AIAA Journal
Vol. 37, No. 5, May 1999

# Airfoil and Wing Design Through Hybrid Optimization Strategies

Alessandro Vicini* and Domenico Quagliarella†
*Centro Italiano Ricerche Aerospaziali, 81043 Capua, Italy*

**Real-world design problems need robust and effective system-level optimization tools inasmuch as they are ruled by several criteria, most often in multidisciplinary environments. In this work a hybrid optimization algorithm has been obtained by adding a gradient-based technique to the set of operators of a multiobjective genetic algorithm. This makes it possible to increase the computational efficiency of the genetic algorithm while preserving its favorable features of robustness, problem independence, and multiobjective optimization capabilities. Aerodynamic shape design problems, including both airfoil and wing designs, are considered.**

## I. Introduction

SEVERAL techniques are available today for design through numerical optimization.[1] In particular in the field of aerodynamic design, beyond methods developed ad hoc and characterized by inverse design capabilities, the techniques more properly related to direct optimization include mature gradient-based methods and more recent approaches such as automatic differentiation, control-theory-based methods, and genetic algorithms (GAs). Generally, one method cannot be said to be superior to others without reference to a specific problem that needs to be faced. The characteristics of importance that need to be evaluated include 1) generality of the formulation vs dependence on the problem; 2) robustness, intended as the capability of avoiding local optima, vs the need for human interaction and expertise; 3) capability of multiple-objective vs single-objective optimization; and 4) computational efficiency vs the need for large computational resources.

From this point of view, the choice of one particular optimization technique implies the renunciation of some possible advantages in favor of others. On the other hand, because aerodynamic shape design represents only a part of the overall design of a flying vehicle and because the need for an effective multidisciplinary approach to the design task is rising, it is important for an optimization tool to combine as many of the favorable characteristics just listed as possible while avoiding the shortcomings. In this sense, a hybrid approach to optimization, in which techniques of different nature are used at the same time, may have extremely beneficial results. In fact, hybrid optimization may exploit the most favorable features of the methods that are combined while masking the corresponding shortcomings.[2]

In this work, a hybrid optimization tool, developed by incorporating a gradient-based optimization routine among the operators of a multiobjective GA,[3] is applied to aerodynamic shape design problems. GAs belong to the class of evolutionary optimization procedures that finds its philosophical basis in Darwin's theory of survival of the fittest.[4] In an attempt to mimic the process of biological evolution, a set of design alternatives, representing a population in this metaphorical transposition, is permitted to evolve through successive generations so as to promote the individuals that better adapt themselves to the environment, i.e., those that better meet the design requirements. Each element is characterized by the value of its fitness, which is the measure of how fit it is for the given environment, in other words, how good the corresponding solution is for the problem at hand. The process of evolution is realized in the reproduction phase using a selection criterion driven by the value of the fitness of the individuals, so that bias is allocated to the best-fit members of the population. The individuals selected for reproduction are recombined using genetic operators (crossover and mutation), so that a combination of their most desirable characteristics may be obtained in the offspring; hence, elements characterized by higher fitness are produced in subsequent generations.

These search methods rely only on the evaluation of the fitness of the elements and do not require the computation of gradients; therefore, they are less susceptible to pitfalls of convergence to a local optimum and can successfully deal with disjoint or nonconvex design spaces. Moreover, they are capable of facing the problem of multiple-objective optimization in a straightforward fashion, using the notion of domination among solutions.[5] These characteristics make GAs very attractive optimization tools and explain the considerable growth of interest that has been focused on them in recent years for engineering applications.[6]

The major weakness of GAs lies in their relatively poor computational efficiency inasmuch as they generally require a high number of evaluations of the objective function. For this reason, the use of GAs may become impractical when this evaluation is expensive, as happens for aerodynamic optimization applications where the solution of complex partial differential equation systems is necessary.

Coupling a GA with a different optimization technique can be an effective way to overcome its lack of efficiency while preserving its favorable features. Many different strategies to hybridize the GA can be realized; the simplest one is that of using the best solution found by the GA as starting point for a subsequent optimization with the other adopted method.[7] However, a closer interaction between the different algorithms, rather than the two-stage optimization described, may more favorably combine the best features of both methods and provide results better than those obtainable using either of the two techniques.

Some representative results obtained with the hybrid GA (HGA) will be demonstrated for aerodynamic shape design problems, including both airfoil and wing optimization test cases.

## II. Hybridization of the GA

A simple GA may by itself be considered as the combination of two different search techniques, namely, crossover and mutation, that are characterized by different behaviors when searching the parameter space. Crossover generates the candidate solutions (offspring) through a combination of two existing ones (parents); the solutions thus obtained, independently from the way the parents are selected and combined, can be far from the starting ones. Thus, crossover is a powerful tool to search the design space and single out the region where the global optima lie, but it lacks the capability of effectively refining the suboptimal solutions found. On the other hand, mutation has a more local effect because the modifications it produces are generally small in the coded parameter space. Hence, mutation has two important roles in simple GAs: 1) to provide the capability to effectively refine suboptimal solutions and 2) to reintroduce in the population the alleles lost by the repeated application of crossover, maintaining population diversity. However, the rate of mutation needed for these two tasks may be different; in particular,

Received May 18, 1998; presented as Paper 98-2729 at the AIAA 16th Applied Aerodynamics Conference, Albuquerque, NM, June 15–18, 1998; revision received Nov. 10, 1998; accepted for publication Jan. 20, 1999.


*Research Engineer, Aerodynamics and Propulsion Department. E-mail: a.vicini@cira.it. Member AIAA.

†Research Engineer, Aerodynamics and Propulsion Department. E-mail: d.quagliarella@cira.it. Member AIAA.

whereas mutation is very good for maintaining population diversity, its refining capabilities may not be optimal for every class of problems. There is in fact a broad class of problems, namely, the ones where the fitness function is differentiable, for which gradient-based techniques are much more efficient to locally improve a given solution. This suggests the introduction of a gradient-based routine among the set of operators of the GA; mutation is then prevalently left with the role of keeping the diversity among population elements at an optimum level.

The GA developed adopts a bit string codification of the design variables; this does not prevent the use of operators requiring real number list encoding, such as extended intermediate crossover and word-level mutation.[8] In these cases, the binary string is decoded into a real number list, the operator is applied, and the set of modified variables is encoded back into a bit string. This scheme allows the use of a free mix of different types of operators.

In particular, in the applications that are described, mutation is alternatively applied at bit or word level, i.e., directly on the bits of the binary coding, or on the integer ranging from 0 to $2^n - 1$, where $n$ is the number of bits adopted for each variable. As an alternative to the standard one-point crossover, the extended intermediate recombination (EIR), described by Mühlenbein,[9] has been used. Let $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ be the selected parent chromosomes; with EIR, the offspring variable is given by

$$z_i = x_i + \alpha_i (y_i - x_i) \tag{1}$$

where $\alpha_i$ is chosen randomly in the interval $[-d,\ 1+d]$. A value of $d = 0.2$ was used for the present applications, and relation (1) was used at word-level coding of the variables. Finally, the selection operator adopted is a random walk: The population is distributed over a toroidal landscape,[8] a starting point is chosen at random, and the parents selected are the best-fit or nondominated ones,[5] for single- and multiobjective problems, respectively, among those met in two successive random walks of assigned number of steps from that starting point.

Besides the described genetic operators, as said before, a routine performing a gradient-based optimization (with a conjugate gradients technique) has been included and called hill climbing operator (HCO). The HCO is used as follows: Through the application of the selection and crossover and mutation operators, an intermediate generation is created from the current one; afterward, if the hybrid option is activated, some individuals may be selected and fed into the HCO to be improved and then introduced into the new generation, as shown in Fig. 1. Regarding the choice of the elements to be fed into the gradient operator, in the case of single-objective optimization, three different strategies are possible.

1) Only the best-fit individual of the current generation is chosen.
2) A number of elements determined by an assigned probability is picked using the selection operator.
3) A number of elements determined by an assigned probability is picked in a purely random fashion.

Of course, these strategies determine different levels of selection pressure, decreasing from strategy 1 to strategy 3; therefore, the relative performance will depend on the optimization problem. The enumerated scheme can be naturally extended for multiobjective optimization. In this case, the elements are not ranked on the basis of a scalar fitness function but are just divided into two classes: the dominated and the not dominated.[5] The set of not-dominated individuals (Pareto front), updated after each new generation, is composed of all potential solutions of the problem, satisfying the design criteria at different levels of compromise. When multiobjective problems are formulated, strategy 1 becomes the (random) selection of a number of elements determined by an assigned probability from the current set of Pareto optimal solutions, whereas strategies 2 and 3 remain the same. Of course, the HCO is by its nature capable of dealing only with scalar objective functions; thus, when multiobjective problems are faced, the objective function fed into the HCO is obtained through a weighted linear combination of the $n$ problem objectives, i.e., as $\mathrm{obj} = \alpha\,\mathrm{obj}_1 + (1-\alpha)\mathrm{obj}_2$ in the case of $n = 2$. The weighting factor $\alpha$ can be chosen at random or assigned explicitly to favor one of the objectives.

Note that the aid provided by the gradient operator simply consists of its capability to improve to some extent the selected individuals; in other words, its role is that of introducing improvements that will then be processed and exploited by the GA, which remains the driving engine of the procedure. From this point of view, the use of this operator has to be limited to the minimum necessary to provide the desired effect, which is that of improving the convergence characteristics of the procedure without causing premature convergence to local minima. Based on these considerations, the use of the HCO is subject to the following rules:

1) The hybrid mechanism (Fig. 1) is not used at each generation but only after each assigned block of $K$ generations.
2) It is not necessary, and it would be detrimental for the computational performances, for the HCO to carry out a converged optimization each time. Therefore, only one or two gradient iterations are generally prescribed.

Furthermore, as the gradient operator basically has to behave as an improved mutation operator, the beneficial effects of hybridization also can be obtained by making it operate only on a subset of the active design variables, thus reducing the number of objective function evaluations that must be carried out each time. The success in this case will depend on the degree of cross correlation among the design variables. The total number of evaluations of the objective function needed $N_e$ can be estimated as follows:

$$N_e = N_{\mathrm{pop}} N_{\mathrm{gen}} [1 + (p_{\mathrm{hco}}/K) N_{\mathrm{it}} (\eta N_{\mathrm{var}} + 4)] \tag{2}$$

where $N_{\mathrm{pop}}$ and $N_{\mathrm{gen}}$ are the population size and the total number of generations, $p_{\mathrm{hco}}$ is the HCO probability, $K$ is the frequency in terms of number of generations for its activation, $N_{\mathrm{it}}$ is the number of gradient iterations, $N_{\mathrm{var}}$ is the number of design variables, and $\eta \in [0, 1]$ is the factor determining the size of the subset of design variables that is passed to the HCO. In the following applications, the design variables that need to be frozen are chosen at random each time the gradient operator is used; in this way, a different subset of design variables is passed to the operator each time.

## III. Applications to Airfoil Design

In the airfoil inverse design problem, a pressure distribution corresponding to a design point determined by the values of Mach number and angle of attack is given, and the geometry of the airfoil producing this target pressure distribution must be found. In this case, the objective function to be minimized is computed by

$$\mathrm{obj} = 10 \int_S \left(c_p - c_p^{(t)}\right)^2 \mathrm{d}s \tag{3}$$

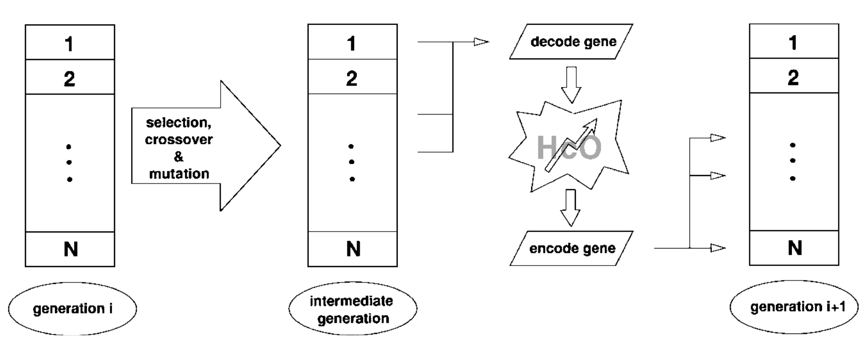



**Fig. 1 HGA.**

where $c_p$ and $c_p^{(t)}$ are the current and target pressure distributions, respectively, and $S$ is the current airfoil contour; the fitness is then obtained as $f = 1/\mathrm{obj}^2$. A full potential transonic flow solver, with nonconservative formulation, has been used to calculate the flowfield. The airfoil geometry is represented by means of two fifth-order B-spline curves, for the upper and lower parts. The coordinates of the control points of the B-spline constitute the design variables[8]; seven control points are used both for the upper and lower surfaces of the airfoil, including those fixed at the leading and trailing edges, for a total of 18 design variables (the first control points at the leading edge can move only in direction $y$). The problem consists in the reconstruction of the CAST-10 airfoil[10] at $M = 0.765$ and $\alpha = 0$. This problem has been solved using a NACA 0012 as initial guess, which can be considered an absolutely generic starting point.

The design variables have been encoded using 8-bit strings (giving a chromosome length of 144 bit) and a 50 individuals population evolved for 100 generations; the hybrid strategies have been activated to select on average only one individual every other generation and to carry out two gradient iterations ($p_{\mathrm{hco}} = 0.02$, $K = 2$, $N_{\mathrm{it}} = 2$, $\eta = 1$). Hence, to consider the same total number of objective function evaluations, the hybrid strategies must be judged approximately at generation 70. Two different GAs, characterized by the set of operators described in Table 1, have been used with and without hybridization. In both cases, the elitist strategy has been used by explicitly transferring the best-fit individual from one generation to the subsequent one. Figure 2 shows the convergence histories, each one averaged over 10 successive trials characterized by different starting populations. The convergence history obtained by the application of the gradient-based method by itself is also shown in Fig. 2; in this case, of course, there are no generations of individuals, but the convergence history is reported in such a way that the number of required objective function evaluations can be obtained from

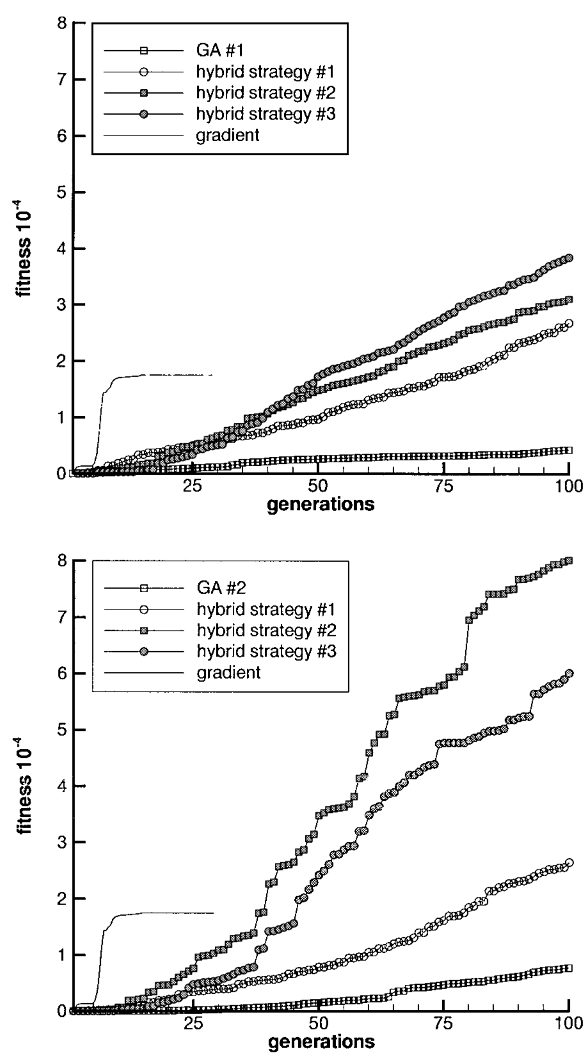



**Fig. 2 Convergence histories for the CAST 10 inverse design problem.**

**Table 1 Set of operators of the two GAs used**

| Operator | GA 1 | GA 2 |
|---|---|---|
| Selection | Random walk, 5 steps | Random walk, 3 steps |
| Crossover | Extended intermediate | One point |
| Crossover probability | 1 | 1 |
| Mutation | Bit | Word |
| Mutation probability | 0.02 | 0.02 |

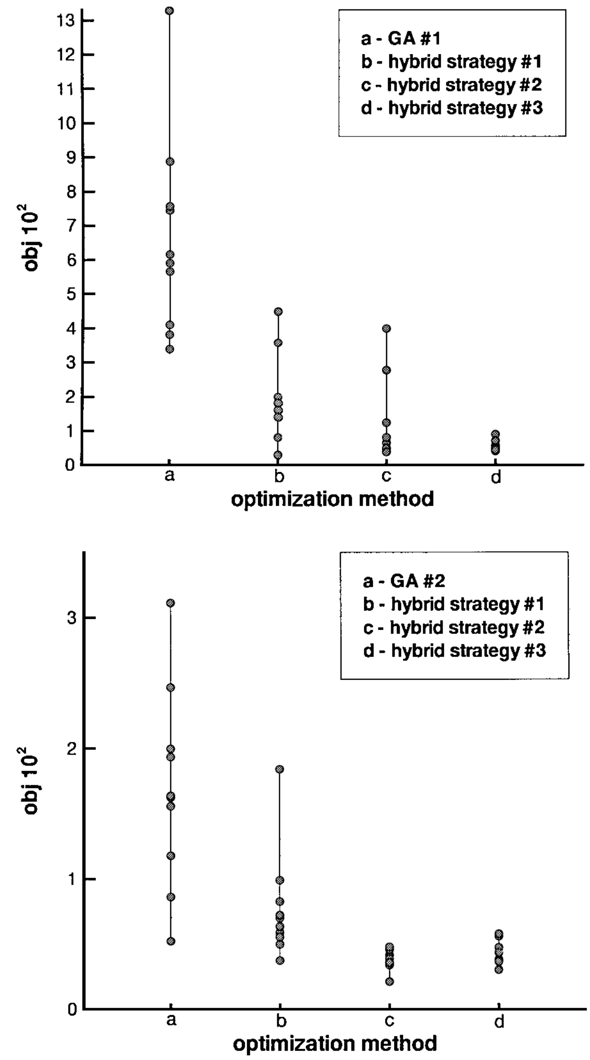



**Fig. 3 Scatter of results obtained in 10 different runs for the CAST 10 inverse design problem.**

the same scale (1 generation = 50 evaluations). Note that a restart procedure had to be used in this case to take the solution out of a local minimum where it was stuck after a few iterations.

As can be seen, for a given GA, hybridization is always beneficial, meaning that a better result can be found with the same amount of computations or that the same result can be obtained with a substantial reduction of computation needed (ranging in this case from 30 to 75%). In particular, strategy 1, when the HCO is applied only to the best-fit individuals, appears as the less effective, probably due to an excessive selection pressure. At the same time, the behavior of the gradient-based method is considerably improved from the point of view of the robustness.

Another important characteristic that needs to be considered is the statistical dispersion of the results obtained starting from different initial populations; in fact, if it is correct to judge the convergence characteristics of a given GA by averaging the results of a number of runs, from an application-oriented point of view it is more important for the algorithm to guarantee satisfactory convergence performances even on a single-run basis. Figure 3 shows all of the values of the objective function obtained at the end of each of

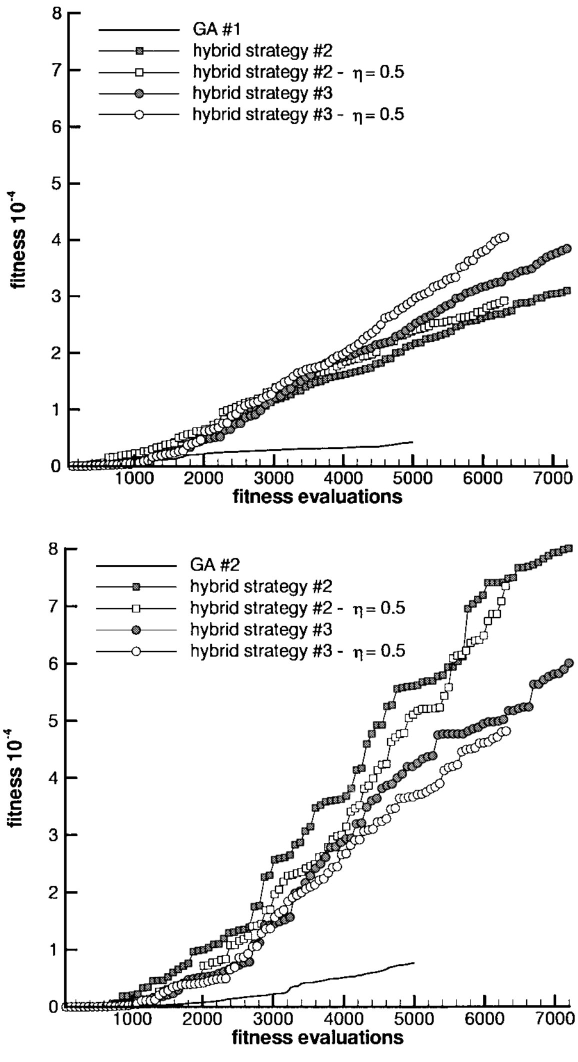



**Fig. 4 Comparison of the convergence histories obtained by letting the HCO operate on all design variables or on a 50% subset.**

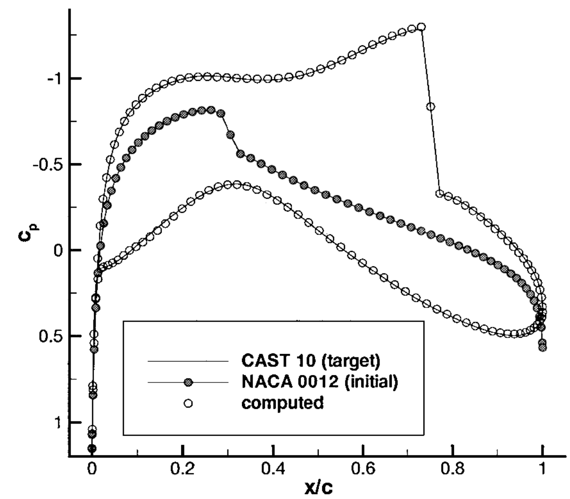



**Fig. 5 Target, initial, and computed pressure distributions.**

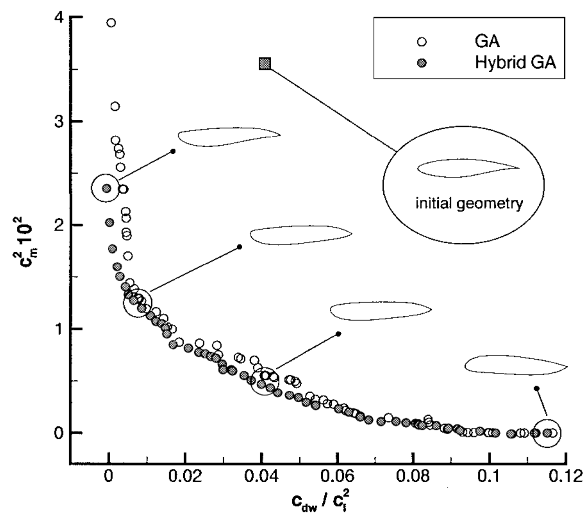



**Fig. 6 Pareto fronts obtained for the RAE 2822 optimization problem.**

the 10 different runs, for each one of the algorithms used; it can be observed how the scatter of the results provided by both basic GAs is much higher than that obtained using the corresponding hybrid algorithms. In particular, the best behavior from this point of view is obtained when the elements to be fed into the gradient operator are chosen at random, so that the level of selection pressure is not increased too much.

The same runs have then been repeated using $\eta = 0.5$; in this way, the HCO acts on a subset of 9 design variables out of 18, which are chosen at random each time. The convergences obtained are shown in Fig. 4, limited to the hybrid strategies 2 and 3, i.e., those giving the best performances. In this case, the actual number of fitness evaluations has been used for the $x$ axis, to better compare the results. We see how freezing some of the design variables has a positive effect when GA 1 is used, whereas for GA 2 convergence is slowed to some extent. Considering that the design variables for this problem are strongly cross correlated, as it is not possible to move one control point of the B-splines independently from the others, this result shows that this approach can generally be used with success. In Fig. 5 one of the pressure distributions obtained is shown together with the target and initial ones.

An example of multiobjective optimization is then presented, consisting of reducing the wave drag of the airfoil while keeping the corresponding pitching moment under control, for a fixed lift coefficient and maximum thickness. The airfoil chosen as initial geometry is the Royal Aircraft Establishment (RAE) 2822 (Ref. 11), at a design point $M = 0.78$ and $c_l = 0.75$. The constraint on lift coefficient is satisfied by letting the flow solver find the angle of attack that produces the specified lift, whereas the thickness of the airfoil is scaled to the desired value after each geometry modification; in this way, every solution is a feasible one. The two objective functions have been evaluated as $\text{obj}_1 = c_{dw}/c_l^2$ and $\text{obj}_2 = c_m^2$. A population of 100 individuals was permitted to evolve for 100 generations; selection was carried out by means of a three steps random walk, with one-point crossover ($p_c = 1$) and bit mutation ($p_m = 0.02$). Differently from the inverse design earlier described, the geometry of the airfoil has been represented as a linear combination of the initial one, $y_0$, and a number of modification functions, $y_i$:

$$y = y_0 + \sum_{i=1}^{N} x_i y_i \tag{4}$$

The coefficients $x_i$ of this combination are the design variables; the functions $y_i$ have been obtained as the difference between the initial geometry and the geometries of other airfoils chosen from a database, so that a particular aerodynamic effect can be associated to each design variable. The allowable range assigned was $x_i \in [-0.2, 1.2]$, $i = 1, N$, and 12 design variables were used. The same run was then repeated using the gradient operator, on average, on one element per generation, chosen at random from the current Pareto front, and carrying out only one gradient iteration. The run in this case was stopped at generation 86, to establish the comparison for the same total number of evaluations of the objective functions.

The Pareto fronts thus obtained are shown in Fig. 6, together with the starting point. It can be seen how the result provided by the hybrid algorithm is a Pareto front characterized by solutions of higher quality and more uniformly distributed; only 12 of the

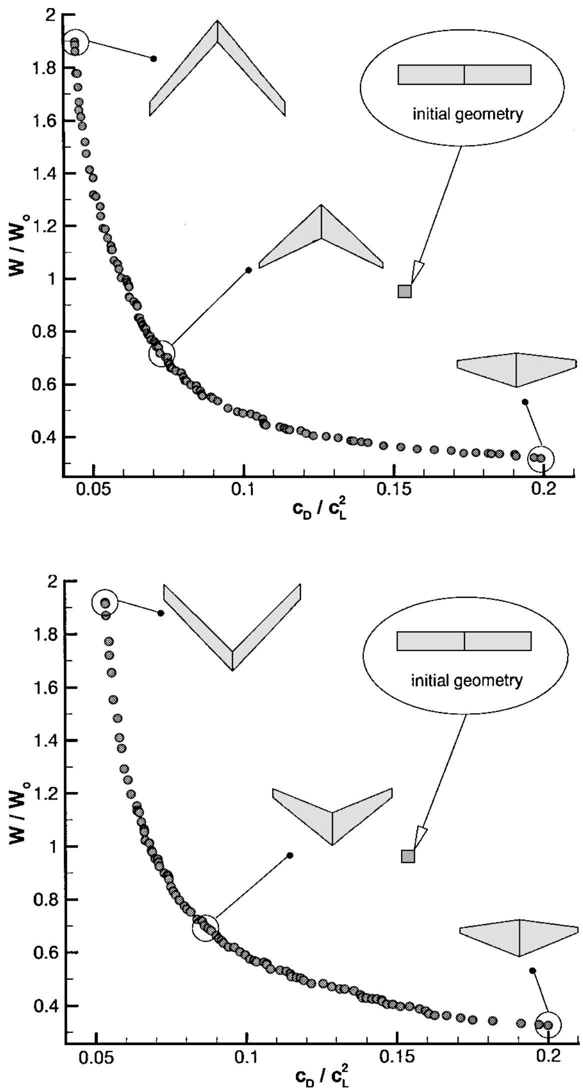



**Fig. 7 Pareto fronts obtained for the wing planform optimization.**

solutions found by the GA are not dominated by those obtained with the HGA.

## IV. Applications to Wing Design

Wing design is a highly multidisciplinary task; therefore, the use of designer expertise is necessary to obtain realistic results, unless the various design criteria and off-design considerations can be included in the formulation of the optimization problem. The multiobjective optimization approach offers great advantages for these kind of problems, avoiding the need of arbitrarily interrelating the different design criteria into a single scalar objective function.

GAs have already been applied to the problem of planform wing design, taking into account aerodynamic and structural requirements. In Ref. 12, structural rigidity considerations are included in the optimization, but a single-objective GA is used, with a selection of design variables that allows limited freedom in the definition of the planform shape. In Ref. 13, nonplanar wing shapes are allowed to maximize the $L/D$ ratio with the condition that the wing does not fail under the applied loads. In both cases, the aerodynamic models are limited to subsonic flow, with structural models based on simple beam theory. In Ref. 14, a parallel Pareto GA is used for the planform optimization of a transonic wing, minimizing aerodynamic drag and structural weight and maximizing tank volume; however, the dimensions of the design space are limited by the use of only three design variables.

In this work, the HGA has been applied to the optimization of the shape of a wing for transonic flow conditions, modifying both the planform and the wing section. The results have been obtained by coupling the HGA with a finite difference full potential flow solver.[15]

**Table 2 Design parameters for the wing planform optimization**

| Design variable | Initial value | Allowable range | Selected wing |
|---|---|---|---|
| $\lambda$ | 0.0 | [0.1, 1.0] | 0.10 |
| $\Lambda$ | 1.0 | [0.0, ±50] | 39.6 |
| $\theta$ | 0.0 | [−10, 10] | −5.2 |
| AR | 7.0 | [6.0, 8.0] | 6.41 |
| $t/c\|_r$ % | 12 | [12, 15] | 12.1 |

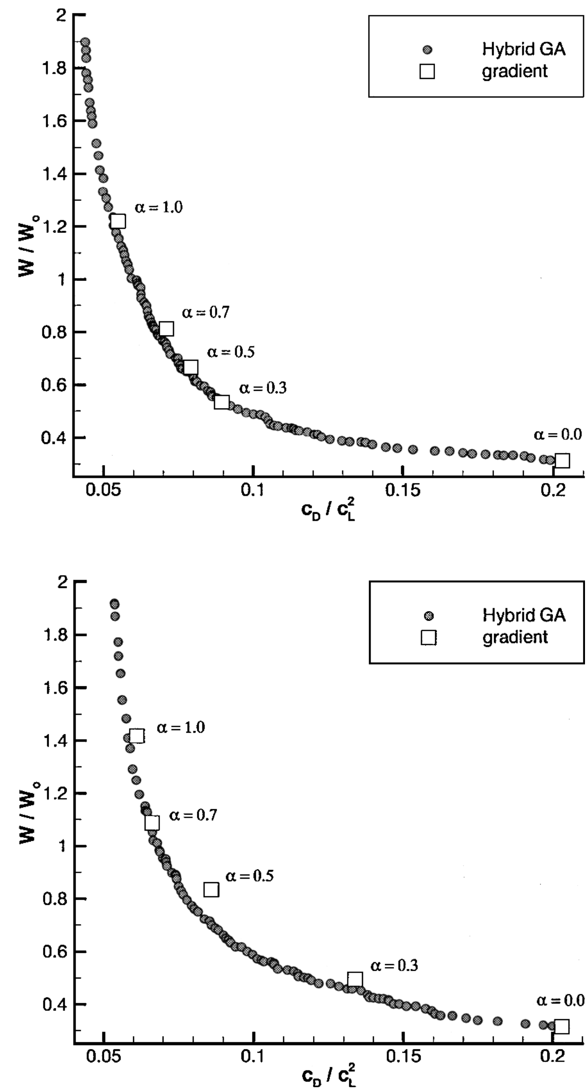



**Fig. 8 Comparison between the Pareto fronts and the results obtained through gradient-based method.**

First, the wing planform design has been accomplished by minimizing inviscid aerodynamic drag, which combines induced and wave drag, and structural weight, at a given Mach number $M = 0.85$ and lift coefficient $c_L = 0.5$. The starting point chosen is a straight, untwisted, and untapered wing of aspect ratio AR = 7, with an RAE 2822 airfoil; for simplicity, the wing planform is maintained trapezoidal, so that all geometric characteristics vary linearly from the root section to the tip. A total of five design variables have been used: four of these act directly on the wing planform, namely, the taper ratio $\lambda$, the sweep angle at 25% of the chord $\Lambda$, the aspect ratio AR, and the twist angle $\theta$; moreover, the thickness at the wing root has also been included among the design parameters, while the thickness at the wing tip has been fixed at $t/c|_t = 10\%$. The wing surface is kept constant, so that the average wing loading is not changed during optimization. In Table 2 the initial values of the design parameters are reported together with the prescribed allowable ranges. The wing twist is distributed symmetrically between the root and the tip; in other words, a twist angle $\theta$ corresponds to an increase in local incidence of $\theta/2$ at the tip and a decrease of $\theta/2$ at the root. The wing weight is computed using the algebraic equation of Ref. 16; this equation combines analytical and empirical (statistical) methods and shows design sensitivity and prediction accuracy that make it possible to use it with success for preliminary design. As can be seen from Table 2, two separate runs have been carried out exploring separately the positive or negative sweep design spaces; in fact, the choice of a positive or negative swept wing is based on considerations of a different nature, including stability and handling characteristics.

The selection has been carried out through a three-step random walk, with one-point crossover ($p_c = 1$) and bit mutation ($p_m = 0.1$); a population of 64 individuals was permitted to evolve for 50 generations. The HCO has been used on one element for each generation, selected from the current Pareto front, with $N_{\mathrm{it}} = 1$. The Pareto fronts obtained are shown in Fig. 7 (where $W_0$ is a reference weight), together with the planform of the wings corresponding to the extremities and to the center of the fronts. It can be seen that, for a given value of aerodynamic drag, the negative swept wings are heavier than the corresponding positive swept ones; therefore, almost all of the solutions with a negative sweep angle would be dominated if the two Pareto fronts were merged. The fronts are populated by 116 and 100 individuals, in the case of positive and negative swept wings, respectively. The use of the HCO does not prevent the development of the complete Pareto front; on the contrary, the solutions are uniformly distributed along the fronts without the need of niching techniques. To evaluate how well these solutions are representative of the real Pareto fronts, the same test case has been solved using the gradient-based method by itself, with the problem formulated through the weighted linear combination approach, i.e., the same used by the HCO: $\mathrm{obj} = \alpha\ \mathrm{obj}_1 + (1-\alpha)\mathrm{obj}_2$; five different values for $\alpha$ have been used: 1, 0.7, 0.5, 0.3, and 0. The solutions thus obtained are compared in Fig. 8 with the Pareto fronts provided by the HGA. As can be observed, in the better cases these solutions lie on the Pareto fronts, and in some cases they fall in the dominated solutions region. Observe, in neither case is the gradient method capable of finding the solution of minimum drag, i.e., that corresponding to $\alpha = 1$.

In Fig. 9, the values of the design parameters of the solutions belonging to the Pareto fronts are shown as a function of aerodynamic drag. As can be expected, the sweep angle varies in an almost linear fashion from the maximum allowable values (positive or negative) to zero. Similarly, the aspect ratio is at its maximum at the low-drag end of the front and rapidly diminishes to the minimum as drag increases. It can be seen how changing the aspect ratio from 8 to 6 implies an increase of aerodynamic drag of about 80%; this increase is composed of induced drag for 60% and of wave drag for the remaining 40%. The taper ratio remains approximately at the minimum allowable value, $\lambda = 0.1$, for most of the front, assuming higher values only for the solutions corresponding to minimum drag; in the case of positive sweep, minimum drag is obtained for a taper $\lambda = 0.5$; whereas for negative sweep, a higher value is necessary, $\lambda = 0.78$. The role of twist is essentially that of redistributing the spanwise loading so as to better approach the elliptic distribution; this explains the opposite sign of the twist angle that is obtained when positive or negative swept wings are considered. It can also be observed that higher values of twist are necessary in the latter case. Finally, the behavior of the thickness at the root section appears less intuitive; only at the low-weight end of the front can a clear trend be observed, with an almost linear increase of drag with the thickness.

As anticipated, after optimization of the planform a further improvement of the aerodynamic characteristics has been obtained by modifying the shape of the wing section. One of the solutions belonging to the Pareto front has been selected as starting point;

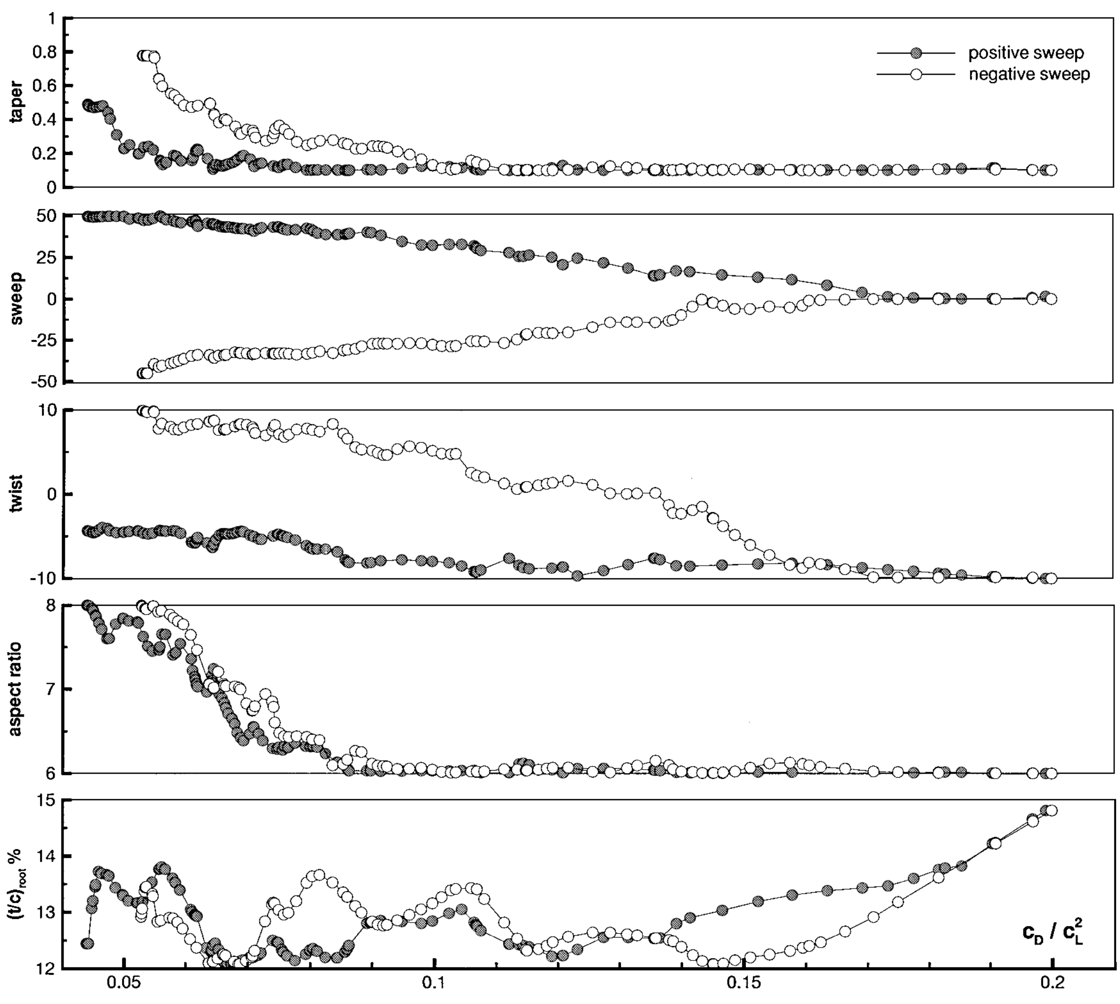



**Fig. 9 Design parameters for the solutions on the final Pareto fronts.**

attention has been focused on the positive sweep angles, and the geometry chosen lies approximately at the center of the front, characterized by $c_D/c_L^2 = 0.776$ and $W/W_0 = 0.65$. The wing section has been modified using the same shape functions technique described in Sec. III; 12 design variables also have been used in this case, and for simplicity the wing profile has been maintained constant in the spanwise direction. Inasmuch as modifying the wing section may have a strong impact on the aerodynamic characteristics but not on the structural weight, which is going to remain almost constant, the optimization problem in this case has been formulated so as to reduce wave drag with control on pitching moment; in fact, the latter determines the level of trim drag. The design objectives have then been formulated as $\text{obj}_1 = c_{Dw}/c_L^2$ and $\text{obj}_2 = (c_M - 0.5)^2$; as in the earlier case, the lift coefficient has been fixed to $c_L = 0.5$, and the maximum thickness has been maintained at the value obtained by the previous run at each spanwise station. The same GA parameters used for the wing planform optimization have been adopted, except for the mutation rate, which has been reduced to $p_m = 0.04$, and for the population size, which has been increased to 100. The Pareto front obtained is shown in Fig. 10, where some of the corresponding wing section shapes are also shown. Depending on the actual design requirements, it is now possible to extract from this front the solution with the desired characteristics. In particular, in Fig. 11 the drag rise curve (at $c_L = 0.5$) of the wing before optimization of the section is compared with those of three wings extracted from the front, i.e., the low-drag end of the front, corresponding to an unconstrained optimization, and the two solutions characterized by the same pitching moment and wave drag coefficients, respectively, of the initial wing. As can be seen, the first two solutions provide an overall improvement of the drag rise curve; at the design point $M = 0.85$ the reduction of wave drag is 32 drag counts for the unconstrained solution and 22 for the fixed $c_M$ one. On the other hand, when the drag coefficient is kept constant so that a reduction of $c_M$ can be achieved, lower wave drag values are obtained for Mach numbers lower than the design one, but with a steeper increase at higher Mach numbers. Finally, in Fig. 12, the pressure distributions on three wing sections, for the initial wing and the low-drag end of the front, are shown.

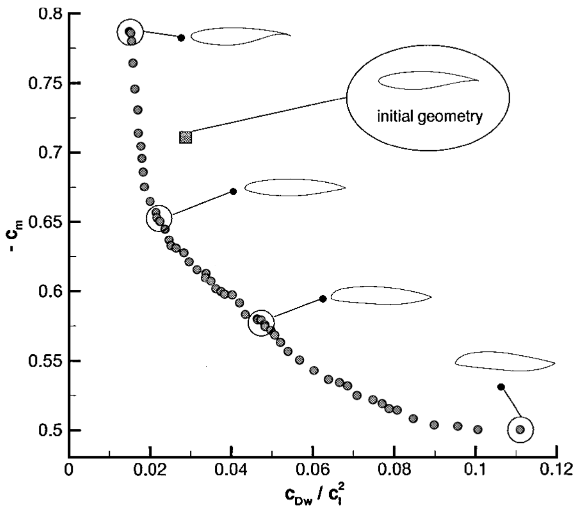



**Fig. 10 Pareto front obtained for the wing section optimization.**

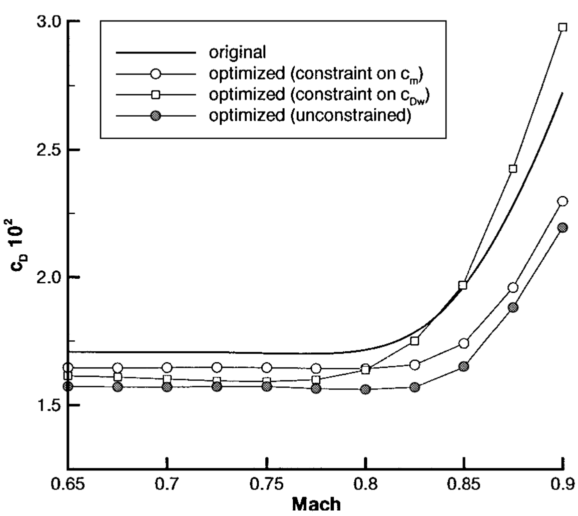
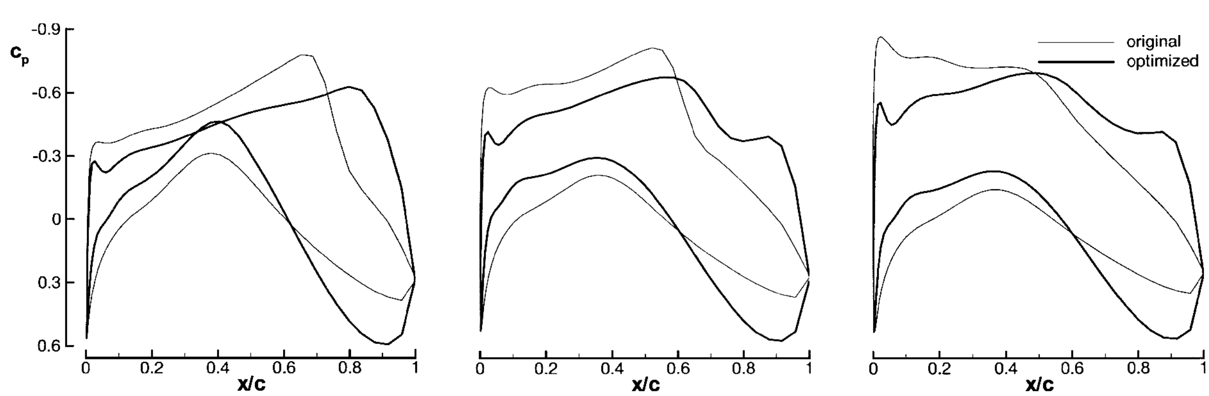



**Fig. 11 Drag rise curve of the original wing and of wings selected from the Pareto front.**

## V. Conclusions

In most practical applications, design problems are governed by several criteria, often derived from different disciplines. To approach such design tasks, robust and effective system-level optimization tools are needed. GAs are characterized by a number of favorable features that make them attractive for these types of problems; multiobjective optimization, in particular, appears particularly suited for multidisciplinary environments because it allows determination of sets of Pareto optimal solutions within the design space, representing different levels of compromise among the design goals or constraints. Therefore, the designer can make the choice introducing an a posteriori selection criteria. The flexibility of the design process, thus, can be increased, because the need for interrelating criteria of different natures is avoided, and the effect of changing constraints can be evaluated off-line.

An effective algorithm for multiobjective applications has been developed through a hybrid approach, by coupling a multiobjective GA to a gradient-based operator. Applications to multiobjective airfoil and wing design have been presented. The basic idea to use a gradient-based routine in the fashion of a genetic operator derives from the observation that mutation in itself is not very effective as a refinement operator, leading to generally poor convergency speed; on the other hand, it has been demonstrated how the beneficial effects of the HCO can be exploited even when its use is considerably




**a)** $- 2y/b = 0.16$ **b)** $- 2y/b = 0.44$ **c)** $- 2y/b = 0.76$

**Fig. 12 Pressure coefficient on the original and optimized (unconstrained) wings.**

limited in terms of number of elements processed and computations carried out for each element. For the class of problems that has been investigated, significant improvements have been obtained both with respect to simple GAs, in terms of computational efficiency, and with respect to gradient-based approaches, in terms of robustness. In particular, it has been possible to use the HGA with success even in the case of multiobjective problems, when a weighting approach must be used to compose a scalar objective function each time resort is made to the gradient-based operator.

## References


[1] *Optimum Design Methods for Aerodynamics*, AGARD R-803, Papers 1–11, Nov. 1994.

[2] Davis, L., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991, Chaps. 4, 5.

[3] Quagliarella, D., and Vicini, A., "Coupling Genetic Algorithms and Gradient Based Optimization Techniques," *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, edited by D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, Wiley, Chichester, England, UK, 1997, pp. 289–309.

[4] Holland, J. H., *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press, Ann Arbor, MI, 1975, Chap. 1.

[5] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison–Wesley, Reading, MA, 1989, pp. 197–201.

[6] Ahmed, Q., Krishnakumar, K., and Neidhoefer, J., "Applications of Evolutionary Algorithms to Aerospace Problems—A Survey," *Computational Methods in Applied Sciences '96*, Wiley, Chichester, England, UK, 1996, pp. 236–242.

[7] Poloni, C., "Hybrid GA for Multi Objective Aerodynamic Shape Optimisation," *Genetic Algorithms in Engineering and Computer Science*, edited by G. Winter, J. Périaux, M. Galán, and P. Cuesta, Wiley, Chichester, England, UK, 1995, pp. 397–416.

[8] Vicini, A., and Quagliarella, D., "Inverse and Direct Airfoil Design Using a Multiobjective Genetic Algorithm," *AIAA Journal*, Vol. 35, No. 9, 1997, pp. 1499–1505.

[9] Mühlenbein, H., "The Science of Breeding and Its Application to Genetic Algorithms," *Genetic Algorithms in Engineering and Computer Science*, edited by G. Winter, J. Périaux, M. Galán, and P. Cuesta, Wiley, Chichester, England, UK, 1995, pp. 59–82.

[10] Ray, E. J., and Hill, A. S., *CAST-10-2/DOA 2, Airfoil Studies Workshop Results*, NASA CP-3052, Nov. 1989.

[11] Cook, P. H., McDonald, M. A., and Firmin, M. C. P., "Aerofoil RAE 2822—Pressure Distributions, Boundary Layer and Wake Measurements," AGARD AR-138, Paper A6, May 1979.

[12] Doorly, D. J, Peiró, J., and Oesterle, J.-P., "Optimisation of Aerodynamic and Coupled Aerodynamic-Structural Design Using Parallel Genetic Algorithms," *Proceedings of the AIAA/NASA/USAF 6th Multidisciplinary Analysis and Optimization Symposium*, AIAA, Reston, VA, 1996, pp. 401–409.

[13] Anderson, M. B., and Gebert, G. A., "Using Pareto Genetic Algorithms for Preliminary Subsonic Wing Design," AIAA Paper 96-4023, Sept. 1996.

[14] Obayashi, S., Yamaguchi, Y., and Nakamura, T., "Multiobjective Genetic Algorithm for Multidisciplinary Design of Transonic Wing Planform," *Journal of Aircraft*, Vol. 34, No. 5, 1997, pp. 690–693.

[15] Jameson, A., and Caughey, D. A., "Numerical Calculation of the Transonic Flow Past a Swept Wing," NASA CR-153297, June 1988.

[16] Torenbeek, E., "Development and Application of a Comprehensive, Design-Sensitive Weight Prediction Method for Wing Structures of Transport Category Aircraft," Technical Univ. Delft, TR LR-693, The Netherlands, Sept. 1992.


A. D. Belegundu
*Associate Editor*