A Domain Decomposition/Nash Equilibrium Methodology for the Solution of Direct and Inverse Problems in Fluid Dynamics with Evolutionary Algorithms

Hong Quan Chen[1], Roland Glowinski[2] and Jacques Périaux[3]

[1] Nanjing University of Aeronautics and Astronautics, Department of Aerodynamics, Nanjing 210016, P.R. China, `hqchenam@nuaa.edu.cn`
[2] University of Houston, Dept. of Math., Houston, Texas 77004- 3308, USA, `Roland@math.uh.edu`
[3] University of Jyväskylä, Dept. of Mathematical Information Technologies, Jyväskylä, Finland, `jperiaux@gmail.com`

Summary. The main goal of this paper is to present the application of a decentralization optimization principle from Game Theory to the solution of direct and inverse problems in Fluid Dynamics. It is shown in particular that multicriteria optimization methods "à la Nash" combine ideally with domain decomposition methods, with or without overlapping in order to solve complex problems. The resulting methodology is flexible and in the case of design problems has shown to perform well when using adjoint based techniques or evolutionary algorithms for the optimization.

The above methodology is applied to the simulation and shape design optimization for flows in nozzles and around aerodynamical shapes.The results of various numerical experiments show the efficiency of the method presented here.

1 Introduction

In this paper we introduce a new methodology to solve inverse problems in Fluid Dynamics using Genetic Algorithms and Game Theory. This methodology amounts to finding (suitable) Nash points for "local inverse problems". These Nash points are approximated by Genetic Algorithms (GAs) suitably constructed. This is an example of a completely general method, presented in [7] and [4]. GAs are different from traditional optimization tools and based on digital imitation of biological evolution. Game Theory replaces here a global optimization problem by a non-cooperative game based on Nash equilibrium with several players solving local constrained sub-optimization tasks. The main idea developed here is to consider two Nash applications of Game Theory under conflict introduced in a flow analysis solver (1) and a GAs optimizer (2) as follows:

(1) a flow analysis solver modeled by the potential equations uses overlapping domain decomposition methods (DDM). A variant of the classical DDM Schwarz method is considered with optimal control/GAs techniques. It uses the distance of local solutions on the overlapping regions as global fitness function described in a previous paper with GAs [11]. Then a Nash/GAs game whose decentralized players are in charge of the matching of local solutions as multi fitness functions is associated to the global problem. During the evolution process the search space of each genetic point at the interfaces of overlapping domain is implemented on adapted interval. This new approach is shown to request less information for convergence than the global one.

(2) the above DDM flow solver is then used to feed a Nash/GAs optimizer for the surface pressure reconstruction of nozzle shapes parameterized with local Bézier's splines. During this Nash iteration, the information exchange between DDM flow solver is nested to the shape-GAs optimizer.

Numerical experiments presented on inverse problems of a nozzle with Laplace's solver illustrate both the efficiency and robustness of decentralized optimization strategies. The promising inherent parallel properties of Nash games implemented with GAs on distributed computers and their possible further extensions to non-linear flows are also discussed.

2 Nash and GAs

2.1 Generalities

Many multi objective optimization problems are still not solved perfectly and some are found to be difficult to solve using traditional weighted objective techniques [17, 6]. GAs have been shown to be both global and robust over a broad spectrum of problems. Shaffer was the first to propose a genetic algorithm approach for multi objectives through his Vector Evaluated Genetic Algorithms (VEGA [15]), but it was biased towards the extreme of each objective. Goldberg proposed a solution to this particular problem with both non dominance Pareto-ranking and sharing, in order to distribute the solutions over the entire Pareto front [5]. This cooperative approach was further developed in [16], and lead to many applications [14]. All of these approaches are based on Pareto ranking and use either sharing or mating restrictions to ensure diversity; a good overview can be found in [3]. Another non cooperative approach with the notion of player has been introduced by J. Nash [10] in the early 50' for multi objective optimization problems originating from Game Theory and Economics. The following section is devoted to an original non cooperative multi objective algorithm, which is based on Nash equilibria.

2.2 Definition of a Nash Equilibrium

For an optimization problem with G objectives, a Nash strategy consists in having G 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.* Let E be the search space for the first criterion and F the search space for the second criterion. A strategy pair $(\overline{x}, \overline{y}) \in E \times F$ is said to be a Nash equilibrium iff:

$$f_E(\overline{x}, \overline{y}) = \inf_{x \in E} f_E(x, \overline{y})$$
$$f_F(\overline{x}, \overline{y}) = \inf_{y \in F} f_F(\overline{x}, y)$$

It may also be defined by:
$u = (u_1, \ldots, u_G)$ is a Nash equilibrium iff: $\forall i, \forall v_i$

$$J_i(u_1, .., u_{i-1}, u_i, u_{i+1}, ..u_G) \leq J_i(u_1, .., u_{i-1}, v_i, u_{i+1}, ..u_G)$$

It may be difficult to exhibit such an equilibrium in particular for non differentiable problems.

2.3 Description of a Nash/GAs

The following stage consists in merging GAs and Nash strategy in order to make the genetic algorithm *build* the Nash Equilibrium for a complete description (see [13, 18]).

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. The first idea is to assign the optimization task of X to a player called *Player 1* and the optimization task of Y to *Player 2*. Thus, as advocated by Nash theory, *Player 1* optimizes s with respect to the first criterion by modifying X, while Y is fixed by *Player 2*. Symmetrically, *Player 2* optimizes s with respect to the second criterion by modifying Y while X is fixed by *Player 1* (see [13] for details).

The next step consists in creating two different populations, one for each player. *Player 1*'s optimization task is performed by population 1 whereas *Player 2*'s optimization task is performed by population 2.

Let X_{k-1} be the best value found by *Player 1* at generation $k-1$, and Y_{k-1} the best value found by *Player 2* at generation $k-1$. At generation k, *Player 1* optimizes X_k while using Y_{k-1} in order to evaluate s (in this case, $s = X_k Y_{k-1}$). At the same time, *Player 2* optimizes Y_k while using X_{k-1} ($s = X_{k-1} Y_k$). After the optimization process, *Player 1* sends the best value X_k to *Player 2* who will use it at generation $k+1$. Similarly, *Player 2* sends the best value Y_k to *Player 1* who will use it at generation $k+1$. Nash equilibrium is reached when neither *Player 1* nor *Player 2* can further improve their criteria.

This setting may seem to be similar to that of Island Models in Parallel Genetic Algorithms (PGA [9]). However, there is a fundamental difference

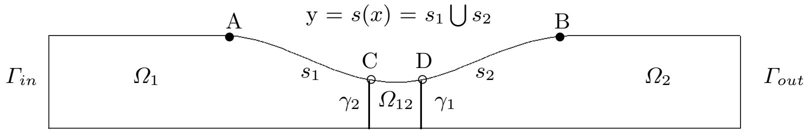

Fig. 1. Description of a nozzle with two subdomains

which lays in the notion of equilibrium for Nash approach. Nash equilibria do not correspond only to robust convergence, but have also very good stability properties compared to cooperative strategies. The mechanisms of the Nash-GAs described here are directly used in the following sections.

3 An Implementation of Nash/GAs Game for the DDM Flow Problem

3.1 Description of the DDM Flow Problem

The DDM optimization problem considered here concerns an incompressible potential flow in a nozzle modeled by the Laplace equation with Dirichlet boundary conditions at the entrance and exit and homogeneous Neumann conditions on the walls. As shown in Fig. 1, the computational domain Ω is decomposed into two subdomains Ω_1 and Ω_2 with overlapping Ω_{12} whose interfaces are denoted by γ_1 and γ_2. We shall prescribe potential values, g_1 on γ_1 and g_2 on γ_2, as extra Dirichlet boundary conditions in order to obtain potential solutions Φ in each subdomain. Using domain decomposition techniques, the problem of the flow can be reduced to minimize the following functional [2]:

$$JF(g_1, g_2) = \frac{1}{2} \parallel \Phi_1(g_1) - \Phi_2(g_2) \parallel^2 \qquad (1)$$

where Φ_1 and Φ_2 are the solutions in the overlapping subdomain Ω_{12}, $\parallel \cdot \parallel$ denotes an appropriate norm, whose choice will be made precise in the examples which follow.

For the minimization problem (1), we have presented a variant of the classical DDM Schwarz method with optimal control/GAs techniques [11] and have made a further extension with genetic treatment at the interface of the subdomains (for details, see [12]). In the following sections, an implementation of Nash/GAs with decentralized players will be addressed.

3.2 Decentralized Multi-Fitness Functions

As mentioned above, in the previous work of references [11, 12], the global fitness function used in GAs is the distance of local solutions on the overlapping domains (see (1)), which could be

$$JF(g_1, g_2) = \frac{1}{2}\int_{\Omega_{12}} |\Phi_1(g_1) - \Phi_2(g_2)|^2 d\Omega\,. \quad (2)$$

In this paper, we use boundary integrals instead of the domain integral and we choose for (2) the criteria introduced in (3). The minimization problem (1) can be reduced to minimize the following function based on boundary integral:

$$JFB(g_1, g_2) = \frac{1}{2}\int_{\gamma_1} |\Phi_1(g_1) - \Phi_2(g_2)|^2 d\gamma_1 + \frac{1}{2}\int_{\gamma_2} |\Phi_1(g_1) - \Phi_2(g_2)|^2 d\gamma_2. \quad (3)$$

Being associated to the global fitness function $JFB(g_1, g_2)$, the decentralized multi fitness functions $JFB_1(g_1, \underline{g_2})$ and $JFB_2(\underline{g_1}, g_2)$ are defined with the following two minimizations:

$$\inf_{g_1} JFB_1(g_1, \underline{g_2}) \quad \text{with} \quad JFB_1(g_1, \underline{g_2}) = \frac{1}{2}\int_{\gamma_1} |\Phi_1(g_1) - \Phi_2(g_2)|^2 d\gamma_1,$$
$$\inf_{g_2} JFB_2(\underline{g_1}, g_2) \quad \text{with} \quad JFB_2(\underline{g_1}, g_2) = \frac{1}{2}\int_{\gamma_2} |\Phi_1(g_1) - \Phi_2(g_2)|^2 d\gamma_2. \quad (4)$$

The inf of the functionals (2) or (3) is zero. Therefore if in searching for a Nash equilibrium (4) we find one such that $\inf_{g_1} = 0$ and $\inf_{g_2} = 0$ then it is the solution of inf (3). There could be other Nash points which would not solve the problem if $\inf_{g_1} > 0$ for instance. The global DDM solution can be found through searching a Nash equilibrium between the above two minimizations based on the treatments described in the next sections.

3.3 An Implementation of Nash/GAs Game

Following the description of section 2.3, we can simulate the DDM flow optimization problem as a Nash game with two decentralized players, *Flow-GA1* and *Flow-GA2* in charge of objective functions $JFB_1(g_1, \underline{g_2})$ and $JFB_2(\underline{g_1}, g_2)$, respectively. Note that each player optimizes the corresponding objective function with respect to non-underlined variables. After discretization of the problem, we have $g_1 = g_{1i}$ and $g_2 = g_{2i}$, $i = 1, ny$ (ny is mesh size in y direction). Following the genetic treatment at the interface of reference [12], for each interface, one point is binary encoded (for instance, g_{11} for γ_1 and g_{21} for γ_2). Other values of g_{1i} and g_{2i} ($i \geq 2$) are corrected by numerical values (for details, see [12]). The whole structure of the implementation based on the information exchange between players is described as follows:

Step 0: (Initialization) Given initial interval (g_{min}, g_{max}) as search space for two genetic points, g_{11} and g_{21}, and then start with two set of randomly created genetic points to form two initial populations for each players, *Flow-GA1* and *Flow-GA2*.

Step 1: *Flow-GA1* and *Flow-GA2* run separately until the iteration number equals the exchange frequency number.

Step 2: Exchange current the fittest flow information between *Flow-GA1* and *Flow-GA2*.

Step 3: Repeat the Step 1 to Step 2 until no player can further improve his fitness.

It should be noted that *Flow-GA1* operates for the left part and *Flow-GA2* for the right part of the nozzle. In fact, we have prescribed $\delta^0 = \frac{1}{2}(g_{max} - g_{min})$ in the initialization step. In this paper, *Flow-GA1* updates $\underline{g_2}$ from the fittest individual of *Flow-GA2*. Besides, the search space of g_{11} is adapted with:

$$(\underline{g_{21}} - \delta^n, \underline{g_{21}} + \delta^n)$$

where $\underline{g_{21}}$ is the first component of vector $\underline{g_2}$ updated by other player, *Flow-GA2*, through Nash-exchange and $\delta^n = fa\delta^{n-1}$, where $fa < 1$. In other words, δ^n is adapted and gradually approached to a small value with the Nash generation, which can ensure accuracy similar to a real value encoding. Numerical experiments have shown that this treatment is helpful for the present method to have the Nash equilibrium. In the meantime *Flow-GA2* player is doing the same as *Flow-GA1* player.

The significant extent of parallelism properties gained from the above method has further improvement compared with previous work of reference [11] or other flow solvers using Domain Decomposition techniques. This DDM flow solver will be used to feed a Nash/GAs shape optimizer described in the following section.

4 Shape Optimization Problem Using Nash/GAs with DDM

The DDM shape optimization problem considered here involves the inverse problem of a nozzle using a reconstruction technique and domain decomposition method using Nash/GAs. For the inverse problem, the global shape optimization is to find a shape (denoted, $y = s(x)$, $x \in [A, B]$, see Fig. 1) of a nozzle which realizes a prescribed pressure distribution on its boundary for a given flow condition. This problem has the following formulation:

$$\inf_s JS(s) \quad \text{with} \quad JS(s) = \frac{1}{2}\int_{[AB]} |p_s - p_t|^2 ds \tag{5}$$

where p_t is a given target pressure and p_s the actual flow pressure on the shape s. Let $s_1(x), x \in [A, D]$ and $s_2(x), x \in [C, B]$ be the split shapes, then if $s(x) = s_1(x) \bigcup s_2(x)$, we consider the two following local optimization problems:

$$\begin{aligned} &\inf_{s_1} JS_1(s_1, \underline{s_2}) \quad \text{with} \quad JS_1(s_1, \underline{s_2}) = \frac{1}{2}\int_{[AD]} |p_{s_1} - p_t|^2 ds_1 \\ &\inf_{s_2} JS_2(\underline{s_1}, s_2) \quad \text{with} \quad JS_2(\underline{s_1}, s_2) = \frac{1}{2}\int_{[CB]} |p_{s_2} - p_t|^2 ds_2 \end{aligned} \tag{6}$$

with the constraint that $s_1 = s_2$ on interval C, D. Then $\inf JS_1 = 0$ on s_1 and $\inf JS_2 = 0$ on s_2 is the solution of (6) considered in the sequel. The global shape optimization solution can be found through searching a Nash equilibrium between the above two minimizations. The DDM flow problem described in the section 3 will provide information to the shape optimization problem using Nash strategy.

4.1 Parameterization of the Shape of the Nozzle

Using GAs, the candidate shapes of the inverse problem mentioned above are represented by a Bézier curve of order n, which reads [1]:

$$x(t) = \sum_{i=0}^{n} c_n^i t^i (1-t)^{n-i} x_i, \quad y(t) = \sum_{i=0}^{n} c_n^i t^i (1-t)^{n-i} y_i$$

where $c_n^i = \frac{n!}{i!(n-i)!}$ and (x_i, y_i) are control points of the curve, t is the parameter whose values vary between [0,1]. To limit the size of the search space, we vary the control points only in the y direction with fixed x_i values. $JS(s)$ is used as fitness function and real coding is used for y_i, which forms a string denoted $\{y_0 y_1 y_2 ... y_{n-1} y_n\}$. One site uniform crossover and non-uniform mutation are used in the present work (for details, see the work of Michalewicz [8]). The treatment of continuity between two split shapes mentioned above will be described in the next section.

4.2 Solution Method and Its Implementation

Following the description of section 2.3, we can now play a practical game of this DDM-shape optimization problem with two players, *Shape-GA1* and *Shape-GA2* in charge of objective functions $JS_1(s_1, \underline{s_2})$ and $JS_2(\underline{s_1}, s_2)$, respectively. With DDM, *Shape-GA1* has a follower *Flow-GA1* with objective function $JF_1(g_1, \underline{g_2})$ and *Shape-GA2* has another follower *Flow-GA2* with objective function $JF_2(\underline{g_1}, g_2)$. Note that each player or follower optimizes the corresponding objective function with respect to non-underlined variables. The whole structure of the implementation based on the information exchange between players is described as follows:

Step 0: (Initialization) Start with a randomly created shape $s(x), x \in [A, B]$ and split it into two curves s_1 *and* s_2 as starting curves for *Shape-GA1* and *Shape-GA2*

Step 1: *Shape-GA1* and *Shape-GA2* run separately until the iteration number equals the exchange frequency number.

Step 2: Exchange current the fittest shape information between *Shape-GA1* and *Shape-GA2*.

Step 3: Repeat the Step 1 to Step 2 until no player can further improve his fitness.

It is noted that *Shape-GA1* operates for the left part and *Shape-GA2* for the right part of the nozzle. In this paper, *Shape-GA1* receives the y coordinate value and slope of the point D from the fittest curve $\underline{s_2}$ of *Shape-GA2*. This value will be used for the end control point of the Bézier curve of s_1 in *Shape-GA1* for the next step. This treatment ensures continuity and is expected to have smoothness at the overlapping segment $\widehat{CD}$. *Shape-GA2* does the same as *Shape-GA1* meanwhile.

The calculation of each shape fitness requires to solve the flow equations by CFD solvers over the whole domain. Combining DDM with the local geometrical optimization, the flow field can be solved separately by two followers *Flow-GA1* and *Flow-GA2* in each subdomain. The *flow-GAs* returns the current fittest flow solution to *Shape-GAs* for computing fitness of each shape and the information exchange between two followers happens during the exchange between the shape players. We are satisfied when each local problem gives "zero" (very small) for the local criteria.

The problem (6) with *Shape-GA1* and *Shape-GA2* is a Nash problem solved with a floating point coded GA, whereas the problem (4) with *Flow-GA1* and *Flow-GA2* is solved with a binary coded Nash GA. Problems (4) and (6) are coupled since a precise solution of the DDM flow solver via (4) is necessary to evaluate candidate solutions of optimization problem via (6).

5 Results and Analysis

With the method presented above, we have tested both the DDM flow problem and the nozzle reconstruction problem, respectively. Exchange frequency for Nash/GAs is 1. The potential values are predicted by a finite element Laplace's solver based on a direct Choleski method. The probability of crossover $Pc = 0.85$ and the probability of mutation $Pm = 0.09$ are not carefully selected but are fixed for *Flow-GAs*. The parameters used in *Shape-GAs* are 0.6 for crossover rate and 0.108 for mutation rate.

We first present the preliminary results of the DDM flow problem with the Nash/GAs game described in the the section 3. The convergence histories of the fittest individual are shown in Fig. 2. Following the trace of the domain integral of the current fittest values of $JF(\underline{g_1}, \underline{g_2})$, we find that the value of the domain integral JF has been reduced from 1.2E-2 to 1.6E-7, which confirms that the present Nash/GAs method works well for the test case.

The numerical results of the method described in the section 4 tested for a nozzle reconstruction problem are presented in Figs. 3-5. As the pressure distribution Cp matches the target, the corresponding nozzle shape is reconstructed successfully (see Fig. 5).

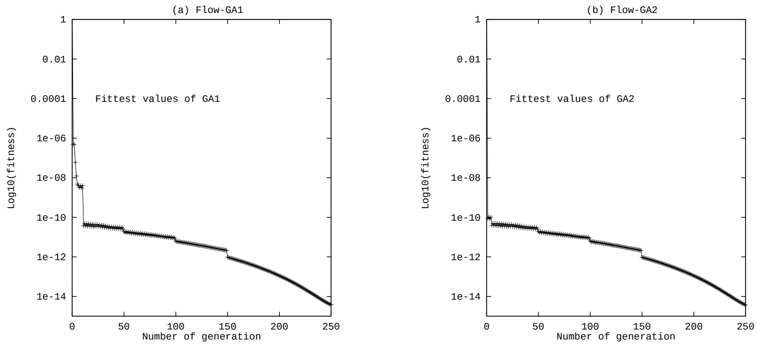

Fig. 2. Convergence histories: (a) Flow-GA1 and (b) Flow-GA2

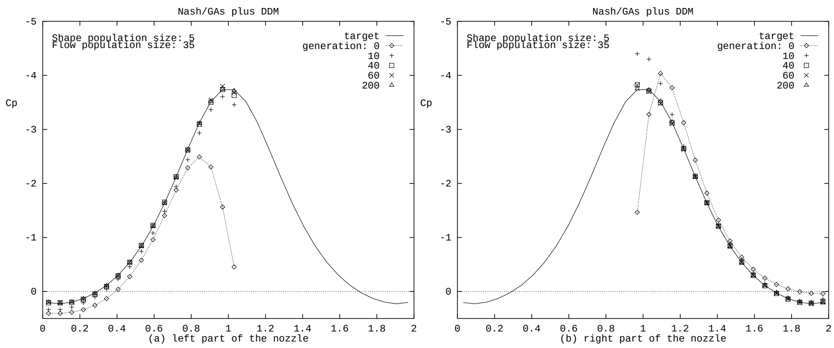

Fig. 3. Successive Cp distributions: (a) left part and (b) right part

6 Conclusion and Possible Extensions

From the experiments described in this paper, it is clear that GAs and DDM may provide robust tools to solve complex distributed optimization problems. It is shown that one can decompose a "global" cost function into a sum of "local" cost functions and under circumstances it is sufficient to look for Nash equilibrium points (or special Nash points). The multi objective techniques with decentralized players discussed here demonstrate convincingly that combining ideas from Economics or Game Theory with GAs may lead to powerful distributed optimization methods for Engineering problems. A significant saving in the above process in terms of elapsed time in a distributed parallel networked environment is anticipated by replacing expensive global commu-

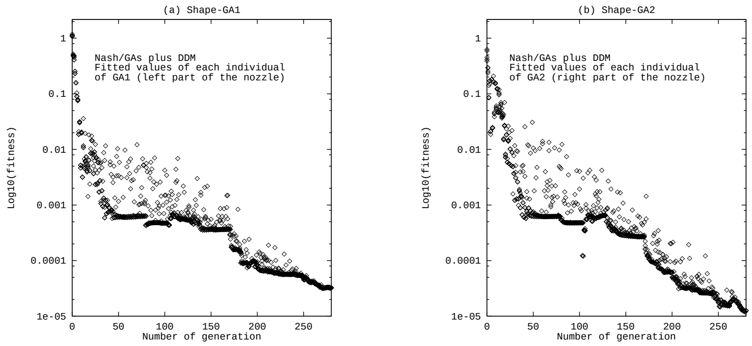

Fig. 4. Convergence histories: (a) Shape-GA1 and (b) Shape-GA2

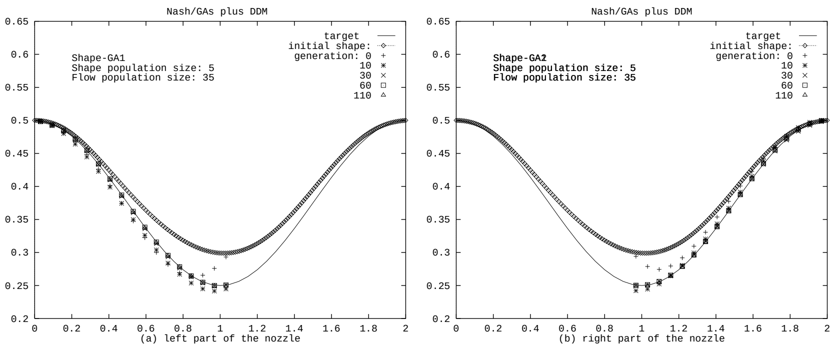

Fig. 5. Successive shapes: (a) left part and (b) right part

nication (standard strong collective optimality) by local communication (non standard weaker individual optimality).

The preliminary results presented above should be checked on many subdomains in dimension 3 and extended to non linear flow situations. Very many other problems can be considered by related methods. Some of them are indicated in the CRAS note by the authors [7] and several papers are in preparation.

Acknowledgement. The authors would like to thank M. Sefrioui, B. Mantel and H. T. Sui for fruitful discussions and providing suggestions and software about Game Theory and GAs for non cooperative optimization strategies, finite element and Bézier spline-parameterization of shapes.

References

[1] J.-A. Desideri. Hierarchical optimum-shape algorithms using embedded Bezier parametrization. In E. Heikkola, Y. Kuznetsov, P. Neittaanmaki, and O. Pironneau, editors, *Numerical Methods for Scientific Computing, Variational Problems and Applications*, pages 45–56. CIMNE Barcelona Publisher, 2003.

[2] Q.V. Dinh, R. Glowinski, J. Périaux, and G. Terrasson. On the coupling of viscous and inviscid models for incompressible fluid flows via domain decomposition. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Paris, 1987)*, pages 350–369. SIAM, Philadelphia, PA, 1988.

[3] C.M. Fonseca and P.J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3:1–16, 1995.

[4] R. Glowinski, J.-L. Lions, and O. Pironneau. Decomposition of energy spaces and applications. *C. R. Acad. Sci. Paris Sér. I Math.*, 329(5):445–452, 1999.

[5] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Mass, 1989.

[6] T. E. Labrujere. Single and two-point airfoil design using Euler and Navier-Stokes equations. Test case TE2 and TE4. In J. Périaux, G. Bugeda, P.K. Chaviaropoulos, K. Giannakoglou, S. Lanteri, and B. Mantel, editors, *Optimum Aerodynamic Design & Parallel Navier-Stokes Computations, ECARP–European Computational Aerodynamics Research Project*, volume 61 of *Notes on Numerical Fluid Mechanics*, pages 214–230. Vieweg, 1998.

[7] J.L. Lions. Private communication, 1999.

[8] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Artificial Intelligence. Springer-Verlag, Berlin, 1992.

[9] H. Muhlenbein, M. Schomisch, and J. Born. The parallel genetic algorithm as function optimizer. *Parallel Computing*, 17:619–632, 1991.

[10] J. Nash. Non-cooperative games. *Ann. of Math. (2)*, 54:286–295, 1951.

[11] J. Périaux and H.Q. Chen. Domain decomposition method using gas for solving transonic aerodynamic problems. In R. Glowinski, J. Périaux, Shi Z.-C., and O.B. Widlund, editors, *Domain Decomposition Methods in Sciences and Engineering. Proceedings of DDM-8, Beijing*, pages 427–431. John Wiley, 1995.

[12] J. Périaux, B. Mantel, and H.Q. Chen. Intelligent interfaces of a Schwarz domain decomposition method via genetic algorithms for solving nonlinear PDEs. Application to transonic flows simulations. In P. Bjørstad et al., editor, *Proceedings of DD9*. John Wiley & Sons Ltd, 1996.

[13] J. Périaux and M. Sefrioui. Genetic algorithms, game theory and hierarchical models: application to CFD and CEM problems. In J. Périaux, G. Degrez, and H. Deconinck, editors, *Genetic Algorithms for Optimiza-*

tion in Aeronautics and Turbomachinery, Lecture Series 2000-07, pages 1–44. Von Karman Institute for Fluid Dynamics, 2000.

[14] C. Poloni. Hybrid QA for multi objective aerodynamic shape optimization. In G. Winter, J. Périaux, and P. Cuesta, editors, *Genetic Algorithms in Engineering and Computer Science*, pages 397–415. John Wiley, 1995.

[15] J.D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In J. J. Grefenstette, editor, *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 93–100, Carnegie-Mellon, Pittsburgh, 1985. Lawrence Erlbaum Associates.

[16] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2:221–248, 1995.

[17] W. Stadler. Fundamentals of multicriteria optimization. In *Multicriteria optimization in engineering and in the sciences*, volume 37 of *Math. Concepts Methods Sci. Engrg.*, pages 1–25. Plenum, New York, 1988.

[18] J.F. Wang and J. Périaux. Search space decomposition of Nash/Stackelberg games using genetic algorithms for multi-point design optimization in aerodynamics. In N. Debit, M. Garbey, R. Hoppe, D. Keyes, Y. Kuznetsov, and J. Périaux, editors, *Proceedings of the 13th Domain Decomposition Methods in Science and Engineering*, pages 139–150. CIMNE Barcelona Publisher, 2002.