69

Intelligent Interfaces of a Schwarz Domain Decomposition Method via Genetic Algorithms for Solving Nonlinear PDEs: Application to Transonic Flows Simulations

Jacques Periaux, Bertrand Mantel and Hong Quan Chen

1 Introduction

Genetic Algorithms (GAs) mimic natural selection based on the Darwinian *Survival of the Fittest* principle. These evolution algorithms inspired from biology share digital information and have been introduced by J.H. Holland [Hol92]. A fitness function is chosen to classify individuals of a population in terms of their adaptation to their environment. Those individuals are candidate solutions of a minimization problem and have a digital DNA representation by binary strings. The robustness of Genetic Algorithms rely essentially on genetic recombination involving selection, crossover, and mutation random operators. They are able to explore very large search spaces to find near-global minima whilst traditional methods with gradient information may fail.

An implementation of GAs for the solution of nonlinear flow problems using domain decomposition methods is presented. It is shown that the conventional Schwarz iterative methods for the matching of overlapped subdomain solutions can be extended to nonlinear situations with a genetic treatment at the interfaces. The fitness function considered in this problem is the distance of local solutions on the overlapping regions. Intelligent interfaces act by disseminating genetic information at one node located on the interface to neighbors for appropriate boundary conditions. Combining the domain decomposition method with the evolution process, converged genetic solutions of transonic shocked flows in nozzles and around lifting multi airfoils are computed successfully.

A particular emphasis is given to the effect of discretization on convergence speed and parallel properties of the evolution method. The available results show that the new method based on local niching strategy has the potential to remove

Ninth International Conference on Domain Decomposition Methods
Editor Petter E. Bjørstad, Magne S. Espedal and David E. Keyes

the dependence of mesh size without a preconditioner and can be also very easily implemented on a distributed parallel computer due to its inherent parallelism.

2 Brief Description of Genetic Algorithms

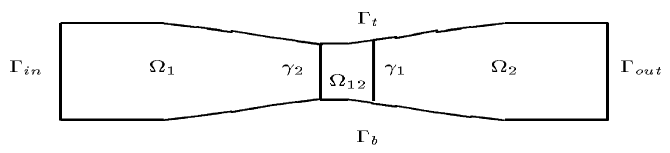
In order to construct implementations of GAs for the solution of nonlinear flow problems using domain decomposition methods, we will briefly restate some of the principle of optimization available in open literature using GAs. Consider a parameter optimization problem of minimizing the cost index $J = f(x)$, where the parameter is x . The first step of optimization process is to code the parameter x as a finite-length string because GAs work with a coding of the parameter set. Here we take a binary string, say of length 8 , such that the lower bound X_{min} for the variable maps to 00000000 and the upper bound X_{max} maps to 11111111 , with a linear mapping in between. The length of string is governed by the required accuracy of the solution. Keep in mind that in general, the string might represent a possible solution to the problem parameterized.

With the described coding, a population of randomly generated strings, say of population size N, would be used as a starting point of the GAs. The strings are then decoded and evaluated to obtain a quantitative measure of how well (fitness values) they behave as possible problem solutions, and transformed to form next generation of size N by using GA-operators. This process continues for successive generations until convergence is achieved or a near optimal solution is found. We can see that the operators play the leading role in the whole processing of the GAs.

There are many existing GA-operators. Among them, three operators: reproduction, crossover, and mutation (which is widely known as the simplest form of a genetic algorithm, namely SGA [Gol89]), are used for many practical uses. Reproduction is a process by which strings with better fitness values receive correspondingly large numbers of copies in the following generation, which is similar to the concept of "survival of the fittest" in the Darwinian theory. The reproduction operator can be implemented artificially in a number of ways, such as Roulette Wheel Selection. In this paper we use Tournament Selection[Gol89] , a simple method of implementing reproduction, to fill up the mating pool where newly reproduced strings are placed and await the action of the other operators.

After reproduction, the operator crossover follows in three steps. First, two newly produced strings are randomly chosen as parents from the mating pool. Second, a position along the two strings is selected uniformly at random. Finally, based on a probability crossover P_c, the paired strings exchange all characters following the crossing site. Clearly, the crossover propagates a structured random information exchange between better fitted parents to produce two offsprings which are expected to combine the better fitted characters of their parents. The operator that merely causes a random alteration of a string position based on probability P_m is called mutation. In present case, this involves changing a bit 1 to a 0 and vice versa. In general, the mutation operator improves the population diversity and prevents the convergence to local minima. For a more thorough discussion of the theoretical foundations of the GAs see the works of J. H. Holland [Hol92] and D. E. Goldberg [Gol89].

Figure 1 Description of a nozzle with two subdomains



3 Implementations of GAs for the Domain Decomposition Problem

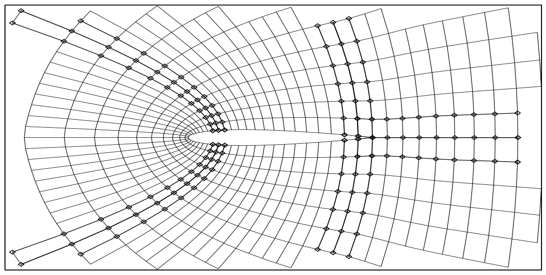
Description of the Problem

The problems of transonic flows in a nozzle and particularly the flows around a lifting airfoil will be investigated in this paper. In order to present general idea of implementations of genetic algorithms for the domain decomposition problem, we will first describe the problem based on the flows in a simple nozzle for sake of simplicity. As shown in Figure 1 , we decompose the computational domain Ω in two subdomains Ω_1 and Ω_2 with overlapping Ω_{12} whose interfaces are denoted by γ_1 and γ_2 . We shall take values, g_1 on γ_1 and g_2 on γ_2, as extra boundary conditions in order to obtain solution in each subdomain. Using domain decomposition techniques, the problem can be reduced to minimize the following function:

$$J(\widetilde{g_1}, g_2) = \frac{1}{2} \parallel \Phi_1 - \Phi_2 \parallel^2, \tag{3.1}$$

where Φ_1 and Φ_2 are the solutions in the overlapping subdomain Ω_{12} , $\parallel \bullet \parallel$ denotes an appropriate norm and $\widetilde{g_1}$ the decoded values of genetic g_1 representation. In the following sections we will present new derivative-free methods of optimization to minimize the above function J based on genetic algorithms.

Figure 2 Subdomains with marked overlappings (mesh size: 87×15)



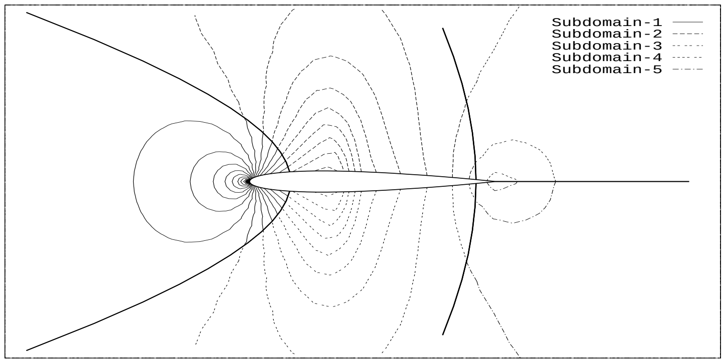
Implementation 1 without niching

For the function being optimized, the variables are rewritten in a code to form a structured string that GAs can directly operate on. For the problem described above, binary codings for multiparameters are used, and we only code g_1 , which can be

$$\widetilde{g_{1i}}; \quad i = 1, N \qquad (N \ \ parameters)$$

each $\widetilde{g_{1i}}$ is coded in l_i bits, thus the length of the string can be $L = \sum_i^n l_i$.

Figure 3 Iso-Mach lines in each subdomain , Mach= 0.8, Attack= 0.0^o



Let us consider population size 25 (i.e. 25 strings). GAs decode each string to return the values of g_{1i} . With g_{1i} known, we can compute the solutions of the domain Ω_1, namely Φ_1. Like Schwarz's alternative method, we take g_2 based on Φ_1, (i.e. $g_2 = \Phi_1 \mid_{\gamma_2})$, thus the solution , Φ_2 , in the domain Ω_2 can be calculated. Now, we send both values in overlapping to cost function:

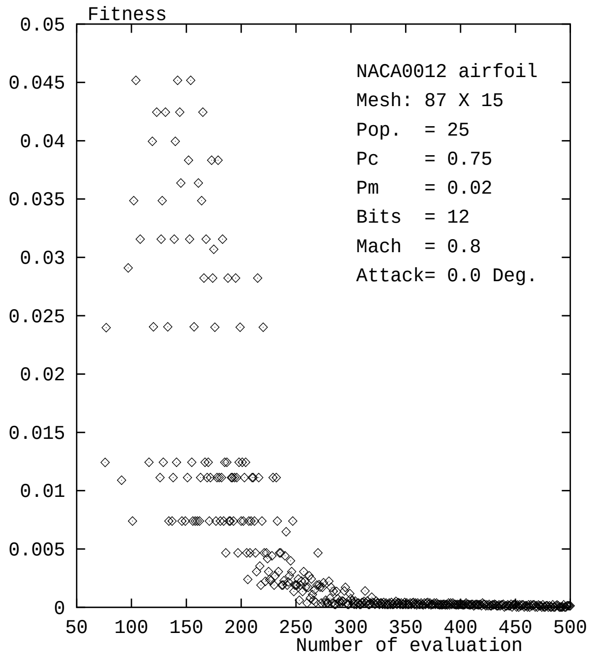
$$J(\widetilde{g_1}) = \frac{1}{2} \parallel \Phi_1 - \Phi_2 \parallel^2 . \tag{3.2}$$

Thus , each string has cost value. New generation of strings can be produced by performing GA-operators and applying survival of the fittest principle.

Implementation 2 with niching

Similar to the Implementation 1 of GAs, we make parameters in each overlapping subdomain belong to one local niche. In each local niche, choose one or several parameters as important parameters. All important parameters are then coded but

Figure 4 Non-lifting case , Mach= 0.8, Attack= 0.0^o



other parameters are nested or coded in a nested way. This could mean that N parameters (g_{1i} , i=1,N) belong to one local niche. We choose g_{11} as key parameter, which is coded in l bits, and other parameters are nested to the decoded value, $\widetilde{g_{11}}$. The nested values can be calculated

$$\Delta_i = g_{1i} - \widetilde{g_{11}} \tag{3.3}$$

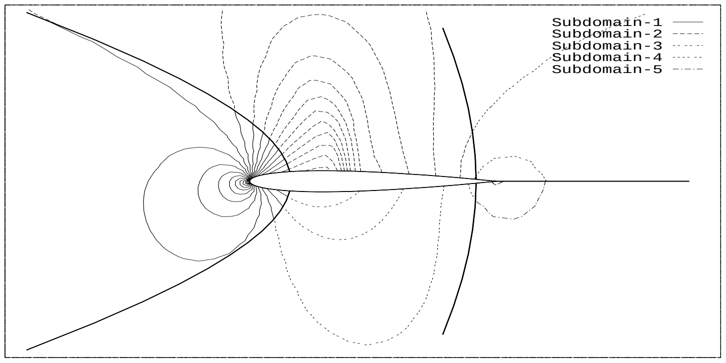
each Δ_i can be coded now in l_i bits. Keep in mind that $\Delta_i(i = 2, N)$ are nested values and the length of $l_i(i \geq\ 2)$ can be short as compared with that for key parameter. This means that we can use local niching strategy to keep the proper representation space, which may reduce the effect of the size of discretization.

In this paper we predict Δ_i based on numerical information, which can be

$$\Delta_i^{n+1} = g_{1i}^n - \widetilde{g_{11}^n} \tag{3.4}$$

where the superscript n is corresponding to the n-th generation and g_{1i}^n are obtained from the previous solution Φ_2 in the domain Ω_2. For simplicity the values of Δ_i can be kept the same in one generation and can be updated with the fittest individual of the previous evaluations.

Figure 5 Iso-Mach lines in each subdomain , Mach= 0.75, Attack= 2.0^o



4 Results and Discussion

The methods presented have been tested for the solution of nonlinear transonic flows in a nozzle with two or three subdomains of two different mesh sizes. Significant improvements of the implementation with niching have been achieved as compared with the method without niching. The numerical results show that the new method based on local niching strategy can accelerate the speed of convergence and has the potential to remove the dependence of mesh size without a preconditioner (see [PC96] and [PMC96] for details).

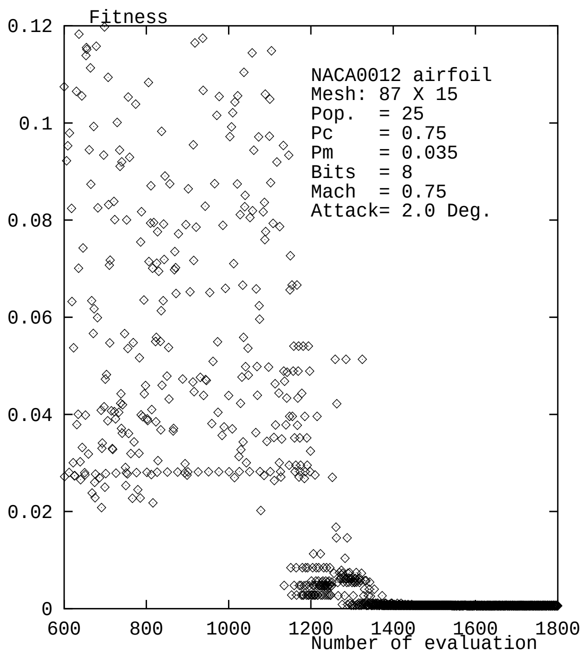
For the computations presented here, transonic flows past a lifting or non-lifting airfoil with multi subdomains are considered as further extension. The extension can be carried out in a straightforward way. The formulae of fitness function can be rewritten as follows:

$$J(\widetilde{g}_1, \widetilde{g}_2, ..., \widetilde{g}_k) = \sum_i^K J(\widetilde{g}_i) \qquad i \ = \ 1 \ , \ K \tag{4.5}$$

where K is the number of subdomain and $\widetilde{g}_i$ are the decoded values of genetic g_i representation on each interface of subdomains.

As shown in Figure 2, we decompose the whole domain in 5 subdomains and hence with 5 marked overlappings, which are placed symmetrically. The C-mesh, consisting of 87×15 was used. This means that on each interface of overlapping we have more than 15 parameters. Based on the local niching strategy, we choose one parameter for each interface as the key parameter. In the present case , 5 key parameters, which are located on the body surface, are to be coded to form a multi parameter string. It should be emphasized that two of these points are located just at the trailing edge, where the cutting lines are located in order to have unique potential. As a result, the circulation can be calculated based on potential values of these two coded points. This means that the lifting circulation is coded implicitly and is fixed for each evaluation, which is different from the traditional method for iteratively determining the circulation. Thus the present case is quite complicated as compared with that used in the nozzle.

Figure 6 lifting case , Mach= 0.75, Attack= 2.0^o



The numerical results for both non-lifting case with Mach= 0.80 and angle of attack $= 0^o$ and lifting case with Mach= 0.75 and angle of attack $= 2.0^o$ are presented in the Figures 3-6. It should be noted that the results of the non-lifting case are obtained by forcing symmetric conditions, such as zero circulation, which results in reducing the search space, thus it appears to have fast convergence as compared with that of lifting case. The iso-Mach lines of figures show the continuity of the solution in the overlapping subdomains. It should be mentioned that the method presented can benefit parallelism from both GAs and domain decomposition methods.

5 Conclusion

The conventional Schwarz iterative method can be extended to nonlinear situations with the genetic treatment on the interface of subdomain without preconditioner and the method presented has the potentiality to avoid the effect of the size of discretization. The solutions of other PDEs using the same concept of GAs are currently underinvestigation.

Acknowledgement

The authors would like to thank Professors, R. Glowinski, Y. Kuznetsov, and O. Pironneau, for the fruitful discussion about the robustness of GAs as preconditioner

companions of classical domain decomposition methods.

REFERENCES

[Gol89] Goldberg D. (1989) *Genetic Algorithms in Search Optimization and Machine Learning.* Addison-Wesley, Reading, Mass.

[Hol92] Holland J. (1992) *Adaptation in natural and artificial systems.* MIT Press.

[PC96] Periaux J. and Chen H. Q. (1996) Domain decomposition method using genetic algorithms for solving transonic aerodynamic problems. In Glowinski R., Périaux J., Shi Z., and Widlund O. (eds) *Proc. Eighth Int. Conf. on Domain Decomposition Decomposition Meths.* Wiley and Sons, Chichester.

[PMC96] Periaux J., Mantel B., and Chen H. Q. (1996) Genetic algorithms applied to domain decomposed flow computations. In Désidéri R.-A., Hirsch C., Tallec P., Pandolfi M., and J.-Périaux (eds) *Computational Fluid Dynamics '96 Proc. Third ECCOMAS Computational Fluid Dynamics Conf.* Wiley and Sons, Paris.