. Giotis and K. C. Giannakoglou, B. Mantel and J. Periaux, Efficient Partitioning Methods for 3-D Unstructured Grids Using Genetic Algotithms, in *Evolutionary Algorithms in Engineering and Computer Science*, John Wiley & Sons, 425-434, (1999).

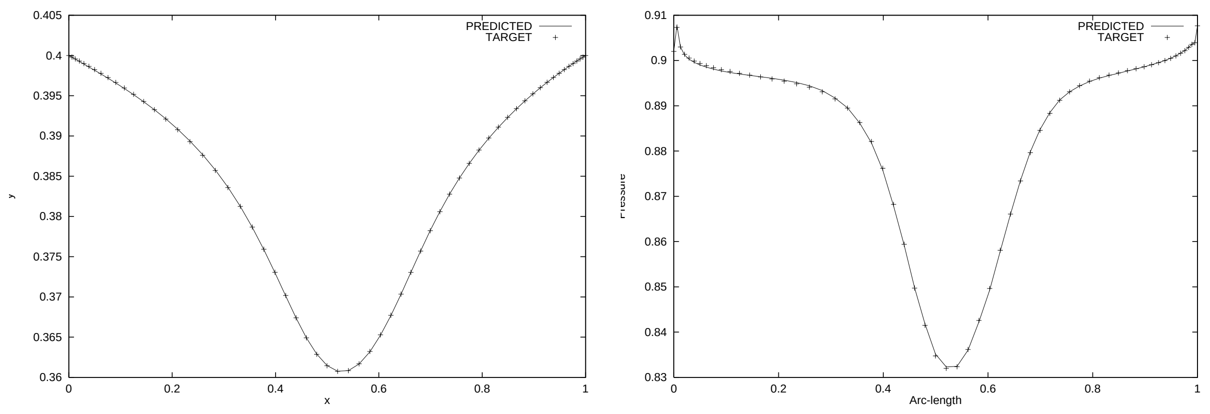

Figure 1: Target and best predicted results in a convergent-divergent duct. Left: duct shapes. Right: pressure distributions.

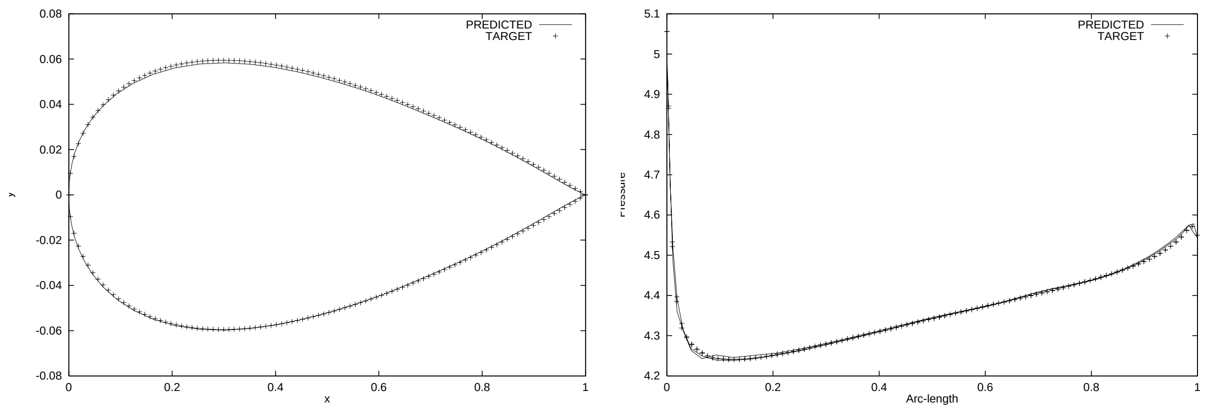

Figure 2: Target and best predicted results in an isolated NACA0012 airfoil. Left: airfoil shapes. Right: pressure distributions.

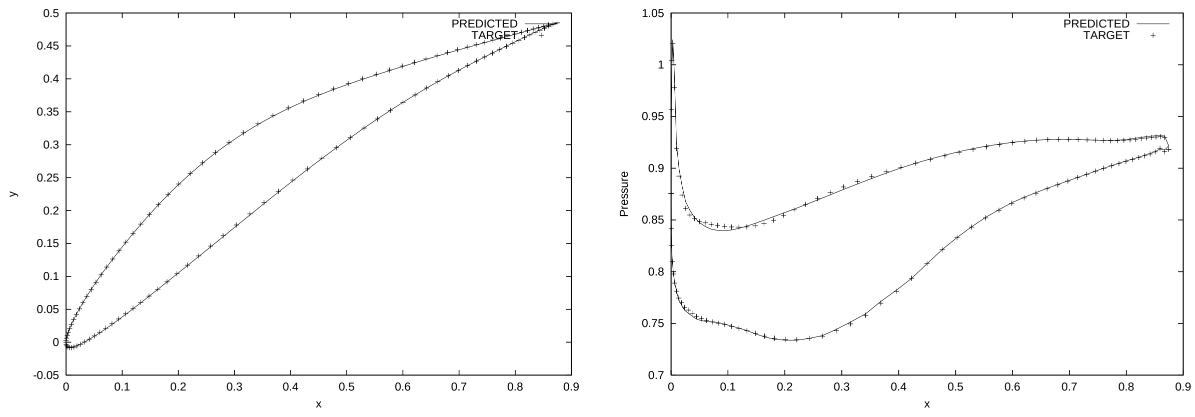

Figure 3: Target and best predicted results in a cascade blade. Left: blade shapes. Right: pressure distributions.

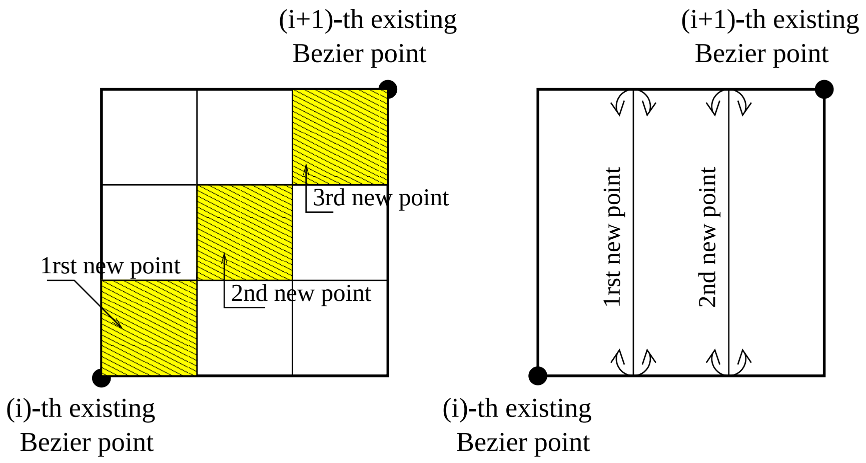

Figure 4: Enrichment in new Bezier-control points after the GA. The two alternatives.

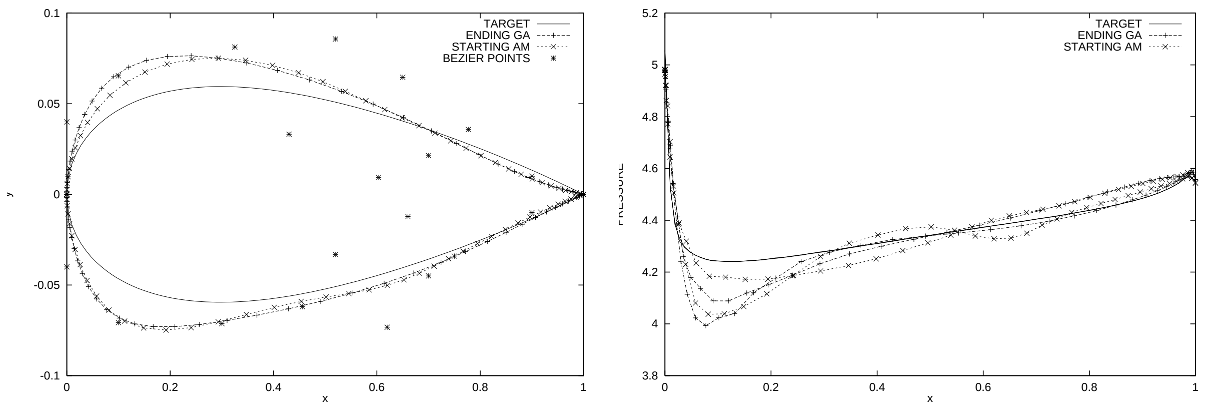

Figure 5: Profile shapes and pressure distributions at the end of GA - start of AM (after enrichment). Control-points after the enrichment.

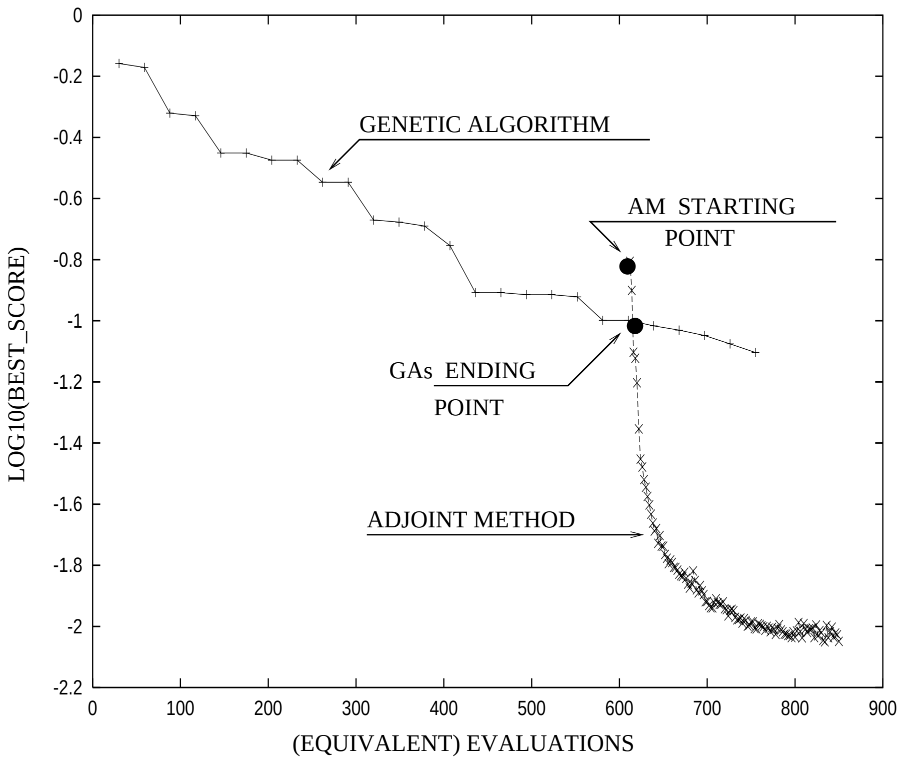

Figure 6: Convergence history of the hybrid GA-AM optimization for the design of NACA0012.

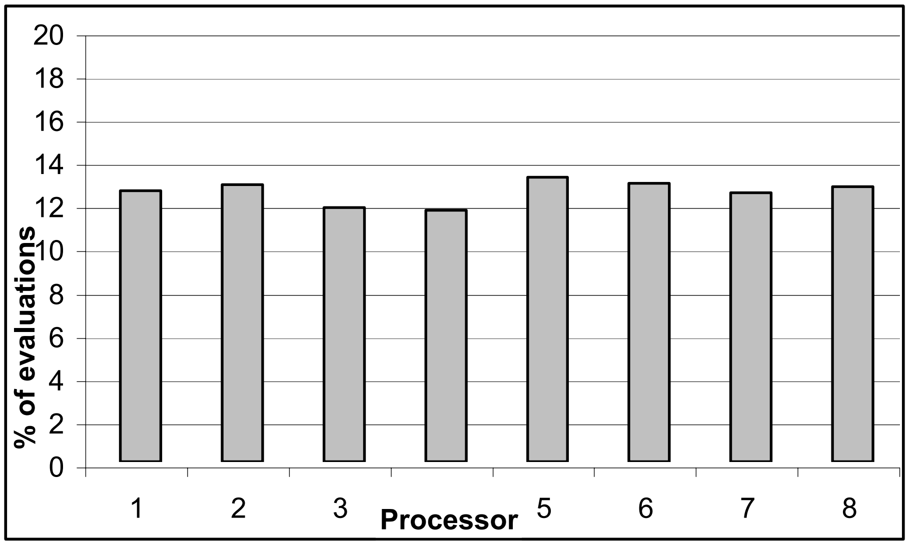

Figure 7: *GA*-based optimization: Processor-loading (in terms of evaluations) in a 8-processor platform.