Acceleration of GA in aerodynamics by Search Space Reduction and Artificial Neural

○ルソー・ヤニック，中村佳朗(名大工)

Yannick ROUSSEAU and Yoshiaki NAKAMURA
Dept. of Aerospace Eng., Nagoya University, Nagoya, 464-8603, Japan

Genetic algorithms (GAs) have been successfully applied to numerical optimization problems, unfortunately GAs still remain computationally expensive, and the high computational cost make the use thereof impractical in most Aerodynamic optimization problems. Computational cost reduction is undoubtedly thought to be a common problem of most Aerodynamic shape optimization.

In our approach, Artificial Neural Network (ANN) is used for function approximation; a number of mathematical computations are performed on the approximated function to obtain a reduced model. This resulting mathematical model is used to locate the variables that affect most the cost function.

Our approach is first tested on analytical functions, then numerical experimentations are conducted to solve shape optimization problem for the design of a wing profile. For each evaluation required by the optimizer, the Navier-Stokes equations with the Baldwin-Lomax turbulence model are solved.

1. Introduction

In an optimization problem not all of the variables are of the same importance, or, having advanced in the optimization process, no more all of the design parameters are important.

Therefore, it would be more efficient to focus on the directions of the search space with the highest "pay-off", instead of consuming CPU time to explore directions with a minor impact on the fitness or cost function value.

In the present, Artificial Neural Networks (ANNs) with Radial Basis Function (RBF) are used with Gaussian activation function to approximate the fitness function.

Reduction of Search Space (RSS) is then applied on the ANN response to locate the variables that affect the most the cost function.

Firstly, the present method is applied to widely used GA test functions, then, to drag minimization of a two-dimensional airfoil.

2. Neural Network for function approximation.

Artificial Neural Networks are very sophisticated modeling techniques capable of approximating extremely complex functions. They are particularly well fitted for approximation when we don't have any knowledge about the function a priori. They are applicable in any situation in which a relationship between the inputs (shape parameters) and outputs (cost functions) exists.

The choice of the type of ANN has been guided by its low cost of training. The RBF network (Fig. 1) is a three-layer feed forward network that uses a linear transfer function for the output units and a nonlinear transfer function (Gaussian) for the hidden units.

Gaussian activation function: $\varphi(u,s)=\exp\left(-u^2/2s^2\right)$

with $u=\| x-t \|$ distance of $x=(x_1,\dots,x_m)$ from center vector t、s spread associated.

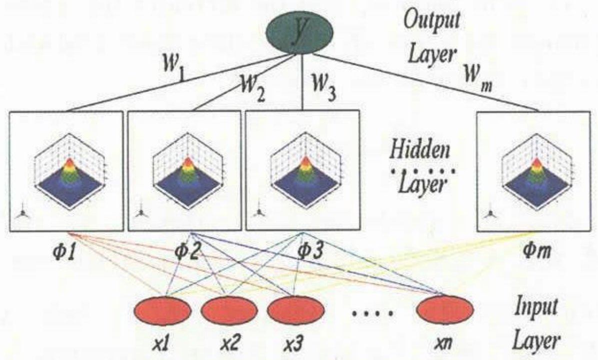

Fig. 1 RBF Neural Network

From the Fig. 1 we can see that the network response to the input x is y:

$$y = w_1\varphi_1(\| x-t_1 \|)+\dots+w_m\varphi_m(\| x-t_m \|) \quad \text{(Eq. 1)}$$

In our approach the ANN is trained by conventional back propagation algorithm using the entire database as centers. The spread is based on Euclidean metrics, weights W are obtained through Least mean Square algorithm using a Quasi-Newton algorithm with BFGS approximation of Hessian matrix [5]. LMS is an iterative procedure; this method of training is robust and provides good results.

3. Reduction of Search Space.

The reduction of the search space is the result of mathematical analysis performed on approximations of the fitness function supplied by Artificial Neural Networks from (Eq. 2).

Suppose a design point $x=(x_1,x_2,\dots,x_n)$ in the Euclidian space E^n, to be optimized with respect to multiple criteria $f: E^n \to \Re$. An important information would be to locate the variables that affect the most f.

The cost function f can be approximated as follows:

$$f(x) = f(x_0) + \nabla f(x_0)^t \delta x + \frac{1}{2}\delta x^t H(x_0)\delta x + \dots \quad \text{(Eq. 2)}$$

Where $\delta x = x - x_0$ is the distance from a given point x_0 in the search space and $H(x_0)$ is Hessian matrix calculated at x_0. After diagonalizing the Hessian matrix in the basis of the eigenvectors G^t of $H(x_0)$, Eq. 2 is reduced to:

$$f(x) = f(x_0) + \sum_{i=1}^{n}\left(\alpha_i \delta\xi_i + \frac{1}{2}\lambda_i \delta\xi_i^2\right) \quad \text{(Eq. 3)}$$

with $\delta\xi = G^t \delta x$

It can be shown [3] that the variables that cause the greatest variation δf_i to f are those associated with the largest values of the quantity:

$$\left|\frac{\alpha_i^2}{\lambda_i}\right| = \xi_i$$

By reordering these values $\xi_1 \le \xi_2 \le \dots \le \xi_p \le \dots \le \xi_n$, optimization can be restricted to the subspace E^{n-p} such that $E^n = E^p \oplus E^{n-p}$ achieving faster convergence.

Having ordered ζ_i, an issue that directly arises is how many of the corresponding parameters to keep for the reduced space optimization. Our approach, which seems to perform well, is to specify a minimum dimension d of the reduced space and additionally keep the parameters, for which the corresponding value ζ_i is of the same order of magnitude as ζ_d

4. Algorithm.

The algorithm can be divided into 3 phases:

- Phase 1: The starting population keeps evolving for a few generations. The genetic operators apply on all the design variables. The evaluated individuals are kept in a database, along with their fitness function values.
- Phase 2: An RBF network is trained, a small percentage, 10%, of the population is moved to the region of the local optimum to enhance exploration (the test cases are updated). The population keeps evolving for a number of generations, with the genetic operators being applied only to the variables identified as the most important. Then, the best individual of the local GA is reinserted randomly into the GA population
- Phase 3: The GA shifts to full optimization, the population keeps evolving for a number of generations, with the genetic operators being applied to all the design variables.
- Phases 2 and 3 are alternated up to convergence.

The algorithm can be summarized by the following flowchart:

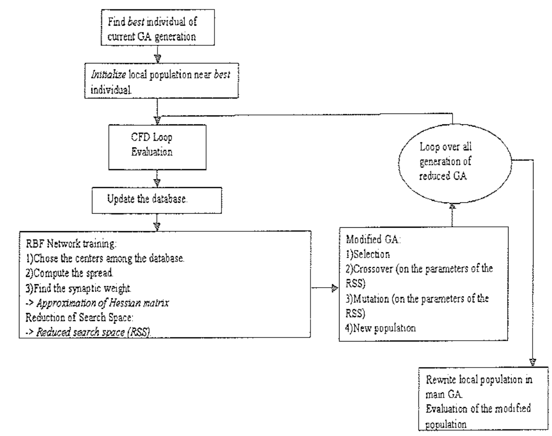

Fig. 2 Flowchart of GA with the Reduced Model

5. Test functions.

GA with Reduced Model algorithm has been tested on Rastrigin and Rosenbrock functions:

$$\textbf{Rastrigin} := 2.n + \sum_{i=1}^{n}\left(0.1x_i^2 - 2\cos(\frac{\pi}{2}x_i)\right) \quad (-5.12 \le x_i < 5.12)$$

$$\textbf{Rosenbrock} := \sum_{i=1}^{n-1}\left(2.(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right) \quad (-5.12 \le x_i < 5.12)$$

Using GA strategies from the table Fig. 3.

Due to the randomness intrinsic of the GA (initial random population, crossover and mutation operators) several runs need to be evaluated and results from Fig. 4-7 show the "best" run of the conventional GA against the "worst" run of the proposed approach.

From Fig. 4-5, unimodal function, we can clearly see the advantages of the algorithm. By doing local search near the best individual of the current GA and by privileging some search directions the convergence is accelerated. Indeed the proposed algorithm act like a "gradient-based" search method, with the particularity that instead of evolving in the descent direction we perform GA on the most "pay-off" variables. Which should not be the case, a priori, of a multimodal function, as we would be trapped into a local optimum.

However, applied to the Rastrigin test function, Fig. 6-7, the algorithm perform relatively well and allow to reach the global minimum in fewer function evaluations compared to the "best" conventional GA. In fact, the dimension of the local search, based on the variance of the individual, is wide enough to move out from the local minimum.

Acceleration of convergence has been verified on test functions, we can apply this methodology to shape optimization problem.

6. Shape parameterization.

In the present study we perform Bezier curve fitting of the set of data points to restrict the design variables to the n Bezier control points. The control points $\left(x^B{}_i, y_i^B\right)$ $i=0, n+1$ of the Bezier curve of degree n that best fit the airfoil $\left(x_j, y_j\right)$, $j=1, Nt$ are determined by minimizing the norm:

$$L_2 = \frac{1}{2}\sum_{1}^{N_t-1}\left(y_B(t)-y_j\right)^2 . \frac{x_{j+1}-x_{j-1}}{2}$$

with t such that $x_B(t)=\sum_0^n B_n^i(t).x^B{}_i$ and the

Bezier parameterization: $\begin{cases} x_B(t)=\sum_0^n B_n^i(t).x^B{}_i \\ y_B(t)=\sum_0^n B_n^i(t).y_i^B \end{cases}$

Since previous work [4] show that cosine distribution of abscises $x^B{}_i$ often used in practice, indeed gives somewhat more accurate results than a uniform distribution and fully optimized distribution of point, we keep the abscises fixed using a cosine distribution. As a consequence, optimization is carried out only on the coordinates y_i^B, $i=0, n+1$.

7. Drag minimization problem.

The methodology has been applied to the drag minimization of the RAE2822 transonic airfoil (*Mach = 0.73;AoA=2.79; Re/m=6.5e5*), parameterized with 8 control points for the upper and lower surfaces (Fig.8).

The RANS equations are solved using a finite volume-cell-centered for the space discretization and a 5th order Runge-Kutta scheme for the time integration. Implicit residual averaging, combination of second and fourth order artificial dissipation, local-time stepping and Baldwin-Lomax turbulence are used, the calculation domain is a C-type grid.

The fitness function is:

$$F\left(Y_1,\ldots,Y_{nparam}\right)=\frac{C_D}{C_{D_o}} \quad nparam=8$$

with C_{D_o} drag of the initial geometry.

Concerning the GA strategy, we used the same strategy as the test function of dimension 8.

After conducting the optimizations, the proposed approach reaches the solution of the conventional GA in only 1550 CFD evaluations and converges more significantly (Fig.9). The pressure-drag is reduced from 0.0135 to 0.012. The corresponding initial and optimized iso-Mach number contours are given on Fig.10 and 11. We can notice that the shock strength has visibly been reduced.

This experiment demonstrates the effectiveness of the proposed approach.

Conclusions

Acceleration of GA by Artificial Neural Network and Reduction of Search Space has been verified. However the results still depends upon many parameters that needed to be tuned by the user (dimension of the local search, when to use the Reduced Model, quality of approximated function, i.e. ANN). Indeed, as the GA evolves, the training of the ANN gets not only more and more time consuming (still reasonable compared to usual CFD calculation), but also it may leads to some discrepancies due to the clustering of the centers (niching). Future investigations need to be done to obtain a self-adaptative and efficient algorithm.

References

[1] D.E. GOLDBERG. *Genetic Algorithms in search, optimization & machine learning.* Addison-Wesley, 1989.
[2] S. HAYKIN. *Neural Networks.* Prentice Hall International, Inc., 2nd edition, 1998.
[3] M. Karakasis and J.-A. Désidéri. *Model reduction and adaption of optimum-shape design in aerodynamics by neural networks.* Rapport de Recherche 4503, INRIA, July 2002.
[4] Zhi Li Tang and Jean-Antoine Désidéri. Towards *Self-Adaptive Parameterization of Bézier Curves for Airfoil Aerodynamic Design.* . Rapport de Recherche 4572, INRIA, Septembre 2002.
[5] LIU, D.C. and NOCEDAL, J. *On the limited memory BFGS method for large scale optimization.* Mathematical Programming 45:503-528. 1989.

Function	Rastrigin		Rosenbrock	
Dimension	2	8	2	8
Population size	10	50	10	50
One point crossover prob.	70%			
Mutation probability	10%	2%	10%	2%
Coding	26 bits			
Generation using Full Model	5	5	5	5
Generation using Reduced Model (RM)	3	5	3	5
High Mutation probability of RM.	90%			
High Crossover probability of RM	33%	20%	33%	20%
Coding	16 bits			
Population of RM	3	5	3	5

Fig. 3 GA strategies for test functions

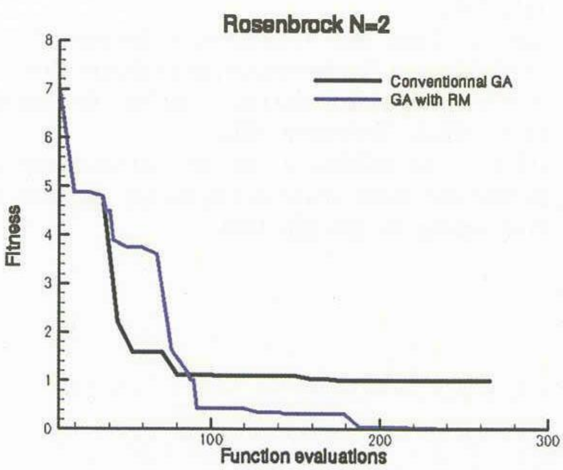

Fig. 4 Convergence history for Rosenbrock N=2

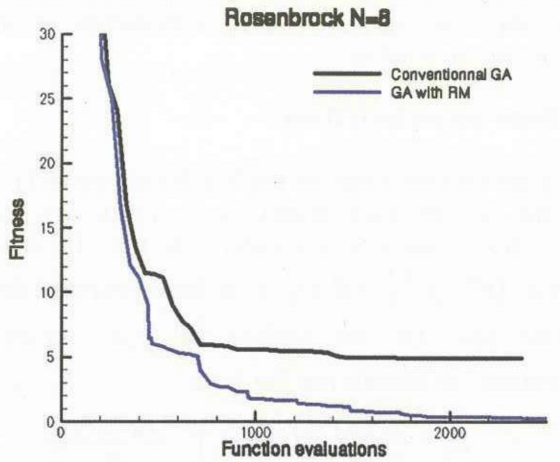

Fig. 5 Convergence history for Rosenbrock N=8

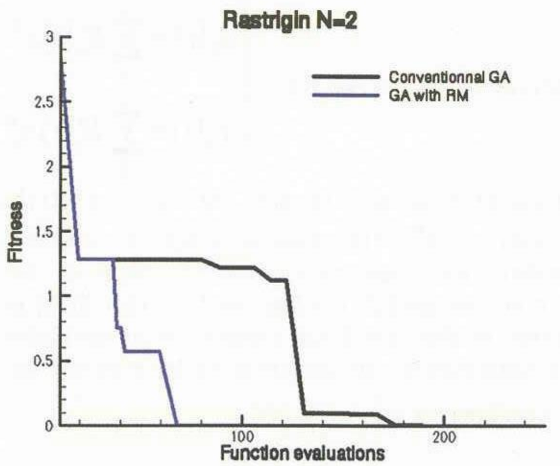

Fig. 6 Convergence history for Rastrigin N=2

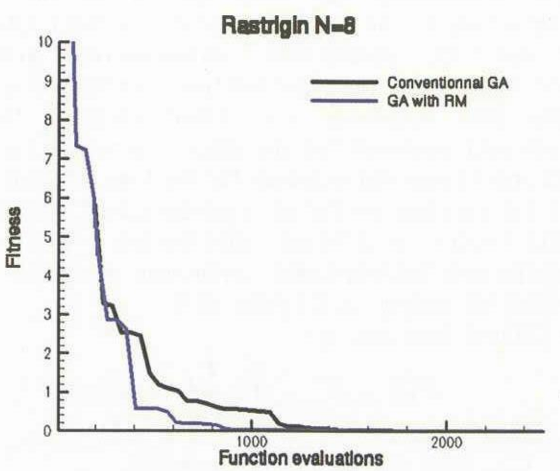

Fig. 7 Convergence history for Rastrigin N=8

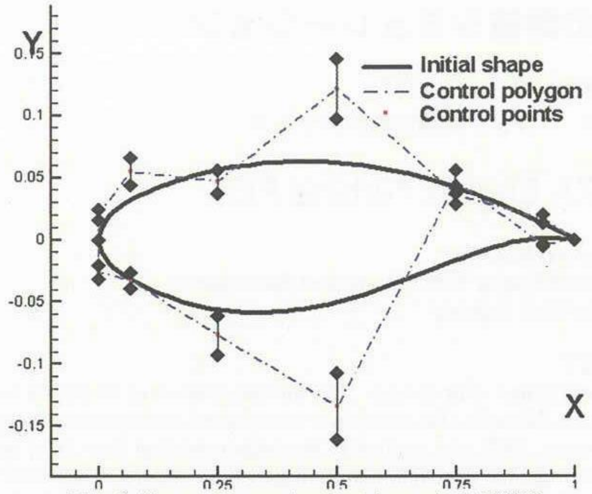

Fig. 8 Shape parameterization of RAE2822

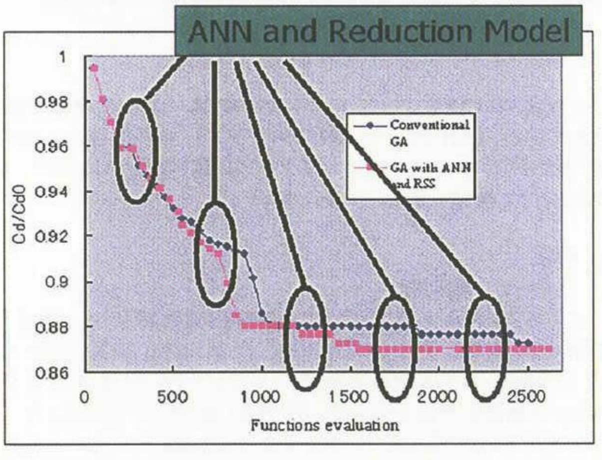

Fig. 9 Convergence History

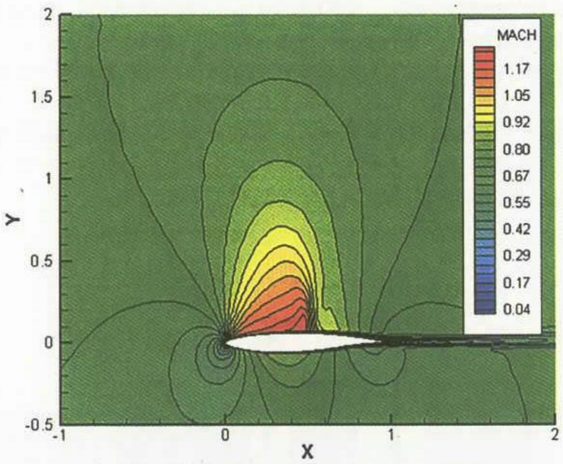

Fig. 10 Initial iso-Mach contour

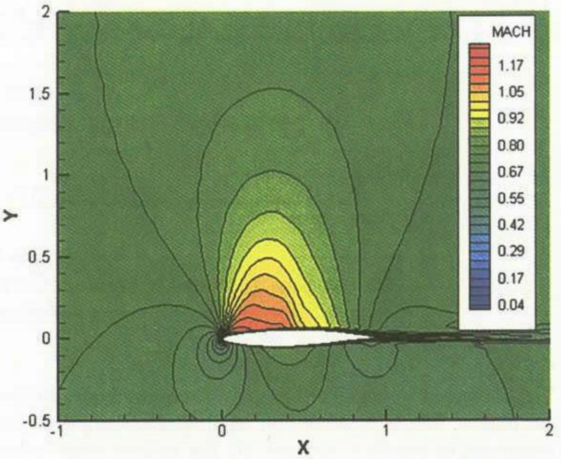

Fig. 11 Iso-Mach contour of the solution