Augmented genetic algorithm with neural network and implementation to airfoil design

Abdurrahman Hacioglu*

Department of Aeronautical Engineering, Turkish Air Force Academy, 34149 Yeşilyurt, İstanbul, Turkey

Abstract

An augmented genetic algorithm with artificial neural network is introduced as a new aerodynamic design and optimization technique. With the purpose of getting a faster algorithm, a neural network and a real coded genetic algorithm are hybridized in a new way. In this way, instead of predicting the computational fluid dynamics calculation of a candidate airfoil, a properly trained neural network is used for predicting the candidate itself. At each step of the genetic process, using the target pressure distribution as an input to the trained neural network produces an airfoil that is a candidate solution of the inverse design problem. The proposed algorithm is tested for the inverse airfoil design problem in the transonic flow case. The results indicate that the computational efficiency of the implemented algorithm is tremendously high.

Keywords: Inverse design; Neural network; Genetic algorithm

1. Introduction

In aerodynamic shape optimization problems, in spite of their robustness, the main disadvantage of genetic algorithms (GAs) is that they require too many computational fluid dynamics (CFD) calculations. That is, using GAs is very time consuming. Generally, there are four ways of accelerating and improving the performance of GAs for aerodynamic shape optimization problems [1]: the use of improved genetic operators; the use of multiprocessing; hybridizing GAs with a numerical optimization method; and reducing the number of calls to the costly direct solver by using inexact evaluations (e.g. using an artificial neural network (ANN) for a certain number of evaluations).

In the last way above, the ANN is used for CFD calculation rather than a costly direct CFD solver. A trained ANN evaluates the aerodynamic performances of some of the aerodynamic shapes in the population during the GA process (a direct CFD solver is used for the remaining aerodynamic shapes in the population). In this manner, the amount of direct CFD solver usage is reduced, and consequently the time consumption of GAs can decrease considerably. To apply this technique, the ANN is trained by using the aerodynamic shapes as the input and their aerodynamic performances as the output. The success of this technique depends on the accuracy of the inexact CFD calculations of the ANN. Additionally, the main characteristic of getting the desired solution depends only on the power of GAs in searching the design space, since the individuals (new candidate solutions) in the new population are produced only by reproduction operations (such as mutation and crossover) at each step of the GA process.

For aerodynamic shape optimization problems, a new way of improving the performance of GAs is suggested in this work. In this way, a new technique of using ANNs and GAs in a hybrid manner is proposed. Instead of inexact CFD evaluations (predicting CFD calculation), the ANN is used to predict a candidate for the desired airfoil at each step of the GA process. This predicted airfoil is added to the new population produced in the genetic manner, so that it can be used at the next step of the GA process. In this case, in addition to GAs, the ANNs also affect the solution procedure because one of the candidate solutions in the new population is produced by the ANN at each step of the GA process. However, this time, the ANN does not need to be accurate in predicting the candidate, since the GA process can explore more fitted individuals from a less fitted candidate, and it can also eliminate defective

* Tel.: +90 (544) 455 0593; Fax: +90 (212) 662 8554; E-mail: hacioglu@hho.edu.tr

Computational Fluid and Solid Mechanics 2005
K.J. Bathe (Editor)

candidates through the selection phase. This new algorithm is called the Augmented Genetic Algorithm with Neural Network (AGANN), and its computational efficiency is tremendously high for airfoil optimization. The GA method used in this work is the Vibrational Genetic Algorithm (VGA), detailed in [2] and [3].

2. Augmentation procedure of genetic algorithm with neural network

In the AGANN, the ANN is used for predicting a candidate airfoil at each step of the GA process. The airfoil geometries and their fitness values (or pressure coefficient, C_p, distributions) in the population at the current step of the GA process are used for training the ANN. The trained ANN produces an airfoil at each step using the target fitness value (or using target C_p distribution) as an input. During the initial stages of the GA process, the response surface obtained from the ANN is not enough to get the desired solution, because the population (the set of training data for the ANN) is probably far from the target. For that reason, the ANN produces an erroneous candidate for the desired airfoil in the initial stages. However, this candidate may be more fitted than the best one produced by the GA process. In this case, the candidate produced by the ANN makes the GA process faster in exploring more fitted individuals. That is, even if the ANN does not produce the desired candidate, it may reinforce the population with a more fitted individual during the GA process. On the other hand, when the GA process continues and the population (the set of training data for the ANN) is getting close to the target, the ANN can produce less erroneous candidates. Briefly, while the ANN reinforces the GA population with a more fitted candidate, the GA can produce a more fitted population by using this candidate, and the more fitted population (set of training data) ensures that the ANN can produce a better prediction. Consequently, this positive interaction makes the genetic process very fast and the desired solution can be obtained quickly.

The main steps of the AGANN are as follows:

1. CFD calculations of the airfoils in the current population are performed to get their fitness values. The selection operation in the current step of the GA process is carried out by using these fitness values, and the other GA operations are performed to get the new population.
2. The ANN is trained by using the airfoil geometries in the current population and their fitness values. For this training, the fitness values of the airfoils are used as the input and the corresponding airfoil geometries are used as the output.
3. The target fitness value is used as an input to get the corresponding airfoil geometry from the trained ANN. This corresponding airfoil is placed in the new population produced by the GA operations, so that it is used as a candidate at the next step of the GA process.

This procedure is repeated at each step of the GA process. The GA process continues until the desired solution is obtained.

The ANN model used in this work is a back-propagation neural network (BPNN) [4]. One non-linear hidden layer with sigmoid (hyperbolic tangent) transfer function and a linear output layer are used in the BPNN. Figure 1 shows the architecture of this BPNN. As

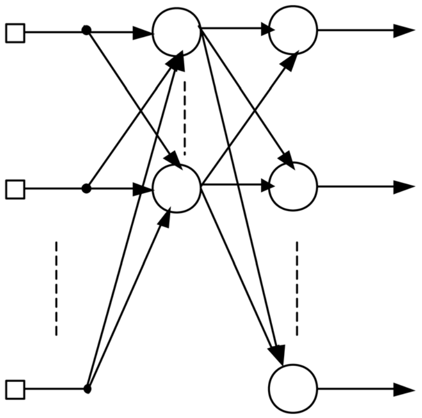

Fig. 1. Architecture of back propagation neural network (BPNN).

mentioned above, in one set of training data, the input parameters are the fitness values of the airfoils in the population, and the output parameters are the airfoil geometries. The training of the ANN is continued for each step of the GA process, by using the airfoils in the current population and their fitness values. The weights of ANN obtained from one set of training data are saved and are used as initial values to train the next set of data. This reduces the ANN training errors for a fixed amount of training epochs and enables a more comprehensive construction of the response surface for the search space.

3. Experiments and results

Experiments were performed for the RAE2822 airfoil.

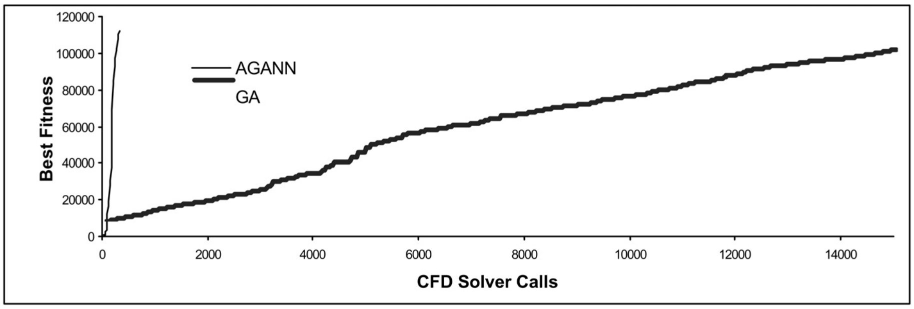

Fig. 2. Iteration history for the experiment.

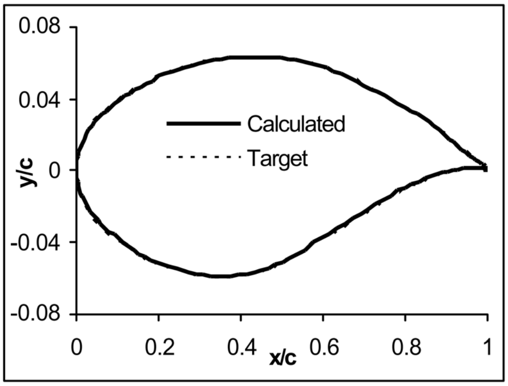

Fig. 3. Calculated and target airfoils.

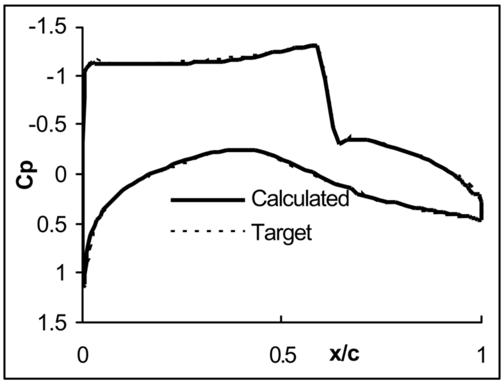

Fig. 4. Calculated and target C_p distributions.

Surface pressure coefficient distribution, C_p, of the RAE2822 airfoil at 2° angle of attack and $M = 0.725$ are given. A full potential flow solver with 161 × 31 O-mesh was used for transonic, inviscid flow. We started from the NACA0012 to reach the desired airfoil. The thickness ratio of NACA0012 was changed ±30% uniformly in the initial population. For the Bezier representations of the airfoil surface, 13 control points were used, always keeping two of them constant.

The results obtained in transonic conditions for the profile RAE2822 are shown in Fig. 2. The line of best fitness value of AGANN is an almost vertical straight line. To obtain a fitness value of 70000 for the GA and AGANN, 10320 and 195 CFD calculations are needed respectively. Comparing these results indicates that the reduction in CFD calculations for AGANN with respect to GA is 98%.

The target profiles, optimized by inverse design, and their pressure coefficients are shown in Figs 3 and 4. It can be seen from Figs 3 and 4 that the proposed algorithm has done a good job in carrying out the inverse design problem.

4. Conclusions

The numerical experiments indicate that the suggested algorithm, AGANN, has a great impact on the reduction in the number of CFD calculations needed for inverse airfoil design in transonic flow conditions. Instead of inexact CFD evaluation, using an ANN for predicting an airfoil (candidate solution of the target pressure distribution) causes a positive interaction between the ANN and GA as explained in Section 2. Consequently, the GA gains great exploration power, since the ANN usually supports the population with the desired solution after only a few generations.

References

[1] Giannakoglou KC. Acceleration GAs using ANN – theoretical background. GAs for Optimization in Aeronautics and Turbomachinery, von Karman Institute for Fluid Dynamics, Lecture Series 2000–07.

[2] Hacioglu A, Özkol İ. Transonic airfoil design and optimisation by using vibrational genetic algorithm. Aircraft Eng Aerospace Tech 2003; 75(4).

[3] Hacioglu A. Using genetic algorithm in aerodynamic design and optimization. Ph.D. thesis, Department of Aeronautical Engineering, Technical University of Istanbul, Turkey, 2003.

[4] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back propagating errors. Nature 1986; 323:533–536.