RESEARCH NOTE

AERODYNAMIC DESIGN OPTIMISATION USING GENETIC ALGORITHM

S. Ziaei-Rad and A. R. Pishevar

Department of Mechanical Engineering, Isfahan University of Technology
Isfahan, Iran, szrad@cc.iut.ac.ir - apishe@cc.iut.ac.ir

(Received: June 17, 2002 – Accepted in Revised Form: August 11, 2003)

Abstract An efficient formulation for the robust shape optimization of aerodynamic objects is introduced in this paper. The formulation has three essential features. First, an Euler solver based on a second-order Godunov scheme is used for the flow calculations. Second, a genetic algorithm with binary number encoding is implemented for the optimization procedure. The third ingredient of the procedure is the use of previous flow that is closest in terms of geometric parameters to the new shape as an initial condition for the new function evaluation. This makes the solution towards the final value progressively faster and reduces the computer time for the convergence of the algorithm. The algorithm is used to optimize two different problems, a simple bump problem, and a two-dimensional transonic airfoil problem using an Euler solver equation. The results indicate that the GA/flow algorithm is robust and can optimize a wide range of problems with a minimum implementation effort.

Key Words Aerodynamic Design Optimization, Genetic Algorithm, Euler Solver, Godunov Scheme

چکیده در این مقاله روشی کارآ و فراگیر برای بهینه سازی شکل هندسی اجسام آیرودینامیک ارائه گردیده است. فرمولاسیون و کد تدوین شده برای این منظور دارای سه خاصیت زیر می باشد.۱- برای حل مسئله حرکت سیال از فرمولاسیون مرتبه دوم گودونف استفاده شده است. استفاده از این الگوریتم امکان پیش بینی دقیق تر محل شوک را در جریان سیال فراهم می سازد. ۲- کد توسعه یافته برای بهینه سازی، از الگوریتم ژنتیک با کد باینری بهره می‌جوید. ۳- برای سرعت بخشیدن به همگرایی، کد از نتایج بدست آمده از حلهای قبلی بعنوان شرایط اولیه در حل جاری استفاده می‌کند. سپس دو مسئله کاربردی به عنوان نمونه توسط الگوریتم حل گردیده است: مسئله جریان سیال بر روی یک مانع قوسی شکل و جریان ترانسونیک بر روی یک ایرفویل NACA چهار رقمی. نتایج نشان می‌دهد که کد تدوین شده فراگیر بوده و می‌تواند مسائل دیگر مربوط به بهینه سازی شکل هندسی اجسام آیرودینامیک را با تغییرات جزئی حل نماید.

1. INTRODUCTION

In the mid-70s, researchers [1-5] began exploring the use of numerical optimization techniques for the design of aircraft components. These early studies primarily focused on airfoil and wing design using low accurate fluid models for the analyses and finite difference calculations for gradient information. The inability of these fluid models to accurately predict nonlinear phenomena limited their applicability. Improvements in computational resources have been directed researchers into the use of higher fidelity Euler and Navier-Stokes equations in calculating flows around isolated components and moderately complex configurations. In mid 80s, Sobieski [6] challenged the aerodynamic community to extend their computational fluid dynamic (CFD) algorithms to include the shape sensitivity analysis of the geometry. This plea ignited intense studies aimed at developing methods that would render the use of nonlinear aerodynamics in shape optimization feasible.

Numerical techniques for optimizing performance in aerospace industries have been studied by many researchers over the years. Perhaps the most widely used are those based on the calculation of the gradients. Reuther [7] presented a review of aerodynamic design using gradient methods. As the main theme of this paper is the use of genetic algorithm (GA) in aerodynamic design optimization, only a brief review of this approach is given here.

Quagliarella and Della Cioppa [8] used GA to optimize the airfoil utilizing potential-based flow solver. Vicini and Quagliarella [9] used GA for multi-point and multi-objective airfoil applications. Obayashi et al. [10] applied the algorithm for multi-disciplinary optimization of transonic wings. Examples of multi-design point wing optimization using Euler and Navier-Stokes flow solvers can be found in papers by Sasaki et al. [11] and Oyama [12-13].

Holst and Pulliam [14] proposed a method for aerodynamic shape optimization of airfoils and transonic wing using a real number encoding genetic algorithm. For the airfoil they used an Euler equation solver while a nonlinear potential solver was used for the transonic wing optimization.

A drawback of the GA approach is mainly its expense. In general, the number of function evaluation required for a GA algorithm exceeds the number required by a finite-difference-based gradient optimization [15,16]. Recently, Doorly et al [17-19] applied a parallel genetic algorithm for aerodynamic design optimization. They applied the method to a 2D wing section and discussed the advantage of the method over the sequential genetic algorithm.

In this paper a progressive optimization was used for design of 2D airfoil section using sequential genetic algorithm. An Euler solver based on a second-order Godunov scheme is used for the flow calculations. Using information from the previous runs as initial conditions for the new design parameters increased the computational efficiency by a factor of five.

In the next sections, first the flow solver will explain briefly followed by a quick description of the genetic algorithm. At the end, the capability of the proposed method is demonstrated by giving a few examples.

2. FLOW SOLVER

The two-dimensional Euler equations governing the inviscid flow of a gas can be considered as:

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

Where Q is the vector of conserved variables and F, G are fluxes in x and y directions, respectively:

$$\begin{aligned} Q &= [\rho, \rho u, \rho v, E]^T \\ F &= [\rho u, \rho u^2 + p, \rho uv, u(E+p)]^T \\ G &= [\rho v, \rho uv, \rho v^2 + p, v(E+p)]^T \end{aligned} \qquad (2)$$

ρ denotes density, u and v the Cartesian velocity components and E is the total energy. The set of equations is completed by an equation of state and a set of proper initial and boundary conditions. In all simulations, the ideal gas behavior with a constant specific heat ratio of $\gamma = 1.4$ is assumed:

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2) \qquad (3)$$

The above set of equations is discretised by an explicit second-order Godunov-type scheme via the locally one-dimensional time-splitting method. For the sake of simplicity, here the algorithm is explained on a regular Cartesian grid but the idea can be extended to the generalized curvilinear coordinates in a similar manner. First, consider that part of Equations 1, which contains only the contribution of fluxes in the x direction:

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} = 0 \qquad (4)$$

The discrete form of this equation can be obtained by integrating (4) over a cell confined to the $[x_{j-1/2}, x_{j+1/2}]$ and a small time interval of $[t^n, t^{n+1}]$:

$$\overline{Q}_j^{n+1} = \overline{Q}_j^n - \frac{\Delta t}{\Delta x}(\hat{F}_{j+1/2} - \hat{F}_{j-1/2}) \qquad (5)$$

where $\overline{Q}_j^n$ denotes the cell average values and

$\hat{F}_{j+1/2}$ is the numerical flux function defined as:

$$\overline{Q}_j^n = \int_{x_{j-1/2}}^{x_{j+1/2}} Q(x,t)dx$$
$$\hat{F}_{j+1/2} = \int_{t^n}^{t^{n+1}} F(Q(x_{j+1/2}, t))dt \qquad (6)$$

Here, $\hat{F}_{j+1/2}$ is calculated by a second-order Godunov type method as follows: first, a monotonised central difference slope-limiter function [20] is used to obtain a second-order representation of initial averaged value within each cell. This representation of initial condition may lead to a set of discontinuous solution at the cell interfaces. Since the gradients of flow variables are also discontinuous at each cell interface, the solution to this initial condition is not self-similar in time and therefore, a Generalized Riemann Problem (GRP) should be solved in the next step to take into account the time behavior of $\hat{F}_{j+1/2}$:

$$\hat{F}_{j+1/2} = F^R_{j+1/2} + \frac{\Delta t}{2}\left(\frac{\partial F}{\partial t}\right)^n_{j+1/2} \qquad (7)$$

Here, $F^R_{j+1/2}$ denotes the flux contribution from a Riemann solver using the constant interface variable values obtained in the first step. The Roe approximate Riemann solver is used to determine $F^R_{j+1/2}$. Depends on the wave pattern emanating from discontinuities at initial time t^{n^+}, the time derivatives are calculated by a closed form solution given by Ben-Artzi and Falcowitz [21]. After updating the initial condition by (5) the same algorithm is used to calculate the contribution of convective fluxes in the other direction.

Now let $L_x^{\Delta t}, L_y^{\Delta t}$ be the local one-dimensional split operator in x and y direction, respectively. Then a fractional step method for the Euler equations become:

$$\overline{Q}^{n+1}_{j,k} = L_y^{\Delta t/2} L_x^{\Delta t} L_y^{\Delta t/2t} \overline{Q}^n_{j,k} \qquad (8)$$

The proposed numerical algorithm is second-order accurate in time and space. The details of implemented scheme were discussed in previous publication [22] and for the sake of brevity are omitted here.

For each test case, one of the following boundary conditions may be implemented. At an inflow boundary depends on free-stream Mach number the flow parameters are either specified according to the known free-stream state or a non-reflective boundary condition is applied. At solid boundaries, the normal velocity component and the normal pressure gradient are set to zero. The zero gradient extrapolation or non-reflective boundary condition depends on local flow Mach number is applied to the outflow boundary.

3. NUMERICAL OPTIMIZATION AND GENETIC ALGORITHM (GA)

3.1 Numerical Optimization Numerical optimization is a vast field, which has been the subject of numerous text books [23-25]. One can categorize the optimization methods into two major classes, namely gradient-based (sometimes referred to as first-order) methods and global-based methods.

The solution procedure for gradient-based methods may be decomposed into four distinct steps: (a) evaluation of the objective function, F, to be minimized or maximized and any constraints, C, to be imposed, (b) evaluation of the gradients of the objective function ∇F, and ∇C constraints, with respect to the vector of design variable, commonly referred to as sensitivity derivatives, (c) determination of the search direction upon which the design variables will be updated, and (d) determination of the optimum step length.

However, in many engineering applications optimization problems with non-smooth, non-differentiable, highly non-linear and many local minima cost functions are commonly encountered. In these applications conventional gradient-based algorithms are ineffective due to the problem of local minima or the difficulty in calculating gradients. Optimization methods that require no gradient and can achieve a global optimal solution offer considerable advantages in solving these difficult optimization problems. The two best-known classes of such global optimization methods are the genetic algorithm (GA) [26-28] and the simulated annealing (SA) [29-31].

3.2 Genetic Algorithm A GA is a search method based on natural selection and genetics. The central theme of the research on GAs has been the balance between the robustness and the efficacy necessary for survival in many different environments. The GAs are computationally simple but powerful and not limited by assumptions about the search space.

The GA may be thought as an evolutionary process, where a population of solutions evolves over a sequence of generations. During each generation, the fitness (goodness) of each solution is calculated, and solutions are selected for reproduction on the basis of their fitness. The probability of survival of a solution is proportional to its fitness value. This process is based on the principle of survival of the fittest. The reproduced solutions then undergo recombination, which consists of crossover and mutation. A genetic representation may differ from the real form of the parameter of the solutions. Fixed-length and binary encoded strings have been widely used for representing solutions.

A simple GA is really easy to use. It uses three basic generic operators: reproduction, crossover and mutation. Reproduction is a process in which individual solutions are copied according to their fitness value (objective function values). Crossover requires a mating of two randomly selected strings of solution. The information on the strings is partly interchanged according to a randomly chosen crossover site. Crossover is applied to take valuable information from the parents, and its applied with a certain probability. Mutation is the occasional random alteration of the value of a string position. Mutation insures against bit loss, and can be a source of new bits.

The genetic algorithm used here requires determination of six fundamental issues: chromosome representation, selection function, the genetic operators making up the reproduction function, the creation of the initial population, termination criteria and the evolution function. Section 3.2 describes each of these issues.

3.2.1 Chromosome Representation and Selection Function Starting with any optimization problem, each member of the population of initial trial solution is encoded as a string (or chromosome), which specifies the particular values of the design variables (in our case the parameters that define the shape of the airfoil). In this work, GA genes are computationally represented using bit strings (binary codes) and the operators (mutation, crossover, …) are designed to manipulate bit string data.

The selection of individuals to produce successive generation plays an extremely important role in the genetic algorithm. A probabilistic selection is performed, based upon the individual's fitness such that the better individuals have an increased chance of being selected. An individual in the population can be selected more than once with all individuals in the population having a chance of being selected to reproduce into the next generation. There are several schemes for the selection process.

- Roulette Wheel is the traditional selection function with the probability of surviving equal to the fitness of individual i, divided by the sum of the fitness of all individuals.
- Norm Geometry Selection is a ranking selection function based on the normalized geometric distribution.
- Tournament Selection is working by selecting *j* individuals randomly and, with replacement from the population and inserts the best of the *j* into new population.

3.2.2 Genetic Operator Genetic Operators provide the basic search mechanism of the GA. The operators are used to create new solutions based on existing solutions in the population. There are two basic types of operators: crossover and mutation.

Crossover takes two individuals and produces two new individuals while mutation alters one individual to produce a single new solution. The application of these two basic types of operators and their derivatives depends on the chromosome representation used. The most important ones that can be used in the program are:

- Arithmetic crossover
- Heuristic crossover

- Simple crossover

Mutation is the occasional random alteration of the value of a string position. The applied mutation methods in the program are:

- Boundary mutation
- Multi-non-uniform mutation
- Non-uniform mutation
- Uniform mutation

3.2.3 Initialization, Termination, and Evaluation Functions The GA must be provided with an initial population. The most common method is to generate random solutions for the entire population. However, since GAs can iteratively improve existing solutions (i.e., solutions from other heuristics and/or current practices), the starting population can be seeded with potentially good solutions, with the remainder of the population being randomly generated solutions.

The GA moves from generation to generation selecting and reproducing parents until a termination criterion is met. The most frequently used stopping criterion is a specified maximum number of generations. Another termination strategy involves population convergence criteria. In general, GAs will force much of the entire population to converge to a single solution. When the sum of the deviations among individuals becomes smaller than some specified threshold, the algorithm can be terminated. The algorithm can also be terminated due to a lack of improvement in the best solution over a specified number of generations.

Evaluation functions of many forms can be used in a GA, subject to the minimal requirement that the function can map the population into a partially ordered set. As stated, the evaluation function is independent of the GA (i.e., stochastic decision rules).

4. NUMERICAL IMPLEMENTATION

The first step in the aerodynamic design optimization is the definition of the objective function. In sensitivity-based methods, it has been shown that to obtain smooth sensitivity derivatives, a central difference schemes with sufficient level of artificial viscosity is needed [32]. This has the effect of smearing the shock and, as a consequence, regularizing the objective function. The drawback is that the objective function will lose accuracy due to high degree of smearing at the shock and, therefore, the position of the minimum may be inaccurate. As GA does not need the calculation of the derivatives, the objective function can be calculated accurately. Therefore, the developed code utilizes an appropriate basic flow solver to accurately resolve all-important features of the flow field in order to accurately evaluate the objective function. The optimization procedure involves the following steps.

1. start with a random set of population (initial generation)
2. selection of parents base on the current population using objective function
3. constructing the geometry and the mesh for each set of design variables
4. apply the boundary conditions to the problem
5. solve the flow equations
6. compute the objective function and the fitness of each solution
7. repeat from step 3 for all population inside a generation
8. apply cross over and mutation rules to create the next generation
9. repeat from step 2 until the objective function reaches its minimum

The implemented genetic procedure for this code is a standard genetic algorithm. It uses binary code for chromosome representations. Roulette Wheel for the selection of individuals, arithmetic crossover, uniform mutation and maximum number of generation as its termination function. In all calculations, the probabilities for the cross over and mutation were set to 0.5 and 0.02 respectively.

Computational cost is mainly affected by step 5. The computational efficiency can be improved progressively by using previous results, which has the closest design parameters to the new design parameters. This can be done as follows: suppose that in ith function evaluation, the design parameters were $P_n(i),\ n = 1,\ldots,N$ and in the present function evaluation parameters are $P_n(j),\ n = 1,\ldots,N$. Then, the initial conditions for the present run is

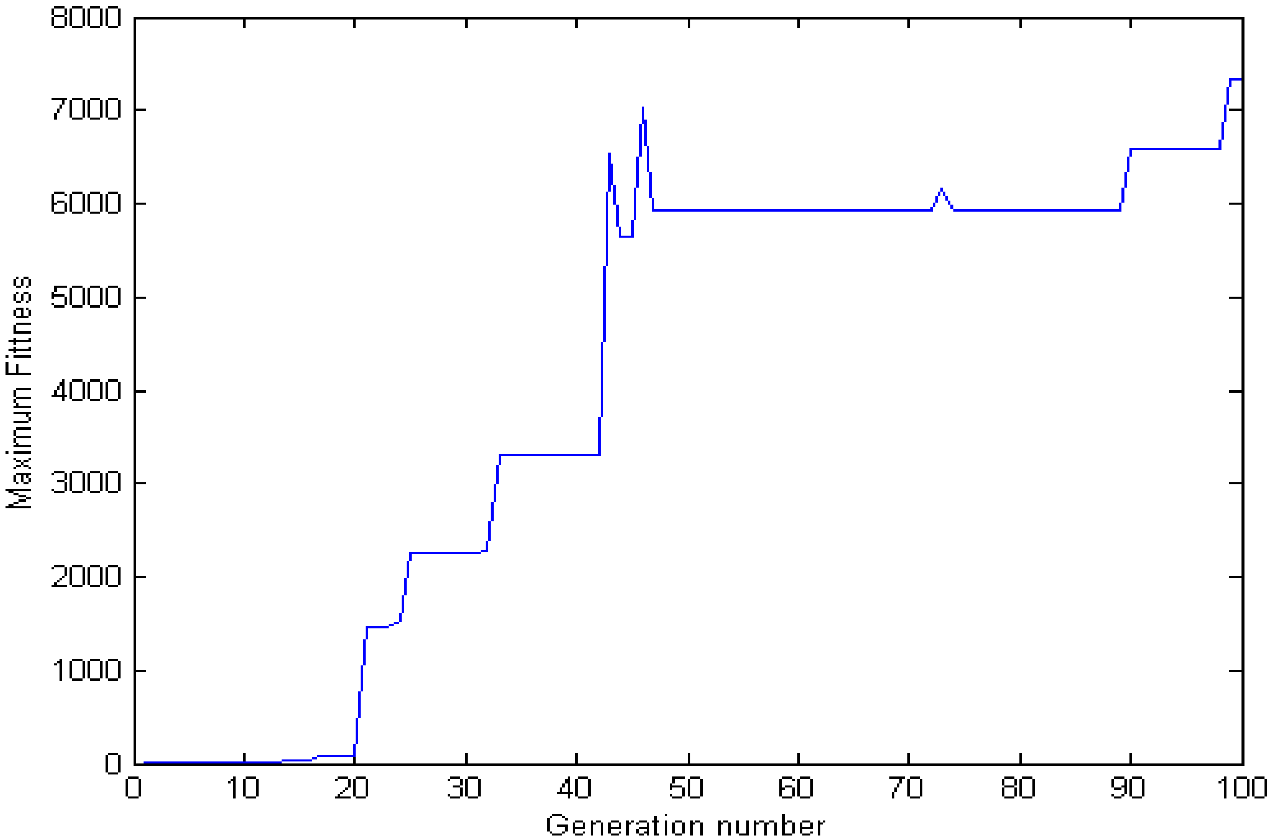

Figure 1. The GA maximum fitness versus generation number for the bump problem.

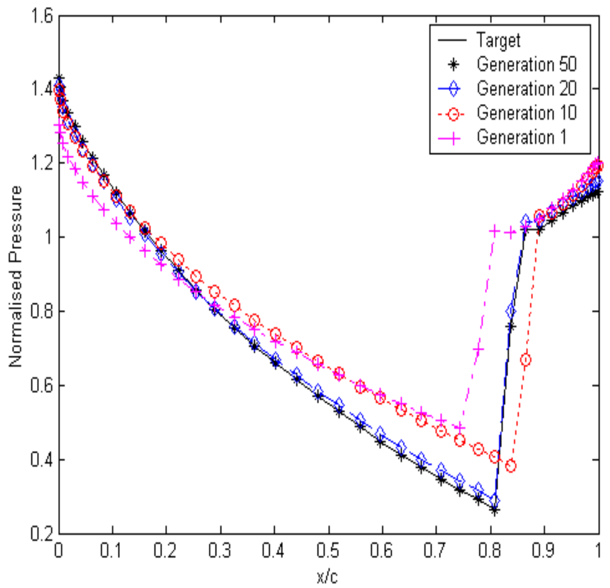

Figure 2. The pressure distribution at different generations for the bump problem.

obtained from the converged results of previous runs that has the minimum value of:

$$\sum_{i=1}^{j-1}\left|\frac{P_n(i)-P_n(j)}{P_n(j)}\right|$$

Depends on the distance between these parameters, the number of iterations in each flow calculation step can be reduced drastically.

5. CASE STUDIES

5.1 Case 1 The first problem used to evaluate the performance of the method is the flow pass over a bump. The shape of the bump was assumed to be part of a circle. The inverse problem here is to find the shape of the bump, which reproduces a specified pressure distribution. As an optimization

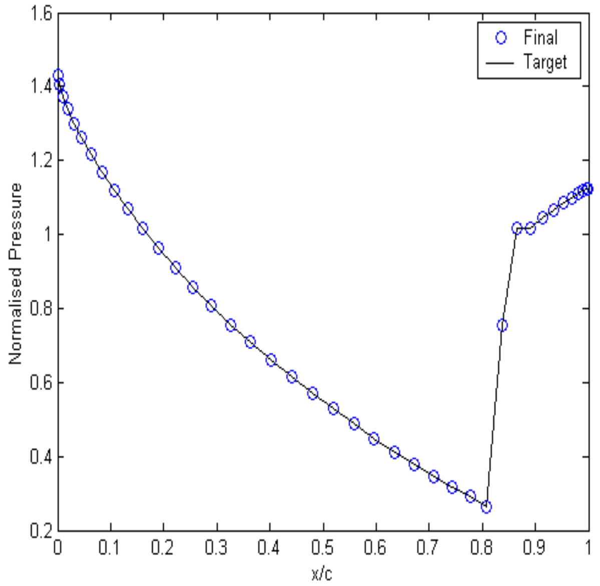

Figure 3. The final and target pressure distribution for the bump problem.

problem the task is to find the shape that minimizes the difference between the desire and actual pressure on the bump. The only parameter change during the optimization procedure is the radius of the bump. The target solution was the computed pressure distribution over a bump with a radius of 1 and at inviscid transonic flow condition of $M_\infty = 0.8$. Each gene in the GA selection is given a random initial value in the range of 0.5 to 2. The objective (fitness) function for this case was defined as:

$$F_{obj} = \frac{1.0}{\sum_{i=1}^{m} (p(i) - p^*(i))^2}$$

where the sum is over m point which define the geometry of the bump. $p(i)$ is the bump non-dimensional pressure distribution and $p^*(i)$ is the target surface pressure distribution. Defining the cost function in this way causes a rapid variation near the optimum point and increases the GA rate of convergence.

The GA was run with micro GA option on and maximum generations of 100. The population size was set to 5 and the probability of the mutation was assumed to be 0.02. Figure 1 shows the GA maximum fitness at each generation versus generation number.

Figure 2 shows the pressure distribution over the bump at different generation while Figure 3 depicts the final and target pressure distribution. After 100 generation, it can be said that the problem was fully

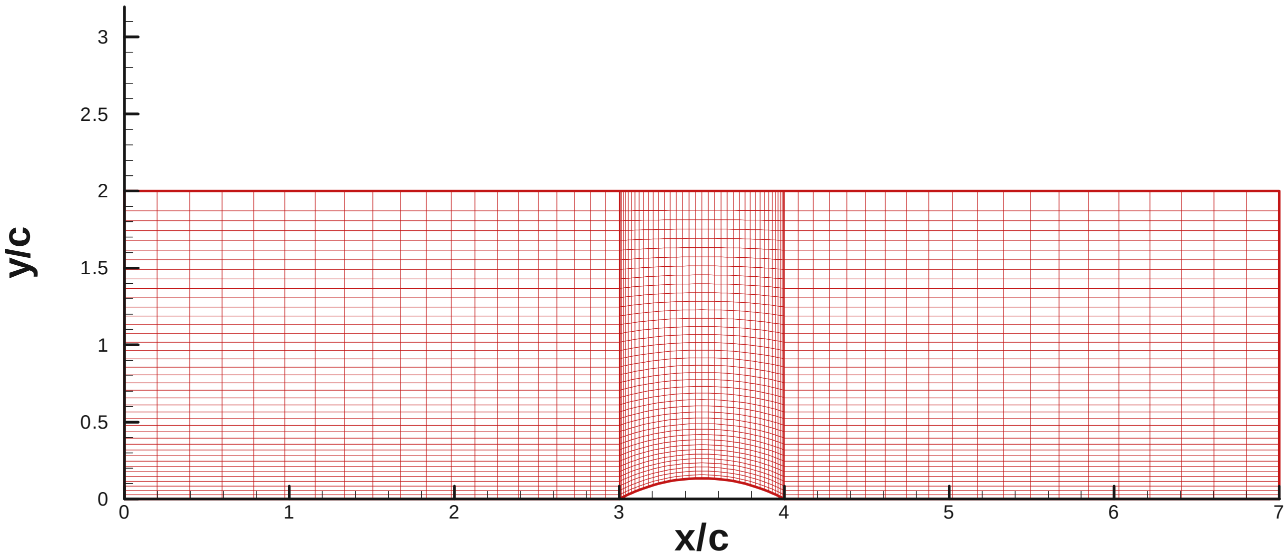

Figure 4. Computational mesh of 60×40 grids for the first case.

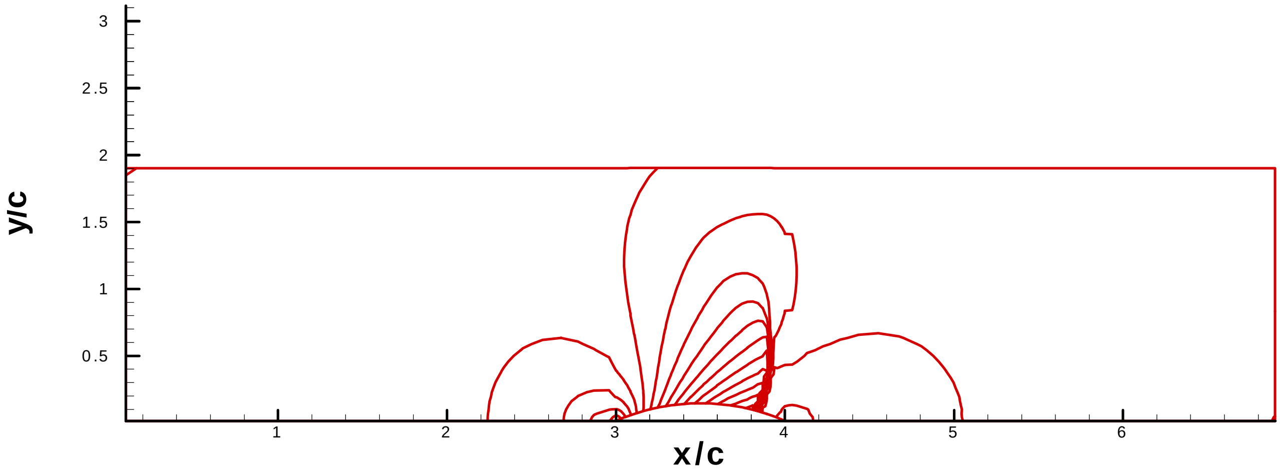

Figure 5. Pressure contours for the first case, $\frac{p_{min}}{p_\infty} = 0.3, \quad \frac{p_{max}}{p_\infty} = 1.3$.

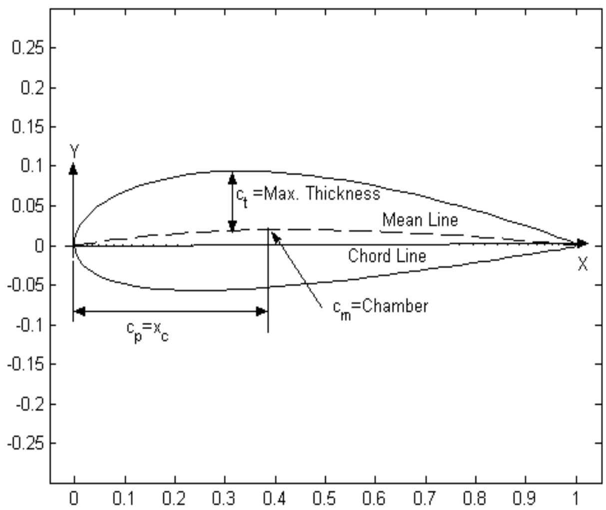

Figure 6. Airfoil parametrisation used for gene encoding.

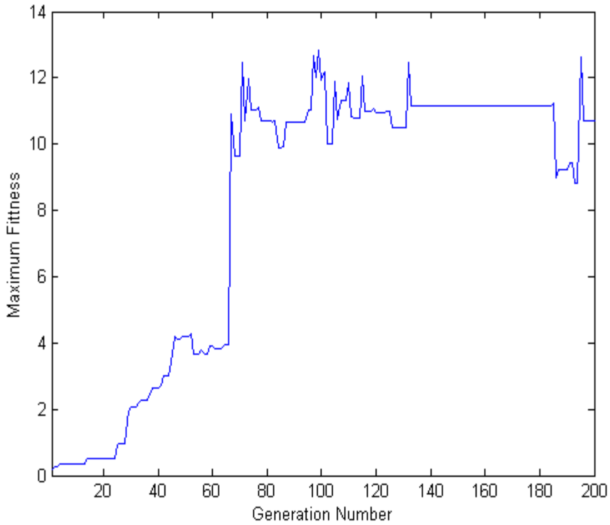

Figure 7. The GA maximum fitness versus generation number for the inverse airfoil design.

converged and the design process has reproduced the target pressure within the specified tolerance.

Figures 4 and 5 show the computational domain and the pressure distribution around the bump at the final value for the bump radius, i.e. r=1.

5-2 Case 2 The GA is next applied to an airfoil in a transonic flow. A two-dimensional implicit flow solver employing a second-order Godunov scheme is used to perform all function evaluations. Each airfoil geometry used in this case study is described by three parameters. The three parameters used for this description, which are also the chromosome used in GA are defined in Figure 6. These parameterization for a NACA four digits airfoil is adopted from ref. 33.

The target solution was the computed pressure distribution for a NACA2415 airfoil at inviscid transonic flow conditions ($M_\infty = 0.8, \ \alpha = 2^\circ$). The maximum/minimum values of each parameter are defined in Table 1. Again the fitness function is

TABLE 1. Maximum/Minimum Values Used for Initialization and GA Processing.

Parameter	Minimum	Maximum	Target Value	Final Value	Error%
c_t	0.05	0.3	0.15	0.1432	4.53
c_m	0.01	0.04	0.02	0.0177	11.5
c_p	0.1	0.7	0.40	0.4001	0.02

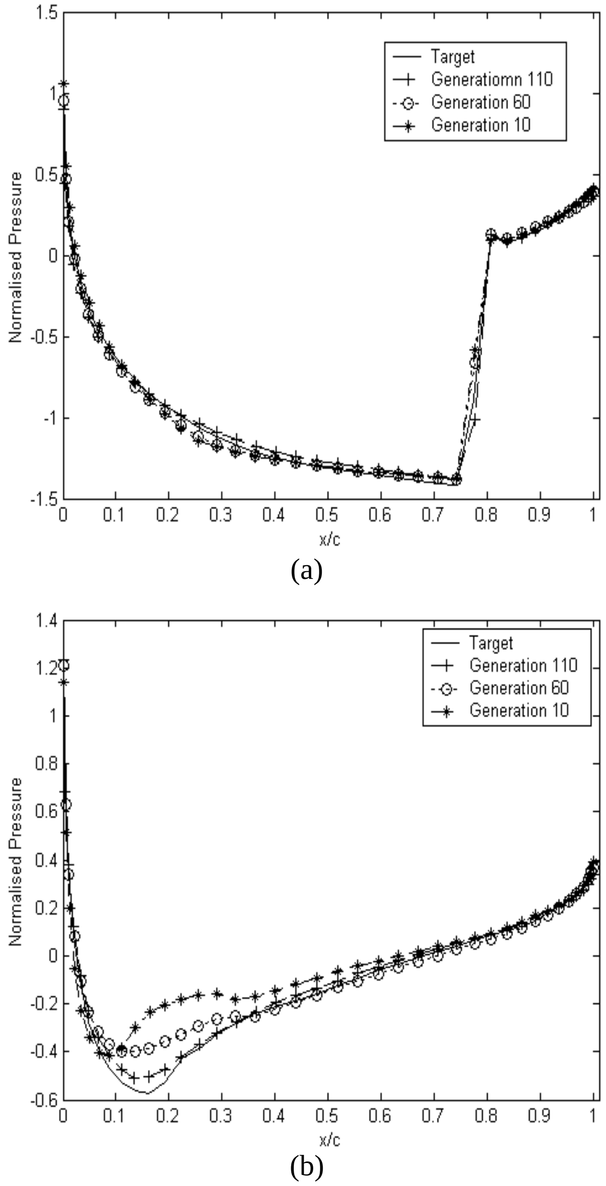

Figure 8. The upper (a) and the lower (b) surface pressure distribution at different generations for the inverse airfoil design.

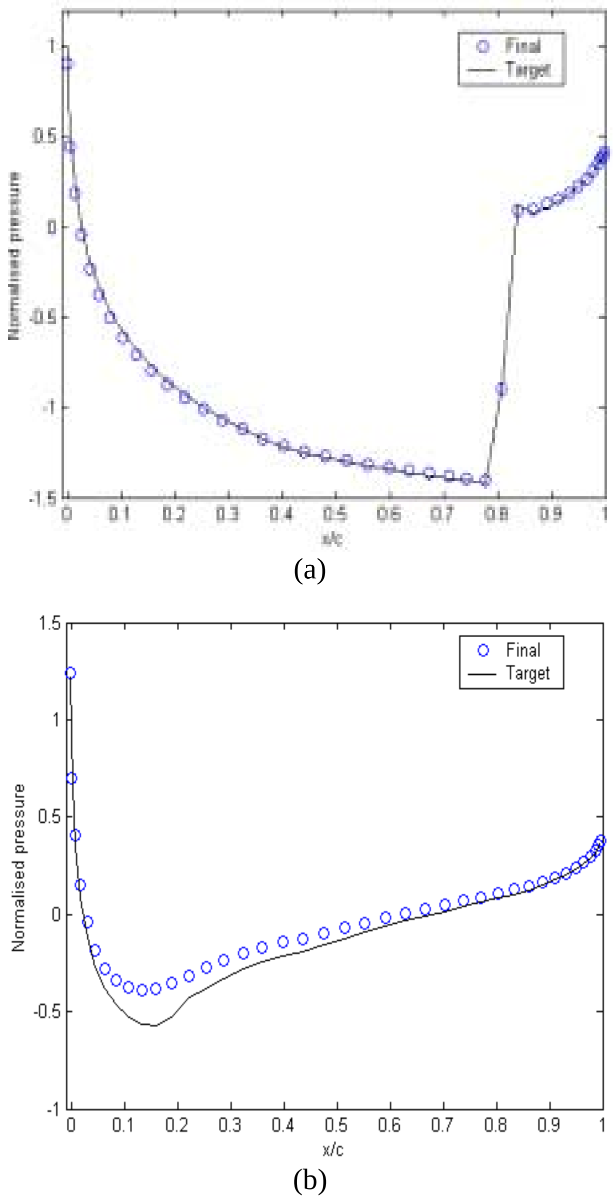

Figure 9. The final and target pressure distribution for the inverse airfoil design (a) upper surface and (b) lower surface.

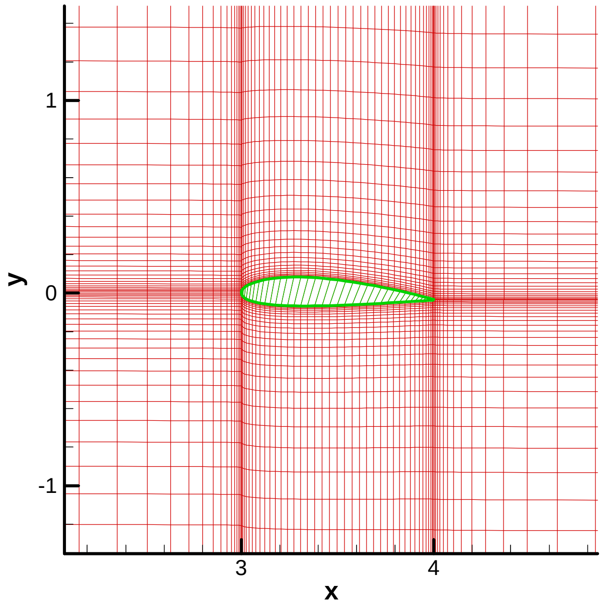

Figure 10a. Computational domain for case 2 Mesh size 60×40.

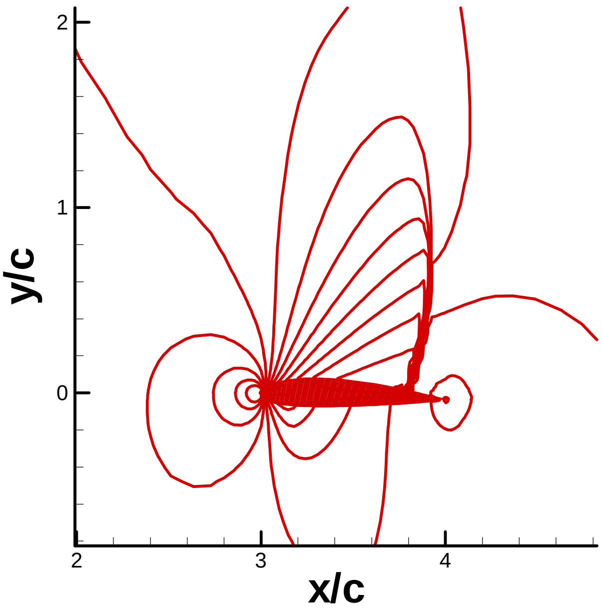

Figure 10b. Pressure contours for case 2 $p_{min}/p_\infty = 0.1$ and $p_{max}/p_\infty = 1.5$.

defined in the same manner as case 1.

The GA was run with micro GA option on and maximum 200 generations. The population size was set to 5 and the probability of the mutation was assumed to be 0.02.

Figure 7 shows the GA maximum fitness at each generation versus generation number. It can be observed that the convergence was reached at the end of 200 generation.

Figure 8 shows the pressure distribution over the upper and lower part of the airfoil at different generation respectively. Figure 9 depicts the final and target pressure distribution for the inverse airfoil design after 200 generation. Again, one can say that the design process has reproduced the target pressure within the specified tolerance.

Another important way to view the performance of the whole procedure is to measure convergence against computer time. It should be noted that the time spent by the CPU for the convergence of the flow solver is about 87% of the total time. However, using a strategy to set initial conditions by the nearest converged solutions reduce this time considerably. Again, the computational domain and the pressure distribution around the airfoil are shown in Figures 10a and 10b, respectively.

6. CONCLUSIONS

In this paper, a genetic algorithm (GA) procedure suitable for aerodynamic design optimization was presented. It uses binary encoding to represent the parameters of the studied geometries as genes in the GA. The uniform mutation, uniform cross over and micro GA options was used to operate GA from one generation to the next.

The developed GA/Euler flow solver code performed well on the presented case studies, namely transonic flow over a bump and a NACA four digits airfoil. Reasonable level of convergence was reached for all cases in several hundreds function evaluations.

The GA/flow solver coupling was quiet easy to set up and only required a few hours for solving the bump problem. Theoretically a wide range of optimization problems can be solved using the developed code with a small implementation effort providing an appropriate objective function.

The results indicate that the GA approach is robust. However, a gradient-based method would probably provide solution to the single-objective and smooth problems in less computer time. Using a progressive methodology explained in the paper, the required convergence time for the solution was reduced by a factor of 5 for the developed GA/Euler flow solver code.

7. ABBREVIATIONS

CFD	Computational fluid dynamics
GA	Genetic algorithm
SA	Simulated annealing

8. NOMENCLATURE

γ	specific heat ratio
ρ	density
Δt	time step
Δx	cell length
E	internal Energy
M_∞	free stream Mach number
p	pressure
p_∞	free stream pressure
u, v	Cartesian velocity components
Q	vector of conserved variables
F	Flux in x direction
G	Flux in y direction
$\overline{Q}_j^n$	Cell average values of conserved variables
$P_n(j)$	Design parameters at generation j
F_{obj}	Objective function (Function to be optimized)

9. REFERENCES

1. Hicks, R. M., Murman, E. M. and Vanderplaats, G. N., "An Assessment of Airfoil Design by Numerical Optimization", ***NASA TM***-3092, July (1974).
2. Vanderplaats, G. N. and Hicks, R. M., "Numerical Airfoil Optimization Using Reduced Number of Design Coordinates", ***NASA TM***-73151, July (1976).
3. Hicks, R. M. and Henne, P. A., "Wing Design by Numerical Optimization", ***AIAA Paper*** 77-1247, (1977).
4. Vanderplaats, G. N., "Approximation Concepts for Numerical Airfoil Optimization", ***NASA TP***-1370, (1979).
5. Lores, M. E., Smith, P. R. and Hicks, R. M., "Supercritical Wing Design Using Numerical Optimization and Comparisons with Experiment", ***AIAA Paper*** 79-0065, (1979).
6. Sobieszczanski-Sobieski, J., "The Case for Aerodynamic Sensitivity Analysis", Presented at the ***NASA/VPI&SU Symposium on Sensitivity Analysis in Engineering***, (Sept. 25-26, 1986), 1125-1142.
7. Reuther, J., "Aerodynamic Shape Optimization Using Control Theory", ***RIACS Report*** 96.09, (1996).
8. Quagliarella, D. and Della Cioppa, A., "Genetic Algorithm Applied to the Aerodynamic Design of Transonic Airfoil", ***AIAA Paper*** 94-1896-CP, (1994).
9. Vicini, A. and Quagliarella, D., "Inverse and Direct Airfoil Design Using a Multiobjective Genetic Algorithm", ***AIAA Journal***, Vol. 35, (1997), 1499-1505.
10. Obayashi, S., Yamaguchi, Y. and Nakamura, T., "Multiobjective Genetic Algorithm for Multidisciplinary Design of Transonic Wing Platform", ***J. of Aircraft***, Vol. 34, (1997), 690-693.
11. Sasaki, D., Obayashi, S., Sawada, K. and Himeno, R., "Multiobjective Aerodynamic Optimization of Supersonic Wings Using Navier-Stokes Equations", ***European Congress of Computational Methods in Applied Science and Engineering, ECCOMAS2000***, Barcelona, Spain, (Sept. 11-14, 2000), 125-143.
12. Oyama, A., "Multidisciplinary Optimization of Transonic Wing Design Based on Evolutionary Algorithms Coupled with CFD Solver", ***European Congress of Computational Methods in applied Science and Engineering, ECCOMAS2000***, Barcelona, Spain, (Sept. 11-14, 2000), 1026-1032.
13. Oyama, A., "Wing Design Using Evolutionary Algorithms", PhD Thesis, Dept. of Aeronautics and Space Engineering, Tohoku University, Senadi, Japan, March (2000).
14. Holst, T. L. and Pulliam, T. H., "Aerodynamic Shape Optimization Using a Real-Number-Encoded Genetic Algorithm", ***AIAA Paper*** 2473, (2001).
15. Obayashi, S. and Tsukahara, T., "Comparison of Optimization Algorithms for Aerodynamic Shape Design", ***AIAA Journal***, Vol. 35, (1997), 1413-1415.
16. Bock, K., "Aerodynamic Design by Optimization", Paper 20, AGARD CP-463, (1990).
17. Doorly, D., Peiro, J. and Spooner, S., "Design Optimization Using Distributed Evolutionary /methods", 37th Aerospace Science Meeting, ***AIAA Paper*** 99-0111, Jan. (1999).
18. Doorly, D. and Peiro, J., "Supervised Parallel Genetic Algorithm in Aerodynamic Optimization", ***AIAA Paper 97-1852***, (2000).
19. Doorly, D., Peiro, J. and Oesterla, J., "Optimization of Aerodynamic and Coupled Aerodynamic-Structural Design Using Parallel Genetic Algorithm", Proc. of 6th ***AIAA/NASA/ISSMD Symposium on Multi-disciplinary Analysis and Optimization***, (1996), 401-409.
20. Van Leer, B.,"Toward the Ultimate Conservative Difference Scheme II, A Second-Order Sequel to Godunov's Method", ***Journal of Computational Physics***, Vol. 32, (1979), 123-134.
21. Ben-Artzi, M. and Falcowitz, J., "A Second Order Godunov-Type Scheme for Compressible Fluid Dynamics", ***Journal of Computational Physics***, Vol. 55, (1984), 534-554.
22. Pishevar, A., "Numerical Simulation of Shock-Wave

Boundary Layer Interactions in a Hypersonic Compression Corner Flow", ***International Journal of Engineering***, Vol. 11, (1998), 197-206.
23. Vanderplaats, G. N., "Numerical Optimization Techniques for Engineering Design", McGraw-Hill Book Co., (1984).
24. Haftka, R. T., and Gurdal, Z., "Elements of Structural Optimization", Kluwer Academic Publishers, (1992).
25. Walsh, G. R., "Methods of Optimization", John Wiley, (1975).
26. Davis (Ed.), L., "Handbook of Genetic Algorithms", Van Nostrand Reinhold, (1991).
27. Goldberg, D. E., "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, Reading, MA, (1989).
28. Holland, J. H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, Michigan, (1975).
29. Corana, A., Marchesi, M., Martini, C., Ridella, S., "Minimizing Multimodal Functions of Continuous Variables with the Simulated Annealing Algorithm", ***ACM Trans. Mathematical Software***, No. 13, Vol. 3, (1987), 262-280.
30. Kirkpatrick, S., Gelatt Jr., C. D., Vecchi, M. P., "Optimization by Simulated Annealing", ***Science***, No. 220, (1983), 671-680.
31. Van Laarhoven, PJM., Aarts, EHL., "Simulated Annealing: Theory and Applications", Dordrecht, The Netherlands, Kluwer Academic Press, (1989).
32. Narducci, R. P., Grossman, B. and Haftka, R. T. "Design Sensitivity Algorithms for an Inverse Design Problem Involving a Shock Wave", Inverse Problem in Engineering, Vol. 2, (1995), 49-83.
33. Abbot, I. H. and von Doenhoff, A. E., "Theory of Wing Section", Dover Publications, New York, (1949).