AERODYNAMIC SHAPE OPTIMIZATION WITH A NEW PARALLEL EVOLUTIONARY ALGORITHM AND NUMERICAL FLOW MODELLING

A. Jahangirian, M. Ebrahimi
Aerospace Engineering Department, Amirkabir University of Technology, Tehran, Iran

Keywords: *Aerodynamics, Optimization, Evolutionary Algorithms, Parallelization, CFD*

Abstract

A hybrid parallel Genetic Algorithm with a new population dispersion method is presented to accelerate aerodynamic shape optimization. The method combines a multi-population and master–slave parallel Genetic Algorithm. To increase the convergence rate of the optimization process a new technique for scattering individuals in the design space is also applied. Based on an online monitoring and by taking feedback from the result of the current population the mutation rate of genes are updated. In order to assess the efficiency of the proposed framework, a geometric inverse design is carried out and the capability of the method for generating flexible shapes is evaluated. An unstructured grid finite volume flow solver with a two-equation k-ε turbulence model is used for the objective function evaluations. The performance of the method is further evaluated by an aerodynamic shape optimization. Result indicates the merits of the framework with increasing the maximum objective value about 3.5 percent as well as decreasing the total computational time up to 30 percent.

1 Introduction

Over the past few years, with the development of mutli-core processors and advance in optimization techniques, numerical methods have played a great role in the design process of aerodynamic shapes. Genetic Algorithms (GA) that is popular evolutionary technique has been widely used by researchers since they are very efficient in finding the global optimum for complex functions [1]. Additionally, it uses only the objective function and does not require its derivatives. Such features make GA attractive to practical engineering applications like aerodynamic shape optimization [2, 3]. However, this optimization tool has the main disadvantage of being computationally time-consuming especially in aerodynamic shape optimization problems where Computational Fluid Dynamics (CFD) is applied for objective function evaluation [4]. Therefore, a leading area of research is to reduce the computational time of optimization problems whereas the high fidelity of the objective function evaluation is kept.

The performance of GAs depends on various factors, such as the population size, the initial population, the selection strategy and crossover and mutation rates. These factors interact with each other which make it hard to find the optimal setup [5]. However mutation which is used to keep the diversity of the population by changing members' characteristics is considered to be one of the most effective of these parameters that significantly affects the general behavior of the optimization algorithm. There has been considerable research to investigate the nature of mutation and its optimum rate [6, 7].

Many studies have suggested different static mutation rate for optimization by evolutionary algorithms. These rates are derived from trial-and-error or by experience. Eiben et al. and Grefenstette proposed mutation rate of 0.05 and 0.01 respectively [8, 9]. Back et al. and Krink et al. carried out comprehensive studies on the effect of mutation rate on optimization process and proposed frameworks to maintaining the diversity that still avoid premature convergence [10, 11]. To treat an optimization problem with a large design space, Adaptive Range GA (ARGA) was proposed by Arakawa and Hagiwara for binary genetic algorithms [12].

The idea behind this method is to adapt the population toward promising design regions according to the distribution of the design variables. It uses the statistics of the top half of the population to adapt the genes in the search space. Hence, the adapted population distributes in the hopeful search region. The method was further extended to real coded applications of single and multiple objective function optimization problems [13]. The main objective of the present study is to develop a parallel GA framework for aerodynamic shape optimization applications that is appropriate and applicatory to the problems with high CPU costs and large memory requirements. Also, an online adaptive mutation rate that consists of two phase of exploration and refinement is introduced which by taking feedback from the relative success or failure of the individuals, increase the convergence rate of the optimization method.

2 Optimization with Genetic Algorithm

Genetic Algorithms are probabilistic methods inspired by the 'survival of the fittest' principle of Darwin's Theory of evolution. Artificial individuals are generated and put into competition for life and only the survivors are transferred to the next steps. A new population will be generated using three operators of crossover, reproduction and mutation and the process repeats until some search termination criteria are satisfied.

In the current study, a real coded GA is applied and chromosomes, genes and fitness are corresponding to the design candidates, design variables and objective function, respectively [14]. According to the nature of the problem and considering the state-of-art, an elitist strategy for the tournament operator is applied, where the two best chromosomes in each generation are transferred into the next generation without any change [15]. The objective function is evaluated using the numerical solution of governing flow equations. Then, the population is optimized according to the objective function value (fitness) through the GA. The crossover operator exchanges the chromosomes of the selected parents, randomly. A simple one-point crossover operator is utilized in this paper with 72% probability of combination, as the use of smaller values was observed to deteriorate the GA performance [15].

For being able to search the whole domain an adaptive mutation rate is applied where probability of mutation is adjusted according to the population diversity. More details about this technique is presented in the following sections.

3 The Flow Analysis Solver

The real cost of evolutionary airfoil shape optimization depends on the number of objective function evaluations using the CFD solver which is determined by the generation number and population size. Therefore it is very important that the CFD solver to possess high efficiency and convergence rate. The aerodynamic evaluation is based on a two-equation k-ε turbulence model that is implemented together with the wall function near wall treatment for computation of Reynolds Averaged Navier–Stokes (RANS) equations [16]. The turbulent flow equations are solved using a finite volume cell-centered implicit scheme that follows the work of Jahangirian and Hadidoolabi on unstructured grids [17]. The governing equations are as following:

$$\frac{\partial Q}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=0 \qquad (1)$$

where Q is the flow variable vector and F and G are the combination of convective and viscous fluxes.

$$F=F^I-F^V,\ G=G^I-G^V \qquad (2)$$

superscripts I and V are used to separate inviscid and viscous terms. In Equation 1 and 2 the Q, F and G are as following:

$$Q=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{pmatrix},\ F^I=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (\rho E+p)u\end{pmatrix},\ G^I=\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (\rho E+p)v\end{pmatrix} \qquad (3)$$

$$F^V=\begin{pmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ u\tau_{xx}+v\tau_{xy}+q_x\end{pmatrix},\ G^V=\begin{pmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ u\tau_{xy}+v\tau_{yy}+q_y\end{pmatrix} \qquad (4)$$

where u, v, p, ρ and E are the velocity components in x and y directions, pressure, density and internal energy of the fluid, respectively. Also heat flux (q) are defined as:

$$q_x = -K\frac{\partial T}{\partial x},\; q_y = -K\frac{\partial T}{\partial y},\; K = {C_p\mu}/{\Pr} \quad (5)$$

where viscosity (μ) is calculated from Sutherland's law and Prandtl number (Pr) is considered 0.9 for turbulence flow. Integrating equation (1) over the control volume, results in the following equations:

$$A\frac{\partial Q}{\partial t} + \sum_{L=1}^{3}(F\Delta y - G\Delta x)_L = 0 \quad (6)$$

where A and L are the area and the side of the triangular cell, respectively. Considering the above equation for each cell, the set of ordinary differential equations is obtained:

$$A_i\frac{d}{dt}(Q_i) + R_i(Q) - D_i(Q) = 0 \quad (7)$$

where A_i is the area of the cell i and $R_i(Q)$ includes the viscous fluxes and convective. To provide numerical stability, the artificial dissipation fluxes $D_i(Q)$ are also added.

To obtain accurate objective function evaluations having suitable computational grid is very essential since the numerical solver performs several hundreds of times in optimization process. Therefore, the successive refinement approach is used in the current research [18]. The method is capable of producing high-quality (regular) stretched cells inside the boundary and shear layers as well as isotropic cells outside these regions. During the optimization process, the airfoil boundaries are changing; therefore, the existing grid is modified in an automatic manner using tension–spring analogy in order to be adapted to the changing domain [19].

4. Parallelization Methodology

Several parallelization methods can be considered for the problems related to the time-consuming CFD simulations [20, 21]. It was only during the recent years that efforts have been made to propose strategies for designing PGAs in the field of the evolutionary aerodynamic shape optimization. Panmictic GAs can be parallelized readily by using master/slave model, which works well for a small number of individuals. However, as the number of nodes increases it becomes inefficient by excessive communications. Cellular PGA is designed to run on massively parallel processing computers. In such an algorithm, selection and mating are limited to small groups that overlap to permit some interactions among all individuals. Hence, good solutions might be disseminated across the entire populations. Sometimes, the Cellular parallel GA is also termed as the Fine-grained PGA. A distributed PGA may sound more complicated, as it consists of several subpopulations that exchange members occasionally. This exchange of members is called migration that is controlled by several parameters. Distributed PGAs are also known as the multi-deme or island model PGA. Figure 1 shows such general models of PGA.

Various PGA models may be used together to produce other Hierarchical PGA (HPGA) models. For example, one may apply a hierarchical PGA that combines a distributed PGA and a master–slave PGA, which we consider in this paper, or even another level of island PGAs. Basically, HPGA is any combination of two or more of the three basic forms of PGA.

Although PGA is widely used in different fields of optimization, many important parameters need to be tuned when it is applied in the field of aerodynamic shape optimization. In the present work, following the work of Ebrahimi and Jahangirian [22] a two-level HPGA including is applied where island and master/slave model are used for the first and second layers, respectively. The main steps of the applied framework are as following:

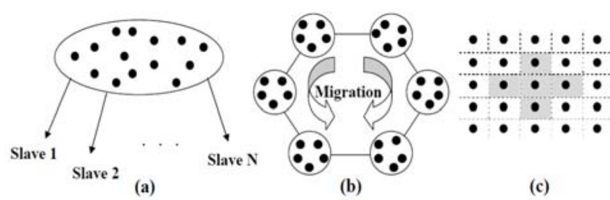

Fig. 1 Different basic PGAs (a) master/slave (b) cellular (c) distributed

1- The main tasks of the framework including GA operators and airfoil shapes generators are transferred onto the master nodes.
2- Parallel subpopulations' evolutions then begin at the selected computing clusters. Whenever they receive a launch request of the subpopulation evolution service, job submission protocol is represented at the master node of the respective clusters.
3- At each cluster, scheduling and resource discovering is conducted to farm the field of available processing nodes for chromosome evaluations.
4- Once, all individuals are evaluated by numerical solver, the obtained results are marshaled back to the master node to undergo the parallel algorithm.
5- The developed subpopulations are sending back to the master nodes to proceed with the migration operation. Such a process repeats until the optimization criteria is met.

5 Population Dispersion

Since genetic drift is one of the main reasons of the delay in the convergence of evolutionary optimization algorithm, applying a proper population dispersion method plays an important role in enhancement of the convergence rate. One of the key factors for keeping the gene diversity, longer than the Simple Genetic Algorithm (SGA) is applying Adaptive Range GA. While the gene diversity helps to the robustness of the framework, the adaptive feature improves its local search capability.
The new technique developed in this paper provides a better diversity in the design space where unlike most of well-known mutation adaptation methods, the proposed one has its own mutation value for each gene. Firstly, the mutation rates for all genes are set to an initial value in a specified boundary. Based on the feedback obtained by monitoring fitness value evaluations of members, an adaptive approach for adjusting mutation rates for the gene locations is proposed. The proposed technique consists of two phases of exploration and refinement. In the phase of exploration, the aim is to scatter all genes across the entire domain. At each generation, the mutation rate (MR_i) is updated based on the feedback taken from the fitness value of individuals. If the fitness value corresponding to the gene location (FVP_i) is less than the average fitness value, based on the formulation (8), the mutation rate for the corresponding gene is increased by the value of a_i. On the other hand, if FVP_i to FV_{Avg} ratio is more than 1, then MR_i is decreased accordingly. For minimization problem, an inverse procedure should be applied.
By taking feedback at each generation, MR_i values are allowed to vary within the lower and upper limits. If an update, results a parameter to exceed the limits, it automatically is changed according to the formulation 9 as following:

$$a_{i,\ new} = \begin{Bmatrix} a_{i,\ old} + \lambda \ if \dfrac{FV_{pi}}{FV_{avg}} < 1 \\ \\ a_{i,\ old} - \lambda \ if \dfrac{FV_{pi}}{FV_{avg}} \geq 1 \end{Bmatrix} \tag{8}$$

$$MR_{i,new} = \begin{Bmatrix} \dfrac{2}{N * L} + a_{i,new} & if \ MR_{i,old} < 0.3 \\ a_{i,old} & if \ MR_{i,old} \geq 0.3 \end{Bmatrix} \tag{9}$$

where N and L are the size and the length of individuals and λ is mutation update in each generation which is kept under 15%. Based on the numerical experiments the exploration phase usually consists of around 10 to 14 generations. After this phase and when the generated data spread out enough in the whole domain, the refinement phase begins. In this phase, the majority of the new GA individuals start to concentrate in the selected regions of the domain while still some members seek regions where have not been searched fairly. At the end of each three generations, the best chromosomes randomly are switched between subpopulations.

6 Result

This section is divided in to three parts; firstly the efficiency of the proposed population dispersion method through a simple geometric reconstruction problem is investigated. In the second part, the performance of the parallelization is inquired and finally the method is applied for aerodynamic shape design and results are discussed.

6.1 Efficiency Assessment

The developed PGA with adaptation mutation strategy is expected to provide better diversity in the design space and decrease the computational time of optimization process. To evaluate the generality and flexibility of the proposed method, an inverse geometric reconstruction procedure is applied and a developed PARSEC parameterization technique is used for airfoil shape generation [2]. An awkward shape is chosen as the initial shape and the target airfoil is RAE2822. An iterative optimization by proposed PGA is carried out in order to assess the capability of the method for producing the goal shape. The objective function for this problem is defined as following:

$$G = \frac{\sum_{i=1}^{n_p}(Y_{di} - Y_{gi})^2}{2n_p} \tag{10}$$

where Y_{di} and Y_{gi} are the design and target coordinates of the surface points with fixed X_i coordinates. This formulation should be minimized in the process of optimization.

Figure 2 shows the initial and target airfoils as well as the final airfoil generated by the proposed method after 200 generations. As illustrated, the trailing edge of the initial shape is considered different with the final shape. To investigate the performance of the proposed method, in Figure 3, the obtained shape at the trailing edge is compared with the one generated without using the propose technique (SGA). This figure indicates that the current work gets the target shape more effectively.

To statistically investigate the efficiency of the proposed technique, in Table 1, mean values (μ) and standard deviation (σ) of the upper crest location (one of the parameterization parameters) are compared at 11th (exploration phase) and 25th (refinement phase) generations with the ones with fixed mutation rates of 1% and 5%. According to this table, in the exploration phase, genes in the proposed technique are scattered in the design space more properly and search is carried out in a wider range of the design space. Better results also are obtained in the refinement phase.

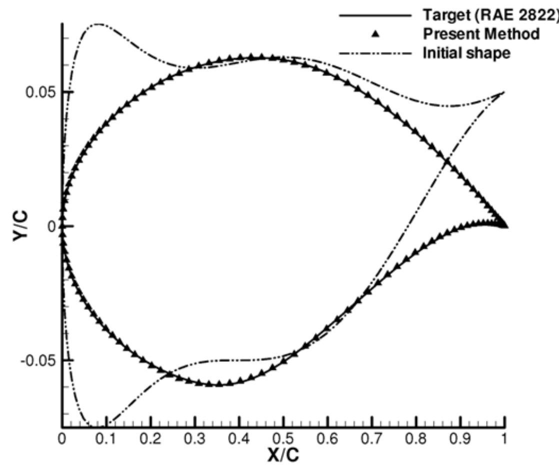

Fig. 2 Initial, target and the final airfoils obtained by inverse geometric reconstruction

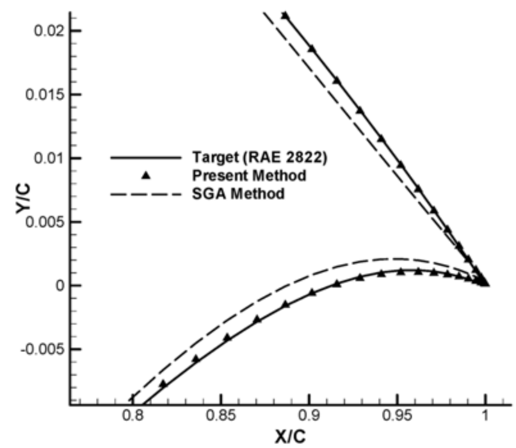

Fig. 3 Target and design airfoils at the trailing edge for inverse geometric reconstruction

Table 1 Statistical comparison of exploration and refinement phase for inverse geometry reconstruction

Applied method	Generation No.	Mean Values (μ)	Standard Deviation (σ)
Simple GA with 1% MR	11	0.0671	3.69E-4
	25	0.0572	3.09E-3
Simple GA with 5% MR	11	0.0642	3.51E-4
	25	0.0563	2.87E-3
Present method	11	0.0547	3.08E-3
	25	0.0539	2.95E-3

6.2 Parallel performance study

In this section, the performance of the parallel strategy under various numbers of individuals and the cluster size is investigated.

6.2.1 Optimizing the population size

When parallelization is applied for evolutionary shape optimization, one of the key factors for the successfulness of the algorithm is the selection of the optimum population size. As the population size, which is equal to the number of processors increases, the computational time of the optimization process will rise. That is due to the fact that each shape requires different number of CFD iterations for evaluation. Higher population size in turn could lead to lower required numbers of generations in order to gain the same level of objective values. Therefore, to minimize the clock time of optimization process, a compromise between the population size and the required numbers of generations should be applied. In Figure 4, for three different airfoil shape optimization problems, the optimization time by Parallel GA against the population size is illustrated. It should be noted that the calculated time here is the period when the program starts; up to the time it reaches to the objective value of 58.5. According to this figure the optimum population size is 20 in these cases.

6.2.2 Parallel Speed-up

When a parallel algorithm is executed, one of the main performance issues is that how much speed-up the parallelization can offer. Such a speed-up is defined as following:

$$S_p = \frac{T_s}{T_p} \tag{11}$$

where T_s and T_p are the execution time of the sequential and parallel algorithms, respectively. To compare the efficiency of the proposed method in terms of the actual clock time of optimization process, a parameter called Cost Function Efficiency (CFE) is introduced using Amdahl's law [23]. The performance of the proposed parallelization strategy for the above airfoil design problem is assessed and presented for the population sizes of 12, 20 and 32. For all cases, the calculation time defines the period when the program starts; up to the time it reaches to the objective value of 58.5.

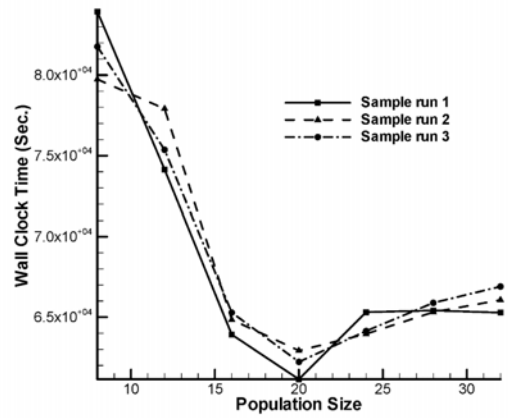

Fig. 4 Computational time of three airfoil shape optimization problems against the population size

Looking at Table 2, it is observed that by increasing the population size, the speed-up is increased. More importantly, it shows that how the usage of a proper parallelization strategy could lead to more CFE, which means applying more subpopulations as well as individuals do not always result in more efficiency. For instance, when the number of subpopulations and population size are 32 and 4 respectively, the CFE is about 25% less than when 20 individuals and 3 subpopulations are utilized. The main reason is that by using more subpopulations the idle time of processors increases. However, no significant different is observed in the number of generations of the optimization process. In addition the semi-liner speed-up indicates that the model is suited for modern cluster work stations.

Table 2 Speed-up and Cost Function Efficiency of the proposed method for different cluster sizes

	One Sub-population (PGA)			Two Sub-population			Three Sub-population			Four Sub-population		
Population size	12	20	32	12	20	32	12	20	32	12	20	32
Speed-up	10.9	18.1	29.0	20.2	34.8	54.7	29.7	51.9	81.1	38.8	67.8	106.0
CFE	63.9	88.0	67.4	68.1	90.7	71.4	69.2	91.8	71.8	64.8	88.7	68.7

6.3 Aerodynamic Shape Optimization

In this section the aerodynamic efficiency of the proposed framework is investigated. In addition, a comparison between the parallel and serial outcome are carried out. More detail about the serial method may be found in [2]. A transonic flow is considered with the Mach number 0.73, Reynolds Number 6.3 million and incidence angle 2.8 degrees. The RAE-2822 airfoil is considered as the initial airfoil and the objective function is the lift coefficient (C_l) to the drag coefficient (C_d) which is computed by solving the Reynolds-averaged Navier–Stokes equations. The computational field is discretized using triangular unstructured grids. The Mach contour and unstructured grid generated around the initial and design airfoil using spring analogy is illustrated in Figure 5 and 6, respectively. According to these figures, there is a rather strong shock wave near the middle part of the initial airfoil upper surface that is weakened in the optimum shape.

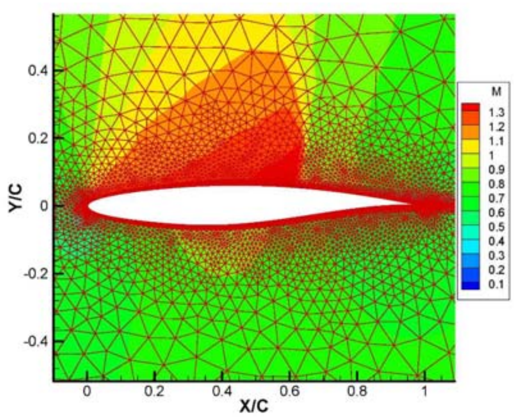

Fig.5 Mach contour and unstructured grids around the initial airfoil

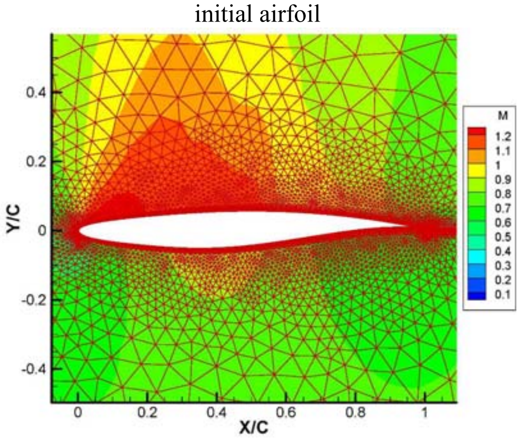

Fig.6 Mach contour and unstructured grids around the design airfoil

The initial and final airfoil shapes are plotted in Figure 7. Also the distributions of surface pressure coefficient (C_p) for the presented PGA method and serial optimization algorithms (considering imposed physical constrains in [2]) is illustrated in Figure 8. This figure also emphasizes the successfulness of the presented method in decreasing the intensity of the shock wave.

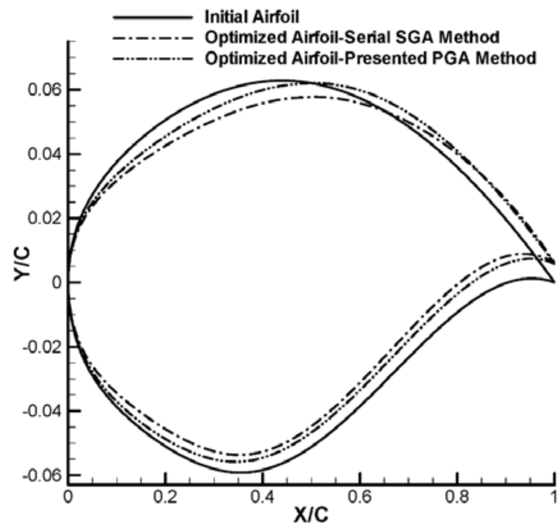

Fig. 7 Obtained airfoil shapes for presented PGA and serial SGA methods

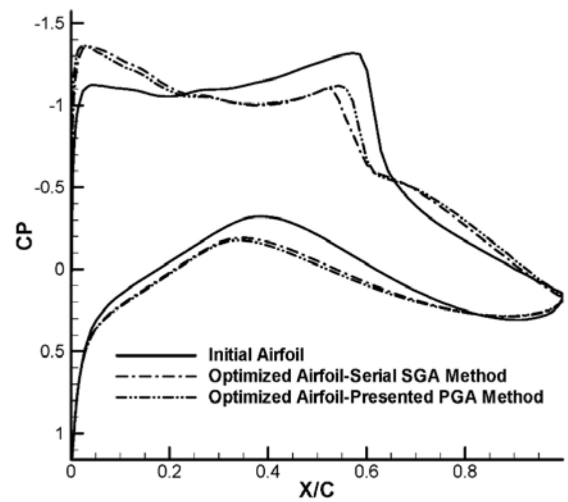

Fig. 8 Obtained surface pressure coefficients for presented PGA and serial SGA methods

The values of lift and drag coefficients and the objective functions for the initial and optimum shapes are also shown in Table 3. According to this table no significant divergence is observed for the parallel and serial solutions. The limited differences between serial and parallel results can be assumed due to the random nature of GA.

Table 3 Lift and drag coefficients for SGA and PGA methods

	C_l	C_d	C_l/C_d	Execution time (hr)
Initial Shape	0.81	0.0261	31.09	-
Parallel SGA Method	0.871	0.0152	57.3	24.7
Presented Population dispersion method-Serial solution	0.883	0.0149	59.26	525
Presented Population dispersion method-Parallel solution	0.882	0.0149	59.19	17.2

The above table also indicates that with applying the presented PGA algorithm not only the objective function is improved around 3.5% but also the optimization time is reduced about 30%.

7 Conclusion

A two-level Parallel Genetic Algorithm optimization method strategy including a master–slave PGA at the lower level and a distributed PGA at the upper one was proposed. Some crucial parameters such as mutation rate and cluster size were optimized. The efficiency of the method was investigated through airfoil shape optimization. It was found that by using the proposed strategy for aerodynamic shape optimization significant increase in the objective function and reduction in the computational time is obtained. The semi-liner speed-up also showed that the method is suited for modern cluster work stations.

References

[1] Shahrokhi A and Jahangirian A. Inverse design of transonic airfoils using genetic algorithm and a new parametric shape method. *Inverse Problems in Science and Engineering,* Vol. 17 No. 5, pp. 681–699, 2009.

[2] Ebrahimi M and Jahangirian A. Aerodynamic Optimization of Airfoils Using Adaptive Parameterization and Genetic Algorithm. *Journal of Optimization Theory and Applications,* Vol. 162 No. 1, pp. 257-271, 2014.

[3] Shahrokhi A and Jahangirian A. surrogate assisted evolutionary optimization method with application to the transonic airfoil design, Engineering Optimization, Vol. 42 No. 6, pp. 497–515, 2010.

[4] Andrés-Péreza E, González-Juárezb D, Martin-Burgosb M, Carro-Calvoc L and Salcedo-Sanzc S . Influence of the number and location of design parameters in the aerodynamic shape optimization of a transonic airfoil and a wing through evolutionary algorithms and support vector machines. *Engineering Optimization,* DOI: 10.1080/0305215X. 2016.1165568, 2016.

[5] Leifssona L and Kozielb S. Aerodynamic shape optimization by variable-fidelity computational fluid dynamics models. *A review of recent progress, Journal of Computational Science*. Vol. 10 No. 1, pp. 45–54, 2015.

[6] Ochoa G. Setting the Mutation Rate: Scope and Limitations of the 1/L Heuristic", *Proceedings of Genetic and Evolutionary Computation Conference*, Morgan Kaufmann (2002).

[7] Khurana M and Massey K. Swarm algorithm with adaptive mutation for airfoil aerodynamic design. *Swarm and Evolutionary Computation*, Vol. 20 pp. 1-13, 2015.

[8] Eiben A E, Michalewics Z, Schoenauer M and Smith J E. Parameter Control in Evolutionary Algorithms. In F. Lobo, C. Lima, and Z. *Michalewics (Eds), Parameter Setting in Evolutionary Algorithms*, Chapter 2, 19–46, 2007.

[9] Greffenstette J J. Optimization of Control Parameters for Genetic Algorithms. *IEEE Trans. On Sys. Man and Cyber*, Vol. 16, No. 1, pp. 122–128, 1986.

[10] Back T, Fogel D.B, Michalewicz Z, and others. *Handbook on Evolutionary Computation. Institute of Physics Publishing*, Bristol, 1997.

[11] Krink T, Thomsen R. and Rickers P. Applying Self-Organised Criticality to Evolutionary Algorithms. In Schoenauer, Deb, M., Rudolph, K., Yao, G., Lutton, X., Merelo, E., and Schwefel, H.P. (eds.), *Parallel Problem Solving from Nature*, PPSN VI, Vol. 1 pp. 375-384, 2000.

[12] Arakawa M and Hagiwara I. Development of adaptive real range genetic algorithm. *JSME International Journal Series C,* Vol. 41 No. 4 pp. 969-977, 1998.

[13] Sasaki D and Obayashi S. Efficient search for trade-offs by adaptive range multi objective genetic algorithms. *Journal of Aerospace Computing, Information and Communication*, Vol. 2 No. 1, 44-64, 2005.

[14] Tes D and Chan Y.Y. Multi-point design of airfoil by genetic algorithm. *In: 8th Annual Conference of the CFD Society of Canada*, Montreal, 2000.

[15] Deb K. Multi-objective Optimization Using Evolutionary Algorithms. *Academic Press*, New York, 2001.

[16] Launder B.E. and Spalding, D.B. The numerical computation of turbulent flow. Computer Methods in

Applied Mechanics and Engineering. Vol. 3, pp. 269–289, 1974.

[17] Jahangirian A and Hadidoolabi M. Unstructured moving grids for implicit calculation of unsteady compressible viscous flows. *International Journal for Numerical Methods in Fluids. Vol.* 47, pp. 1107–1113, 2005.

[18] Jahangirian A and Johnston L.J. Automatic generation of adaptive unstructured grids for viscous flow applications. *In 5th International Conference on Numerical Grid Generation in CFD*, Mississippi State University, 1996.

[19] Shahrokhi A, Jahangirian A. and Fouladi N. Navier–Stokes optimization using genetic algorithm and a flexible parametric airfoil method. *In: ERCOFTAC Conference on Design Optimization: Methods and Application,* Spain, University of Las Palmas de Gran Canaria, 2006.

[20] Ebrahimi M, Daneshtalab M, Liljeberg P, Plosila J. and Tenhunen H. Cluster-based Topologies for 3D Networks-on-Chip Using Advanced Inter-layer Bus Architecture. Elsevier Journal of Computer and System Sciences. Vol. 79, No. 4, pp. 475-491, 2013.

[21] Ebrahimi M and Jahangirian A. New analytical formulations for calculation of dispersion parameters of Gaussian model using parallel CFD. Environmental Fluid Mechanics. Vol. 13, No. 2, pp. 125-144, 2013.

[22] Ebrahimi M and Jahangirian A. A Hierarchical Parallel Strategy for Aerodynamic Shape Optimization with Genetic Algorithm. *Journal of Computer Science & Engineering,* Vol. 22, No. 6 pp. 2379-2388, 2015.

[23] Ahmdal G. Validity of the single processor approach to achieving large scale computing capabilities. *In: AFIPS Conference Proceedings,* Thompson Books, Washington DC. 30 pp. 483–485 (1967).

Contact Author Email Address

ajahan@aut.ac.ir

Copyright Statement