A98-31522

ICAS-98-2.9.1

MULTI-OBJECTIVE STRATEGIES FOR COMPLEX OPTIMIZATION PROBLEMS IN AERODYNAMICS USING GENETIC ALGORITHMS

J. Périaux * M. Sefrioui *† B. Mantel*

* Dassault Aviation. 78 Quai Marcel Dassault, 92214 Saint-Cloud, France.
† LIP6, Université Paris VI. Case 169, 4 place Jussieu, 75252 Paris, France.

Abstract

Deterministic optimizers are powerful tools for solving optimization problems dealing with smooth, unimodal objective functions. They require few objective function evaluations, particularly if compared to stochastic optimization methods. However, deterministic methods face serious problems for search spaces with rugged landscapes.

Genetic Algorithms (GAs) are a stochastic derivative-free search proceduresrunning on a natural selection mode. A decisive advantage for GAs in complex industrial environment is robustness and simplicity. Indifference to problem specifics, codings of decision variables, process of population, randomized crossover and mutation operators are the main characteristics which contribute to the robustness of GAs.

There are three main sections in this paper. The first section introduces the fundamentals of (1) genetic algorithms with binary or floating point codings and (2) game theory and multiple objective optimization (cooperative and non-cooperative).

The second section presents two CFD applications: inverse problem in optimum shape design for a transonic Euler flow condition using Bezier spline parametrization of the airfoil population and lift maximization for a multielement landing configuration.

The third section deals with single and multi objective optimization of scattered waves by active control elements. The problem in finding optimal distribution of active control elements in order to minimize the RCS of perfectly conducting reflectors Electromagnetics. Both problems are solved by means of Genetic Algorithms via a fitness evaluation through the solution of the Maxwell equations corresponding to the RCS of the radar illumination. Pareto and Nash solutions are computed and compared using a combination of GAs and Games Theory. It is shown through numerical experiments that Nash Gas are faster and more robust than Pareto GAs, although Pareto solutions are better.

Fundamentals of GAs and Games Theory

Introduction to GAs

Despite traditional deterministic optimization methods have proven to be powerful and fast tools for solving optimization problems with smooth, differentiable and unimodal objective functions requiring few objective function evaluations when compared to stochastic optimization methods severe difficulties appear when dealing with highly multimodal functions or non-convex optimization problems. One of the most powerful stochastic alternatives to those methods are Genetic Algorithms [6,4]. GAs are blind search methods based on the mechanics of natural selection and Darwin's main principle : *survival of the fittest.*

The basic principle is a DNA-fitness parallel established between an individual and a solution in the one hand, and between an environment and a problem in the other : a *good solution* to a given problem is an individual likely to succeed in a given environment. Each individual (i.e. solu-

21st ICAS Congress
13-18 September 1998
Melbourne, Australia

tion) has a *fitness function* which measures how fitted it is for the environment (i.e. problem), or in other words how good the solution is.

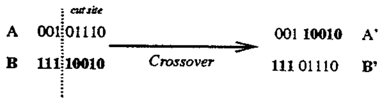
The most outstanding advantages of GAs are *robustness* and simplicity : they are computationally *simple* and *powerful* in their search for improvement and are not limited by restrictive assumptions about the search space (continuity, existence of derivatives, unimodality). Furthermore, they accommodate well with discontinuous environments and noise and rely on a careful balance between exploration of the search space and exploitation of the results.

Binary coding

As mentioned earlier, GAs are different from the conventional search procedures encountered in engineering optimization. To understand the mechanism of GAs, consider a minimization problem with a cost index $J = f(x)$, where the parameter is x. The first step of the optimization process is to encode x as a finite-length string. The length of the binary string is chosen according to the required accuracy. For a binary string of length $l = 8$ bits, the lower bound x_{min} for the variable x is mapped to 00000000 and the upper bound x_{max} is mapped to 11111111, with a linear mapping in between. Then for any given string, the corresponding value x can be calculated according to: $x = x_{min} + \frac{1}{2^l-1}.(x_{max} - x_{min})$. With this coding, the initial population is constituted by N individuals, and each of them is a *potential* solution to the problem. We must now define a set of GA operators that use the initial population and then create a new population at every generation. There are many GA operators, but the most important are *selection, reproduction, crossover* and *mutation.*

Selection consists in choosing the solutions which are going to form a new generation. The main idea is that selection should depend on the value of the fitness function: the higher the fitness is, the higher the probability is for the individual to be chosen (akin to the concept of *survival of the fittest*). But it remains a ***probability***, which means that is not a deterministic choice: even solutions with a comparatively low fitness may be chosen, and they may reveal very good in the course of events (e.g. if the optimization is trapped in a local minimum).

Reproduction is a process by which a string is copied in the following generation. It may be copied with no change, but it may also undergo a mutation, according to a fixed mutation probability P_m. However, the main way to fill up the new generation is through the operator called crossover.



First, two strings are randomly selected and put in the mating pool. Second, a position along the two strings is selected according to a uniform random law. Finally, based on the crossover probability P_c, the paired strings exchange all characters following the cross site. Clearly the crossover randomly exchanges a structured information between parents A and B to produce two offspring A' and B', which are expected to combine the best characters of their parents.

The last operator, called mutation, is a random alteration of a bit at a string position, and is based on a mutation probability P_m. In the present case, a mutation means flipping a bit 0 to 1 and vice versa. The mutation operator enhances population diversity and enables the optimization to get out of local minima.

The main parameters to adjust the convergence of GAs are the size N of the population, the length l of the bit string, the probabilities P_c and P_m of crossover and mutation respectively. Binary coding is used in the sequel for both the position of the elements in CFD for multielements configuration and in CEM for the location of active antennas in RCS minimization.

Floating-Point representation

Binary-coded genetic algorithms are very well suited to combinatorial problems and they facilitate theoretical analysis, but the robustness and implicit parallelism of genetic algorithms does not depend on the binary representation. There are various domains where floating-point representations have given better results [7].

Crossover

A two-point crossover for a real-coded genetic algorithm is almost the same as the one for binary-coded genetic algorithms. Let $A = (y_1, y_2, y_3, y_4, y_5, y_6)$ and $A' = (y'_1, y'_2, y'_3, y'_4, y'_5, y'_6)$ be the parents chosen during the selection process. If the first cut point c_1 is after the 3th gene and the second cut point c_2 is after the 5th gene, the crossover of A and A' will produce $B_1 = (y'_1, y'_2, y'_3, y_4, y_5, y'_6)$ and $B_2 = (y_1, y_2, y_3, y'_4, y'_5, y_6)$.

All the y_i are *real numbers* and not binary strings coding real numbers.

Non-Uniform Mutation

Most of the exploration is performed by crossover operators in binary-coded genetic algorithms. The exploration is mainly achieved through the exchange of building blocks. That makes mutation somewhat secondary in most cases. But in a real-coded representation, crossover cannot lead to *new* values of the variables because it is a mere exchange of floating-point numbers (the cutting point cannot be *within* a variable, it is always between two variables). And that is the reason why mutation is so important: it is the only way to introduce completely new values for the variables. Otherwise, the space search would be limited to a combination of the starting point variables. We used a non-uniform mutation[7]. If a gene y_i is to be mutated, the new value y'_i is randomly generated within the interval $[\mathrm{Min}_i, \mathrm{Max}_i]$.

$$y'_i = \begin{cases} y_i + (\mathrm{Max}_i - y_i).r.(1-\frac{t}{T})^b & \text{if a random digit is 0} \\ y_i - (y_i - \mathrm{Min}_i).r.(1-\frac{t}{T})^b & \text{if a random digit is 1} \end{cases}$$

r is a random number from [0,1],t is the the number of generations,T is the maximum number of generations, b is the refinement parameter, and Min_i and Max_i are the lower and upper bounds of y_i.

This adaptive mutation offers a clear balance between exploration and accuracy. At the first stages of the algorithm, we have $\frac{t}{T} \simeq 0$ which leads to a wide exploration whereas the last stages are devoted to refinement. Besides non-uniform mutation, we also used a Distance-Dependend Mutation (DDM) which dynamically computes the mutation rate of the parents according to their relative distance[13].

Floating point coding is used in the sequel in CFD for optimum shape design of airfoil represented by Bezier splines parametrization.

Games Theory

Introduction

The most obvious way to deal with a multiple objective optimization problems is to consider a scalar objective, obtained through a linear combination of the different objectives. However, this view suffers from several drawbacks. The choice of the weights associated to each criterion is arbitrary, and there is a notable loss of information. Besides, it is very sensitive to weights modifications[10].

The best solution seems to be a model which takes into account all the information without trying to aggregate it. Such models have long existed in Games Theory. Games Theory was first developped in Econonomy, to offer a model for conflictual situations[16], but it has been successfully applied to a vast range of problems, including population dynamics[14]. Any conflictual situation is considered as a *game*, where each *player* has to consider his own objective but still keep at the same time an eye on the evolution of the whole game.

There are two major families of games: *cooperative games*, such as Pareto games, and *non-cooperative games*, such as Nash games.

In the following, each criterion of a multi-objective optimization shall be associated to a player. Let A and B be 2 players, with scalar objective functions $f_A(\vec{x}, \vec{y})$ for A and $f_B(\vec{x}, \vec{y})$ for B. A controls all the variables of $\vec{x}$ and B controls all the variables of $\vec{y}$. $\bar{A}$ is the set of strategies for A and $\bar{B}$ is the set of strategies for B.

Pareto Equilibrium

The concept of Pareto optimality is the basis on which is grounded most of cooperative multiple objective optimizations[9] . It is based upon the principle of *dominance*:

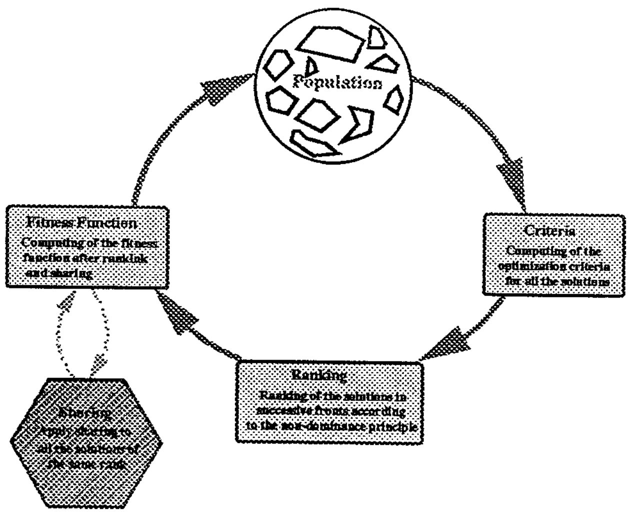

Population
Criteria
Computing of the optimization criteria for all the solutions
Ranking
Ranking of the solutions in successive fronts according to the non-dominance principle
Sharing
Apply sharing to all the solutions of the same rank
Fitness Function
Computing of the fitness function after ranking and sharing

Figure 1: Pareto GA

For a Pareto game with n players, a strategy $(\vec{v_1}^*, .., \vec{v_n}^*)$ **dominates** a strategy $(\vec{v_1}, .., \vec{v_n})$ iff:

$$\begin{cases} \forall i, 1 \leq i \leq n, f_i(\vec{v_1}^*, .., \vec{v_n}^*) \leq f_i(\vec{v_1}, .., \vec{v_n}) \\ \exists\, i \text{ such as } f_i(\vec{v_1}^*, .., \vec{v_n}^*) < f_i(\vec{v_1}, .., \vec{v_n}) \end{cases}$$

For a minimization problem, a strategy $(\vec{x}^*, \vec{y}^*)$ is Pareto optimal if it is non-dominated, or in other words if there is no other strategy $(\vec{x}, \vec{y}) \in \bar{A} \times \bar{B}$ such as $f_A(\vec{x}, \vec{y}) \leq f_A(\vec{x}^*, \vec{y}^*)$ and $f_B(\vec{x}, \vec{y}) \leq f_B(\vec{x}^*, \vec{y}^*)$

Sharing and niching techniques are the two main ingredients to build an algorithm mixing GAs and Pareto concepts [15,12]. Figure 1 shows the architecture of such an algorithm.

The main interest of the Pareto approach is to build a database of optimal solutions. Afterwards, interpolated solutions can be picked up for Engineering purpose according to the chosen weights of the criteria.

Nash Equilibrium

Nash optima define a noncooperative multiple objective optimization approach first proposed by J. F. Nash [8]. For an optimization problem with n objectives, a Nash strategy consists in having n players, each optimizing his own criterion. However, each player has to optimize his criterion given that all the other criteria are fixed by the rest of the players. When no player can further improve his criterion, it means that the system has reached a state of equilibrium called *Nash Equilibrium*.

A strategy $(\vec{x}^*, \vec{y}^*) \in \bar{A} \times \bar{B}$ is a Nash Equilibrium iff:

$$\begin{cases} f_A(\vec{x}^*, \vec{y}^*) = \inf_{x \in \bar{A}} f_A(x, \vec{y}^*) \\ f_B(\vec{x}^*, \vec{y}^*) = \inf_{y \in \bar{B}} f_B(\vec{x}^*, y) \end{cases}$$

Nash equilibria are very difficult to find when no explicit differentiable function is provided, but GAs can manage to do it. GAs and Nash strategy are merged in order to *build* Nash Equilibrium.

The Nash GA works as follows: let $s = XY$ be the string representing the potential solution for a dual objective optimization, where X corresponds to the first criterion and Y to the second one. Player 1 optimizes X (Y is fixed by Player 2) and Player 2 optimizes Y (X is fixed by Player 1). Each player is associated to a population. The Nash equilibrium is reached when neither player can further improve its criteria. Figure 2 sketches the Nash GAs approach (see [11] for more details).

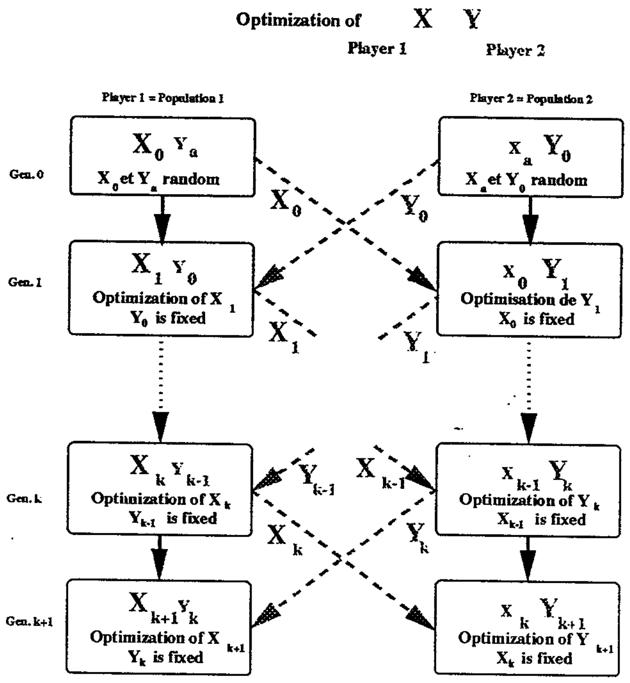


Figure 2: Nash GA strategy

Example of a two-objective optimization

Let player A and B have the following objective functions:

$$\begin{cases} f_A(x, y) = (x-1)^2 + (y-3)^2 \\ f_B(x, y) = (x-4)^2 + (y-2)^2 \end{cases}$$

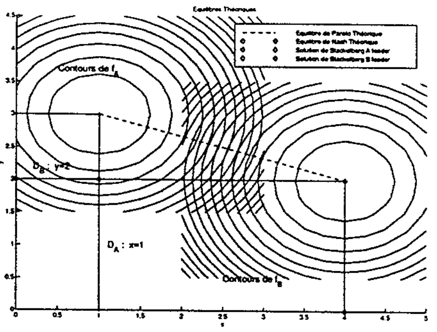


Figure 3: Analytical Equilibrium for Pareto and Nash Games

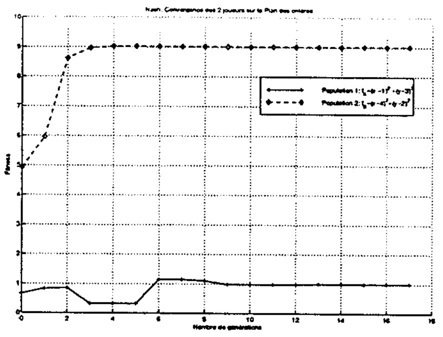


Figure 4: 2 players convergence, Nash GAs

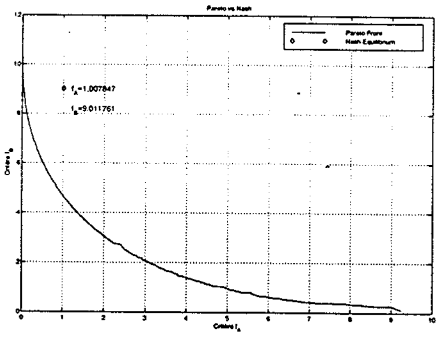


Figure 5: Pareto Equilibrium, Pareto GAs

Analytical Solution

To build the Analytical Pareto Equilibrium, we need to introduce the parametric function $f_p(x,y)$, where $0 \leq \lambda \leq 1$:

$$f_p(x,y) = \lambda f_A(x,y) + (1-\lambda) f_B(x,y)$$

The Pareto optimal solutions are the solutions of:

$$\begin{cases} \frac{\partial f_p(x,y)}{\partial x} = 0 \\ \frac{\partial f_p(x,y)}{\partial y} = 0 \end{cases}$$

Since λ varies in $[0,1]$, the points $P_0 = (4,2)$ and $P_1 = (1,3)$ are the extremities of the Pareto set, which is a line.

Figure 3 shows the contour of functions f_A et f_B along with the analytical results obtained for the various multiobjective optimization methods.

The Analytical Nash Equilibrium can be obtained through the intersection of the rational reaction sets D_A and D_B. D_A is the solution of $\frac{\partial f_A(x,y)}{\partial x} = 0$, while D_B is the solution of $\frac{\partial f_B(x,y)}{\partial y} = 0$. Since D_A is the line $x = 1$ and D_B the line $y = 2$, their intersection gives the Nash Equilibrium, which is the point $P_N = (1,2)$.

Optimization Results

The aim of this subsection is to show that the analytical equilibria can be found by Pareto and Nash GAs, when x and y take their values in $[-5,5] \times [-5,5]$.

Figure 4 shows the results obtained with a Nash GA. It represents the convergence of the two populations (i.e. the two players) during the optimization process. Player A converges towards 1, whereas Player B converges towards 9. Those results agree with the analytical results, since the figure shows the convergence on the criteria space and not on (x,y). We have $f_A(1,2) = 1$ and $f_B(1,2) = 9$.It follows that $(1,9)$ can be written $(f_A(1,2), f_B(1,2))$, which corresponds to the analytical Nash Equilibrium $P_N = (1,2)$.

Figure 5 shows the result obtained with a Pareto GA. It also offers a comparison with the Nash Equilibrium. Let (C_x, C_y) the coordinates of a point of the Pareto Equilibrium. There is a bijection between each (C_x, C_y) in the criteria space and the points of the line defined by $P_0 = (4,2)$ and $P_1 = (1,3)$ in the search space.

Pareto and Nash strategies are used in the sequel for multiobjective optimization in CFD and CEM non-convex problems.

Optimum shape design

RAE2822 Reconstruction Problem

We have applied genetic algorithms to the field of airfoil optimum design because the deterministic methods that are widely used in aerodynamics are not really robust towards local optima, even if they converge faster than GAs [10].

The first problem we present is a reconstruction problem: it is an inverse problem that consists in finding the shape (denoted γ) of an airfoil which realizes a surfacic target pressure distribution for a given Euler flow condition. This problem can be seen as the minimization of:

$$J(\gamma) = \frac{1}{2}\int_{\gamma} |p^{\gamma} - p^{target}|^2 d\gamma$$

Where p^{target} is a given target pressure and p^{γ} is the actual flow pressure on γ.

We first compute the pressure distribution for a given shape thanks to a CFD solver[2]. Then, we start the optimization process with another shape (either a shape corresponding to a given starting point or a randomly generated shape) and try to retrieve the first shape. Let n be the number of discretization points of the profile. The following discretized cost function f_n (fitness) is used:

$$f_n(\gamma) = \frac{1}{2}\sum_{i=1}^{n}(P_i^{\gamma} - P_i^{target})^2$$

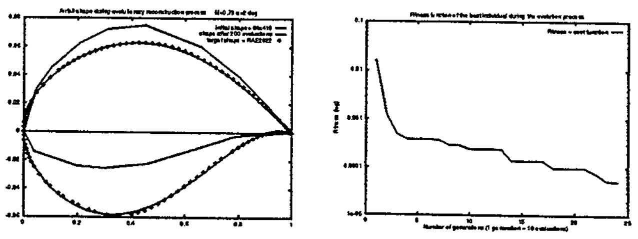
Where P_{γ} is the pressure of the evaluated shape via an Euler flow analysis solver and P^{target} denotes the pressure distribution of the target shape.

We use a seven-order Bezier Spline representation, which corresponds to two fixed points (one at each extremity) and six control points. A Bezier spline $Q(t)$ of order n is defined by the Bernstein polynomes $B_{n,i}$: $Q(t) = \sum_{i=0}^{n} B_{n,i}P_i$ with $B_{n,i} = C_n^i t^i (1-t)^{n-i}$. P_i are the coordinates of the control points, $t \in [0,1]$, and $C_n^i = \frac{n!}{i!.(n-i)!}$.

We define a Bezier spline for the leeward side and another Bezier spline for the windward side. The whole airfoil shape is then defined by the merging of the two splines. For the leeward side, the Bezier spline is completely determined by the coordinates $P_i = (x_i, y_i)$ of the control points. We fix all the x_i of the control points $P_1 \ldots P_6$. The parameters which are optimized are the ordinates y_i of those points. We do the same for the windward side and we obtain 12 real numbers, which are the ordinates of the whole airfoil shape control points. A chromosome C is a vector of $\mathbb{R}^{12}$ whose genes are the ordinates of the control points.

$$C = (\underbrace{y_1, .., y_6,}_{leewardside} \ \underbrace{y_9, .., y_{14}}_{windwardside}), y_i \in \mathbb{R} \cap [\text{Min}_i, \text{Max}_i]$$

Min_i and Max_i are the lower and upper bounds of the variable y_i.



(a) Shape Evolution (b) Convergence

Figure 6: Reconstruction Problem

For the example (Figure 6), the starting shape is a NACA64a410 airfoil and the target shape is the RAE2822. The logarithmic convergence evolution shows that the accuracy of the solution reaches 10^{-4} after only 140 evaluations (14 generations) [13].

The same approach can also be used for shock-drag reduction problems [10]. The viscous effects (coupling of boundary layer with Euler flows) to reduce the viscous drag are under investigation.

High-lift Multi-element Configuration

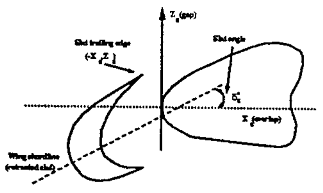
High-lift Multi-element systems consist of leading and trailing edges devices (fig. 7). Leading edges devices increase the maximum lift of an airfoil by delaying its stall angle. Trailing edge devices produce a lift increment. There is an interest to increase the $C_{L_{max}}$ at a fixed approach speed to increase the available payload. Trailing edge devices (flaps) are often designed to produce a lift increment while maintaining a high L/D.

The lift coefficient of such a system is the combination of the lift coefficients of each element, and it takes into account their interaction. The quality of the flow around each element is strongly dependent on their relative position. Thus, optimizing the configuration can be seen as a combinatorial problem. Furthermore, the fitness function variation can be strongly non linear: the separation point on profiles can move rapidly from one position to another due to the wake/boundary layer interaction. The fitness function is non convex with several local optima. This is why we selected a GA based optimization tool.

Problem definition

The physical problem is : Find X^* such that $C_L(X^*) = \max_X C_L(X)$, where X denotes the feasible design variable set.

From the initial configuration, the cruise configuration for transonic flow for example (fig. 8), the high-lift configuration is determined by parameters which represent the cinematic of the moving elements. We define three geometrical design parameters associated to each moving element: δ_x the deflection angle, R_x the overlap, and F_x the gap (fig. 7).


(a) Slat

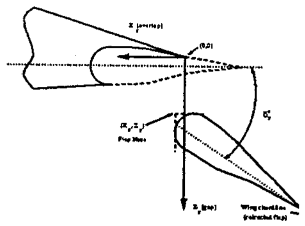

(b) Flap

Figure 7: Multi-element geometry

All geometrical design parameters refer to the fixed main-element position. We considered two kinds of optimization for design variables : only geometrical (for a given angle of attack), or both aerodynamical (angle of attack) and geometrical . In the last case, the set of design parameters is: $\alpha, \delta_s, R_s, F_s, \delta_f, R_f, F_f$. The problem is to find the solution set $(\alpha^*, \delta_s^*, R_s^*, F_s^*, \delta_f^*, R_f^*, F_f^*)$ which maximises C_L.

Flow Solver

The DAMIEN code is a Dassault Aviation in-house two-dimensional solver for multi-element airfoil[5 3]. It has been developed from existing theories coupling inviscid flow[5] by integral methods with viscous laminar and turbulent flows (boundary layers).

The main characteristics of this solver include transition criteria, the treatment of stagnation line for boundary layer initialization, the treatment of separated zones and the wake/boundary

Parameters	$\delta_s, R_s, F_s, \delta_f, R_f, F_f$
Fitness	C_L
Slat variables	δ_s:[28;33], R_s:[-15;-9.5], F_s:[9.5;15]
Flap variables	δ_f:[-36;-28], R_f:[10;18.5], F_f:[9;9.6]
Accuracy	10^{-6}
Pop. size	40
Generations	60

Table 1: Optimization parameters

Slat solution	$\delta_s = 28.16^o$, R_s=-10.03, F_s=9.58
Flap solution	$\delta_f = -35.97^o$, R_f=10.02, F_f=9.08
Fitness	$C_L = 4.84$

Table 2: Optimization results

layer interaction which have been validated with numerous ONERA wind tunnel experiments[3].

The DAMIEN flow solver is not differentiable but is considered as a "black box" for GAs.

Combinatorial Optimisation

A classical binary coded genetic algorithm is used for the combinatorial optimisation problem, with elistist strategy, tournament selection, and a two point crossover.

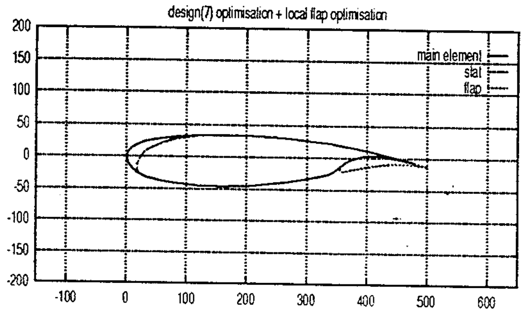


Figure 8: Multi-element Initial configuration

Table 2 presents the results of the GA optimization with 6 parameters (the angle of attack is fixed). Figure 9 shows the corresponding optimal configuration.

The figure 10 shows that the convergence is obtained after 600 evaluations; that is in agreement with the design parameter histories. The history of slat parameters presents an interesting situation: the GA escapes a local optimum after 15 generations. The C_{Lmax} of the configuration has been increased to 4.84.

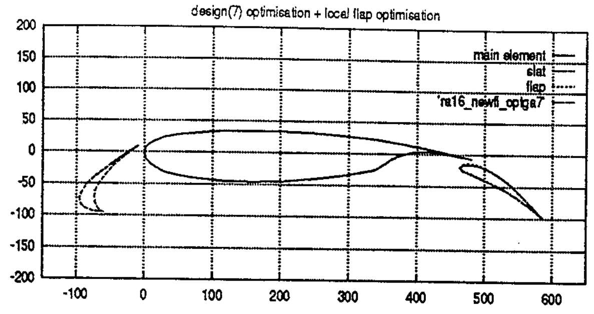


Figure 9: Multi-element Optimal configuration

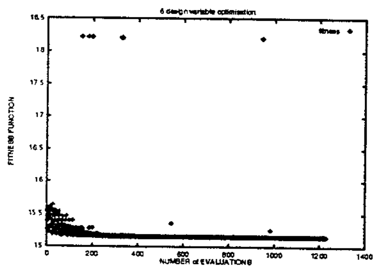
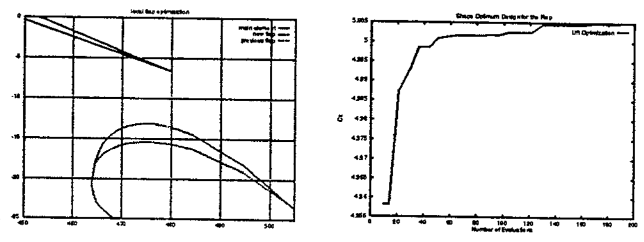


Figure 10: Fitness function History C_L

These 2-D results are destinated to be used in a 3-D analysis based on Weissmeyer's theory applied to determine the lift distribution across the wing with specific treatment of contamination of the separation line.

A further optimization is added by local *shape* flap modification in order to improve the lift coefficient.

Local shape optimisation



(a) Flap Optimal Shape (b) Convergence

Figure 11: Local Shape Optimum Design

A part of the flap, covered by the main element in the transonic regime, is represented by a Bezier spline. The y-coordinates of Bezier control nodes are the design parameters. We treated this optimization task with a real-coded GA (akin the one used for the inverse problem). Figure 11.a shows the optimized shape of the flap. Figure 11.b represents the evolution of the best solution during the optimization process, and shows that the lift coefficient can be improved by local shape variations of the flap.

The lift value has been incremented from a value of 4.84 (the result of the previous combinatorial optimization) to a value of 5.0047.

To meet real design requirements, one should in further analysis study a constrained cost function based on C_{Lmax} but also aim at maintaining a high L/D. Further studies will be undertaken in a near future in that sense.

Multielement Multiobjective optimization

In this section, we apply Pareto GAs and game theory introduced at the beginning of the paper to the solution of the high-lift multi-element optimization. The game involves three players, namely the flap, the main body and the slat. This optimization can be seen as a game where the players try to minimize the local high-lift for each element, which corresponds to a Pareto equilibrium in 3D. Although there are three criteria, we consider only two sets of design variables (for the flap and the slat) since the main body has a fixed position: there is no direct link between the number of criteria and the number of design variables sets. For a given configuration (an individual) the local lift around the three elements can be computed from the flow solver around the global configuration.

Numerical experiments of this cooperative approach show promising results and will be presented during the course of the conference. This method offers a choice of alternative solutions (on the pareto equilibrium) depending on the respective importance given to each element, i.e. in function of changing aerodynamic parameters.

Optimal Backscattering

Problem Formulation

Let us consider a reflecting obstacle R corresponding to an airfoil. Our aim is to minimize the Radar Cross Section (RCS) of R, which corresponds to the backscattered wave. We suppose that K active elements can be distributed among N possible sites on the surface of the airfoil. Each active element generates a wave which is likely to minimize the effects of reflection. Practically, the surface ∂R of the obstacle is divided in N different panels P_j, each one corresponding to an active element which may be active or passive. Due to the linear properties of Maxwell equations, the global electromagnetic field backscattered by the airfoil can be computed by linear combinations of the elementary solutions of Maxwell equations.

Representation

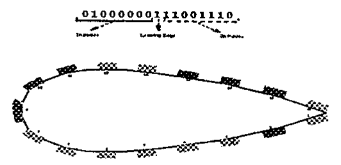


Figure 12: active elements for NACA0012

The problem consists in finding the optimal distribution of K active elements among N possible sites. Since each element can be either active or passive, a straightforward binary representation for the GA can suit very well the problem. For example, if we want to distribute 7 active elements among 17 possible sites, we can code the position of the active elements with a binary string of length 17, with the 0 corresponding to a passive element and the 1 corresponding to an active element. Let $s = x_1 x_2 .. x_{17}$ be such a string. We'll have the following constraints :

$$\sum_{i=1}^{17} x_i = 7 \text{ with } x_i = 0 \text{ } or \text{ } x_i = 1$$

Furthermore, we consider only feasible solutions by applying constraints to both crossover and mutation. The cut sites are selected among the ones that will respect the constraint $\sum_{i=1}^{N} z_i = K$ for the two produced offspring. In the case where the reflector is a NACA0012, figure 12 corresponds to the mapping between string representation 01000000111001110 and the physical location of the active elements [11].

Fitness Evaluation

For a given radar illumination the signature of a reflector can be computed via the solution of time periodic solutions of the Maxwell equations. A solution method based on exact controllability techniques advocated in J.-L. Lions' HUM approach can be found in [1]. Fitness evaluations are then computed through the solutions of the Maxwell Equations.

active elements are characterized by several parameters, among others phases and amplitude, which can be tuned. A trivial approach consists in solving the following embedded minimization problem : $\text{Min}_z \ \text{Min}_{\{A_z, \phi_z, ..\}} J(z, A_z, \phi_z, ..)$, with $A_k, \Phi_k \in I\!R^2$, z a boolean vector of dimension N and :

$$z = \{z_i\}_{i=1..N} \text{ and } \left\{ \begin{array}{l} \forall i = \{1, ..., N\} \ z_i = 0 \text{ or } 1 \\ \sum_{i=1}^{N} z_i = K \end{array} \right.$$

The vector z represents the distribution of the active elements on ∂R. A_z and ϕ_z represent the amplitude and phases of the elements of z.

In order to compute the reflector's RCS, we linearly combine the scattered wave corresponding to elementary solutions of Maxwell equations, where each elementary solutions is associated to an active element. The data needed to compute the RCS, namely the backscattering of the radar illumination and all the elementary solution for each active element, are computed once and for all, and stored. That approach is made possible by the linear properties of Maxwell equations.

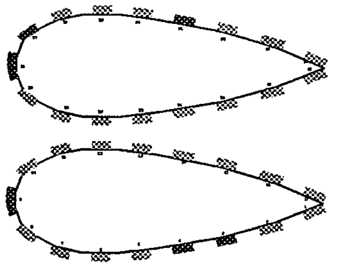
Practically, the surface ∂R of the obstacle is divided in N different panels P_j, each one corresponding to an active element. The contribution of each P_j active element to the global electromagnetic field backscattered by the obstacle R corresponds to the solution of Maxwell equations with the boundary equations:$A_j \theta_j(x) e^{i(\omega t + \phi_j)}$. In this expression A_j denotes the amplitude coefficient, $\theta_j(x)$, $0 \leq \theta \leq 1$ a form function equal to 0 out of P_j, ϕ_j the phases angle, and $\omega = \frac{2\pi}{T}$ with T the period of the radar illumination.

The global scattered field can the be expressed as :

$$U = u_{\text{inc}} + \sum_{i=1}^{N} z_i \ u_{ea_i}(A_i, \ \theta_i, \ \phi_i) \tag{1}$$

It is a non differentiable problem, and its combinatorial properties make it a good candidate for GAs.

Single Objective Optimization


(a) Optimal Distribution

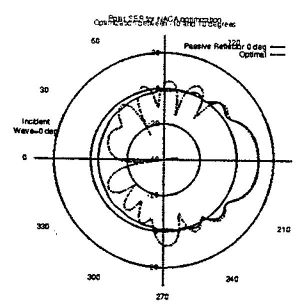
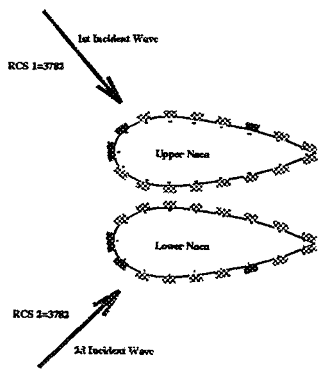
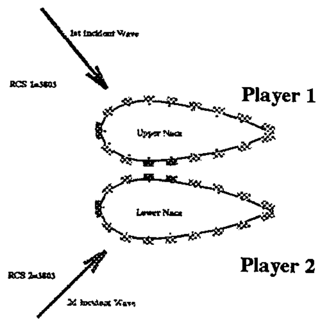

(b) Polar RCS

Figure 13: Optimal Solution, BINACA0012

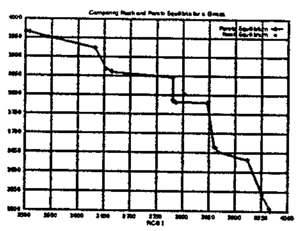


(a) Pareto



(b) Nash

Figure 14: Nash and Pareto Equilibria, BINACA0012


(a) Pareto vs Nash

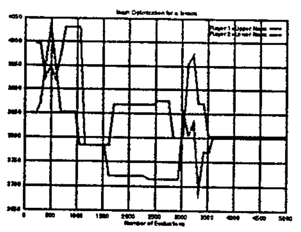

(b) Nash Convergence

Figure 15: Comparison and Nash Convergence

Figure 13 presents the results of a single objective optimization on a BINACA0012. There are 34 potential sites, and 6 active elements to locate. The obstacle is illuminated by a 0^o incident wave and the RCS is minimized in a 10^o sector centered on the incident angle. The GA needs 15000 evaluations to find the optimal distribution (exhaustive search= $C_{34}^{6} = 1344904$). Figure 13 shows that the RCS has dramatically decreased in $[-10^o, 10^o]$.

Application to Optimal Backscattering

We consider that the BINACA0012 might be illuminated by either a -45^o a $+45^o$ incident waves. The optimization task consists in finding the best distribution in order to cope as well as possible with both cases. The first criteria c^1 is the RCS computed for $\alpha \in [-35^o, -55^o]$. The second criteria c^2 is the RCS computed for $\alpha \in [+35^o, +55^o]$. The problem consists in finding a distribution which is a compromise between both criteria c^1 and c^2.

For the Nash approach applied to the RCS optimization of a BINACA0012, Player 1 is the upper NACA0012 and Player 2 is the lower NACA0012.

The optimal distributions obtained by Pareto and Nash are compared in figure 14 . Since there is a single Nash solution and 23 solutions on the Pareto Front, we have taken the Pareto solution for which $c^1 = c^2$. The comparison shows that the Pareto solution is slightly better than the Nash solution (figure 15).

Pareto GA needs on average 195000 evaluations to converge ($\frac{195000}{23} \simeq 8400$ evaluations for each solution). Nash Equilibrium is reached after only a total number of 4000 evaluations (fig 15.b) on average for both players. The main conclusion is that even if Pareto GAs yields better solutions, Nash GAs are faster. Besides, Nash GAs are more robust towards small variations of the criteria[11].

CONCLUSION

GAs are simple and robust search methods for capturing global solutions of non convex optimization design problems. They surpass classical methods for solving CFD problems with integer, discrete and smooth design variables such as the high lift design of a multi-element configuration. The cooperative Pareto or competitive Nash GAs are capable of solving distributed multi criteria optimisation problems such as the RCS minimisation of scattered fields with active aerodynamic reflectors in CEM.

The optimization problems presented in this lecture illustrate the robustness of Genetic Algorithms for the design of new aerospace products of increasing complexity.

However many important problems related to fast computation of fitness functions, flexible parametrization of design variables, complicated physical modelling are still open.

Another promising feature of GAs merged with Game Theory is the development of a new design concept based on distributed optimization implemented in a parallel environment.

ACKNOWLEDGMENTS

This work is supported in part by the European Commission through the DECISION IT ESPRIT project. The authors thank G. Auchmuty, M.O Bristeau, J.-C. Courty, R. Glowinski, G. Leflour, J.L. Lions, O. Pironneau for many theoretical and technical discussions related to the multidisplinary fields of this paper, namely applied mathematics, numerical analysis, computer science, game theory and aerodynamics.

1. M. O. Bristeau, R. Glowinski, and J. Periaux. Using exact controlability to solve the helmholtz equation at high wave numbers. In R. Kleinmann, T. Angell, D. Colton, J. Santosa, and I. Strakgold, editors, *Mathematical and Numerical Aspects of wave propagation*, pages 113–122, Philadelphia, 1993. SIAM.
2. F. Chalot, M. Mallet, and M. Ravachol. A comprehensive finite element navier-stokes solver for low and high-speed aircraft design. In *32nd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, 1994. AIAA.
3. J. C. Courty. Problemes de confluence sillage couche limite a l'aide d'un modele elabore de turbulence. In *13eme colloque d'Aerodynamique Appliquee*, Ecole Centrale de Lyon, 1976.
4. D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, Mass., 1989.
5. J. L. Hess and A. M. O. Smith. Calculation of potential flow about arbitrary bodies. *Progress in Aeronatical Sciences*, 8, 1967.
6. J. H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, 1975.
7. Z. Michalewicz. *Genetic algorithms + data structures = evolution programs.* Artificial Intelligence. Springer-Verlag, New York, 1992.
8. J. F. Nash. Noncooperative games. *Annals of Mathematics*, 54:289, 1951.
9. V. Pareto. *Cours d'Economie Politique.* Rouge, Lausanne, Switzerland, 1896.
10. J. Periaux, B. Mantel, M. Sefrioui, B. Stoufflet, J.-A. Desideri, S. Lanteri, and N. Marco. Evolutionary computational methods for complex design in aerodynamics. In *36th American Institute of Aeronautics and Astronautics Conference*, Reno, January 1997. AIAA-98-0222.
11. J. Periaux, M. Sefrioui, and B. Mantel. GA multiple objective optimization strategies for electromagnetic backscattering. In D. Quagliarella, J. Periaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, Trieste, Italy, November 1997. EuroGen 97, John Wiley.
12. M. Sefrioui. *Algorithmes Evolutionnaires pour le calcul scientifique. Application à la mécanique des fluides et à l'electromagnetisme.* PhD thesis, Université Pierre et Marie Curie, Paris, Avril 1998.
13. M. Sefrioui, J. Periaux, and J.-G. Ganascia. Fast convergence thanks to diversity. In L. J. Fogel, P. J. Angeline, and T. Bäck, editors, *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, San Diego, CA, Feb 29 1996. IEEE Computer Society Press, MIT press.
14. J. M. Smith. *Evolution and the Theory of Games.* Cambridge Univ Press, 1982.
15. N. Srinivas and K. Deb. Multiobjective optimisation using non-dominated sorting in genetic algorithms. In *Evolutionary Computation 2 (3)*, pages 221–248, 1995.
16. J. von Neumann. Zur theorie der gesellschaftsspiele. *Math .Ann.*, 100:295–320, 1928.