INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **30**: 149–159 (1999)

MULTIDISCIPLINARY SHAPE OPTIMIZATION IN AERODYNAMICS AND ELECTROMAGNETICS USING GENETIC ALGORITHMS

RAINO A.E. MÄKINEN[a], JACQUES PERIAUX[b] AND JARI TOIVANEN[a,*]

[a] *University of Jyväskylä, Department of Mathematics, P.O. Box 35, FIN-40351 Jyväskylä, Finland*
[b] *Dassault Aviation, Aerodynamics and Scientific Strategy, 78 Quai Marcel Dassault, 92214 St Cloud Cedex, France*

SUMMARY

A multiobjective multidisciplinary design optimization (MDO) of two-dimensional airfoil is presented. In this paper, an approximation for the Pareto set of optimal solutions is obtained by using a genetic algorithm (GA). The first objective function is the drag coefficient. As a constraint it is required that the lift coefficient is above a given value. The CFD analysis solver is based on the finite volume discretization of the inviscid Euler equations. The second objective function is equivalent to the integral of the transverse magnetic radar cross section (RCS) over a given sector. The computational electromagnetics (CEM) wave field analysis requires the solution of a two-dimensional Helmholtz equation which is obtained using a fictitious domain method. Numerical experiments illustrate the above evolutionary methodology on an IBM SP2 parallel computer. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: shape optimization; genetic algorithms; fluid dynamics; electromagnetics

1. INTRODUCTION

In traditional optimal shape design problems in aerospace engineering, only one objective function of one discipline is minimized [1,2]. However, one discipline such as aerodynamics, electromagnetics, etc., is usually not enough to describe the essential properties of the product which is to be optimized. Therefore, it is necessary to consider multidisciplinary problems. For example an airfoil should have certain aerodynamical properties while the radar visibility, i.e. the electromagnetic backscatter, is minimized [3–5]. In multidisciplinary optimization problems, there are usually several conflicting criteria. This kind of problems with several objective functions are called multiobjective optimization problems and their numerical solution require tools different from the standard optimization techniques for single objective optimization [6].

When working with multiobjective optimization problems, it is convenient to define the concept of *Pareto optimality*. For simplicity, only the case of a minimization problem with two objectives is considered. In order to define a relation between two designs, we have a design 1 with objectives $f_1(\alpha_1)$, $f_2(\alpha_1)$ and a design 2 with objectives $f_1(\alpha_2)$, $f_2(\alpha_2)$. Now the design 1 *dominates* the design 2 if $f_1(\alpha_1) \leq f_1(\alpha_2)$ and $f_2(\alpha_1) \leq f_2(\alpha_2)$ and $\exists i \in \{1, 2\}$ such that $f_i(\alpha_1) < f_i(\alpha_2)$.

* Correspondence to: University of Jyväskylä, Department of Mathematics, P.O. Box 35, FIN-40351 Jyväskylä, Finland.

Contract/grant sponsor: The Academy of Finland; Contract/grant number: Grant 34063

CCC 0271–2091/99/100149–11$17.50

A design is said to be *non-dominated* if there is no feasible design in the entire solution space which dominates it. The Pareto set is the set of all such non-dominated designs. Therefore, a Pareto optimal design is optimal in the sense that no other design exists which is better or equal with respect to both objectives.

In multiobjective problems, one interest is to give several Pareto optimal solutions to the decision maker for the selection of the most suitable solution. The finding of several non-dominated solutions is computationally a very laborious task. Therefore, efficient methods and powerful computers are required. Genetic algorithms (GA) are naturally parallel and they can be adapted to this kind of problem. Therefore, we have chosen to employ a GA on a parallel computer and one of the key motivations of this paper is to consider parallel genetic optimization tools implemented on a distributed memory multiprocessor.

This paper is a continuation of development of more realistic shape optimization problems in CFD and CEM. In references [3] and [4], the flow is modeled by the incompressible potential flow. Here we have chosen to use the Euler equations. Also, the problems are usually either multidisciplinary or multiobjective, while we consider a problem which is both at the same time.

One ingredient of our problem, which increases the difficulty, is the non-linear lift constraint. On the other hand we are carrying on the investigation and the development of genetic algorithms on parallel computers for optimal shape design problems. References [5] and [7] are earlier steps in this direction.

The following section describes GAs, their modification for the problem under consideration and a parallel implementation of GAs. Then, the CFD and CEM analysis solvers are shortly introduced. After this the actual multiobjective multidisciplinary optimization problem for the airfoil design is given. In the last sections we present the numerical experiments computed on an IBM SP2 parallel computer together with the concluding remarks.

2. PARALLEL OPTIMIZATION WITH GENETIC ALGORITHM

Genetic algorithms (GAs) are stochastic processes designed to mimic the natural selection based on Darwin's principle of survival of the fittest. J.H. Holland [8] introduced a robust way of representing solutions to a problem in terms of a population of digital chromosomes that can be modified by the random operations of crossover and mutation on a bit string of information. In order to measure how good the solution is, a fitness function is needed. To simulate the process of breeding a new generation from the current one, the following steps are used:

1. Reproduction according to fitness: the better the string is, the more likely it is to be chosen as a parent;
2. Recombination: the parent strings are paired, crossed and mutated to produce offspring strings;
3. Replacement: the new offspring replace the old population.

GAs have been shown to be robust adaptive optimization methods with inherent parallelism for problems where the traditional methods can fail, for example, when searching for a neighborhood of a global minimum [9]. Without any gradient information, they can explore the search space in parallel with the population of individuals and exchange the beneficial information through crossover.

In GAs, the objective functions can be independently evaluated at each generation. On a parallel implementation based on the master–slave prototype, the master computes the genetic

operations and the slaves compute the object function values. Therefore, the amount of communication between the master and the slaves is rather small. A good parallel efficiency for this kind of an approach has been demonstrated [7,5].

For multiobjective optimization problems, it is necessary to make some modifications to the basic GA. There exist several variants of GAs for this kind of problem; see for example Vector Evaluated GA (VEGA) [10] and Non-dominated Sorting GA (NSGA) [11]. For further information on GAs for multiobjective optimization, see [12] and references therein. In the following, we describe the basic ideas of NSGA.

The fitness values are computed using the following procedure:

ALGORITHM: **Non-dominated sorting**

Choose a large dummy fitness value F;
repeat
Find the non-dominated individuals among the individuals whose fitness values are not set;
Set the fitness value of individuals found in previous step to F;
Decrease the dummy fitness value;
until (fitness values of all individuals are set).

An example of the fitness values obtained using the previous procedure is shown in Figure 1. In NSGA, the diversity in the population is maintained using the fitness value sharing [9]. Furthermore, it uses the binary coding, the roulette wheel selection, the classical crossover and mutation operators in the recombination.

The GA which we have developed and used is based on NSGA. Since our key idea is to employ the *tournament selection*, it is necessary to make some modifications. The fitness values are computed exactly in the same way as in NSGA. For each tournament, a fixed number of individuals are selected randomly. The individual which has the highest fitness value wins the

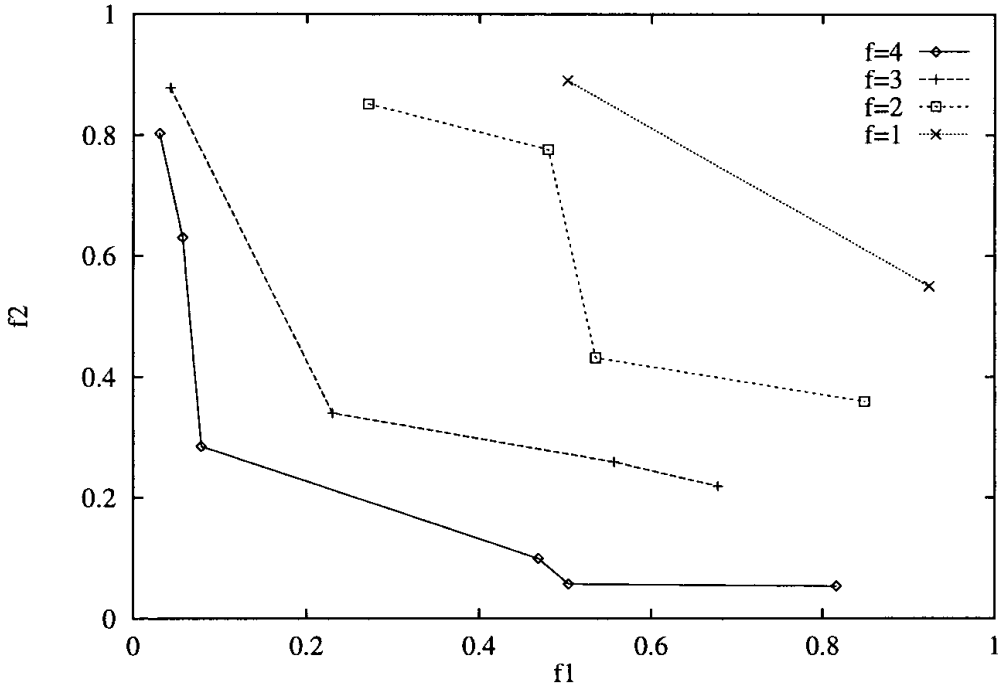

Figure 1. An example of how the fitness values are assigned.

parent 1	2.15	4.13	3.12	1.78	3.34
parent 2	0.83	3.52	1.16	2.75	4.44
child 1	2.15	4.13	1.16	2.75	4.44
child 2	0.83	3.52	3.12	1.78	3.34

Figure 2. An example of the one site floating point crossover.

tournament, i.e. it is selected to be a parent in the breeding. If there are several such individuals then the first one to enter the tournament wins.

Unfortunately, if there were no modifications to the previous tournament selection, the population would usually converge towards one point on the set of Pareto optimal solutions whereas the aim was to obtain several points from the Pareto set. Therefore, some kind of mechanism is required in order to maintain diversity in population. The most obvious way would be to use the fitness value sharing. It has been shown that this approach fails to preserve the diversity in population [13]. Therefore, a modified algorithm is proposed.

Instead of using some previously considered method, we develop a new way to preserve the diversity of the population. We shall call this approach a tournament slot sharing. A sharing function is defined by

$$\mathrm{Sh}(d_{ij}) = \begin{cases} 1 - \left(\dfrac{d_{ij}}{\sigma_{\mathrm{share}}}\right), & \text{if } d_{ij} < \sigma_{\mathrm{share}}, \\ 0, & \text{otherwise,} \end{cases}$$

where d_{ij} is the genotypic distance between the individuals i and j, i.e. in our case the euclidean distance between the vectors defining the designs i and j. The parameter σ_{share} is the maximum *sharing distance* for a tournament slot. This very same sharing function $\mathrm{Sh}(d_{ij})$ is also used in the classical fitness value sharing. Now the probability for the individual i to enter a tournament is computed using the formula.

$$p_i = \frac{1/\sum_{j=1}^{n} \mathrm{Sh}(d_{ij})}{\sum_{k=1}^{n} \left(1/\sum_{j=1}^{n} \mathrm{Sh}(d_{kj})\right)}, \tag{2.1}$$

where the parameter n is the size of population. Hence, this is the same as the roulette wheel selection for the rivals in a tournament. Each individual's slice of roulette wheel is proportional to the inverse of the sum of all sharing functions associated to this individual.

An elitist mechanism is added to our algorithm since it guarantees the cost function values to decrease monotonically from one generation to the next. Also, it usually accelerates the convergence. This is implemented by copying from the old population to the new population all the individuals which would be non-dominated in the new population. Hence, the number of copied individuals varies from a generation to another. As a coding, we have used the floating point coding [14]. This is a rather natural choice, since the design is defined by a

vector of floating point numbers. The crossover is made using one crossover site. Figure 2 shows an example of a crossover.

The mutation utilizes a special distribution promoting small mutations. More precisely, the mutation is performed in the following way for a single string: The floating point numbers of the string under consideration are gone through one by one. Let us assume that x is one of these numbers and it is to be mutated. Let l and u be the lower and upper limits for x. The mutated x is denoted by x_m and it is computed as follows:

1. Set $t = (x - l)/(u - l)$;
2. Compute

$$t_m = \begin{cases} t - t\left(\dfrac{t - \text{rnd}}{t}\right)^p, & \text{rnd} < t, \\ t, & \text{rnd} = t, \\ t + (1 - t)\left(\dfrac{\text{rnd} - t}{1 - t}\right)^p, & \text{rnd} > t, \end{cases}$$

   where rnd is a random number from the closed interval [0, 1];
3. Set $x_m = (1 - t_m)l + t_m u$.

In step 2, the parameter p defines the distribution of the mutation. We call this parameter the *mutation exponent*. If $p = 1$ then the mutation is uniform. The probability of small mutations grows as the value of p grows. Hence, we are ready to present the following GA and the parent selection procedure:

ALGORITHM: **The modified NSGA**

```
Initialize population;
Compute object functions (in parallel);
do i:=2,number_of_generations
   Compute fitness values using non-dominated sorting;
   Compute probabilities for each individual to enter tournament;
   repeat
      Select two parents;
      Form two children using crossover;
   until (new population is full);
   Perform mutation;
   Compute object functions (in parallel);
   Copy individuals from old population according to elitism;
enddo.
```

ALGORITHM: **Select parent**

```
repeat
   Select one individual to tournament using the probabilities p_i in (2.1);
until (tournament is full);
Find best the individual from the tournament according to the fitness values.
```


3. CFD AND CEM SOLVERS

The flow is modeled by the two-dimensional Euler equations, which in conservative form read

$$\frac{\partial W}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=0, \tag{3.1}$$

where the vector of conservative variables W and the flux vectors F, G, are

$$W=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{pmatrix}, \quad F=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uH\end{pmatrix} \quad \text{and} \quad G=\begin{pmatrix}\rho v\\ \rho vu\\ \rho v^2+p\\ \rho vH\end{pmatrix}. \tag{3.2}$$

In (3.2), the following notations have been used: ρ is the density, u and v are the Cartesian velocity components, p is the pressure, E is the total energy and H is the total enthalpy.

The state equation (3.1) is discretized by using the finite volume method. The steady state solution is obtained by an implicit pseudo-time integration. The convergence is accelerated using the multigrid algorithm [15]. The Euler flow analysis solver is called FINFLOW [16].

The wave scattering is modeled by the time-harmonic two-dimensional Maxwell equations, which can be reduced to the Helmholtz equation with the Sommerfeld radiation condition. The solution w is the scattered time-harmonic wave. The Helmholtz equation is restricted into a rectangular domain II and an absorbing boundary condition $L(w)=0$ is introduced. Now the problem reads

$$\begin{cases}\Delta w+\omega^2 w=0 & \text{in } \Pi\backslash\text{airfoil},\\ w=-z & \text{on } S,\\ L(w)=0 & \text{on } \partial\Pi,\end{cases} \tag{3.3}$$

where the constant ω is the wave number, z is the incident wave and L is a second order absorbing boundary condition operator [17].

The discretization is made using the linear finite elements. A fictitious domain method is used to solve the arising system of linear equations [4,18].

4. DRAG AND BACKSCATTER REDUCTION

In this shape optimization problem, we minimize the drag coefficient and the amplitude of the backscattered wave while the lift coefficient must not be less than a given value.

Let U_{ad} be the set of design variable vectors α in $\mathbf{R}^n$ which define the shapes of the geometrically admissible airfoils. The set of the physically admissible designs is defined by

$$U_{\text{ad}}^*=\{\alpha\in U_{\text{ad}}\,|\,C_{\text{l}}(\alpha)\geq C_{\text{l}}^{\min}\},$$

where $C_{\text{l}}=C_{\text{l}}(\alpha)$ is the lift coefficient and $C_{\text{l}}^{\min}$ is the lower bound for the lift coefficient. This problem can be formulated as a multiobjective minimization problem

$$\min_{\alpha\in U_{\text{ad}}^*}\{C_{\text{d}}(\alpha), J(\alpha)\}, \tag{4.1}$$

where $C_d(\alpha)$ is the drag coefficient and $J(\alpha)$ measures the amplitude of the backscattered electromagnetic wave. The function J is defined by the integral

$$J(\alpha) = \int_\Theta |w_\infty(\alpha)|^2 \, ds, \tag{4.2}$$

where Θ is the sector where the backscatter is minimized and w_∞ is the far field pattern computed from the solution w.

The non-linear lift constraint is handled by adding a quadratic penalty function to both object functions. Let ε be a small positive penalty parameter. Then, the penalized object functions are

$$C_d^\varepsilon(\alpha) = C_d(\alpha) + \frac{1}{\varepsilon} (\min \{C_l(\alpha) - C_l^{\min}, 0\})^2$$

and

$$J^\varepsilon(\alpha) = J(\alpha) + \frac{1}{\varepsilon} (\min \{C_l(\alpha) - C_l^{\min}, 0\})^2.$$

Now the penalized multiobjective minimization problem reads

$$\min_{\alpha \in U_{ad}} \{C_d^\varepsilon(\alpha), J^\varepsilon(\alpha)\}. \tag{4.3}$$

5. NUMERICAL RESULTS

The parametrization of an airfoil shape is defined using two Bezier curves [19], one curve for the upper part and another one for the lower part. Each curve has nine control points. The control points on the leading and trailing edges remain fixed and the y-coordinates of the other control points are allowed to change during the optimization process. The first two design variables are the y-coordinates of the Bezier control points directly above and below the leading edge. The next 12 design variables are the sums and the differences of the y-coordinates of the corresponding control points from upper and lower sides of airfoil. The last design variable is the angle of attack. Therefore, the total number of design variables is 15. When the design variables are chosen with this slightly tricky way, it is sufficient to have only box constraints, i.e. upper and lower limits, for the design variables in order to keep the designs feasible.

In the numerical experiments, the airfoils are operating at a transonic Mach number $M_\infty = 0.75$. The discretization for the Euler solver is made using C-type grid with 192×48 grid nodes (128 of them on the airfoil). In the Helmholtz problem, the computational domain is $[-0.1, 1.1] \times [-0.3, 0.3]$ excluding the airfoil. The leading edge of the airfoil is on the origin and the length of the airfoil is one. We have chosen the wave length so that the airfoil is 20 wave lengths long. The mesh is 481×241 rectangular mesh which is locally fitted to the airfoil. Hence, there is 20 nodes per wave length. For the NACA64A410 airfoil a part of the mesh is shown in Figure 3.

During the optimization process, the grid for the Euler solver is depending continuously and smoothly on the design parameters. The FINFLOW Euler solver does not offer a way to solve the adjoint state equation. Hence, it is not possible to obtain the gradients of the drag and the lift coefficients using the sensitivity analysis. Also, it is expensive and difficult to compute accurately the gradient using a difference approximation. Since the meshes for the Helmholtz

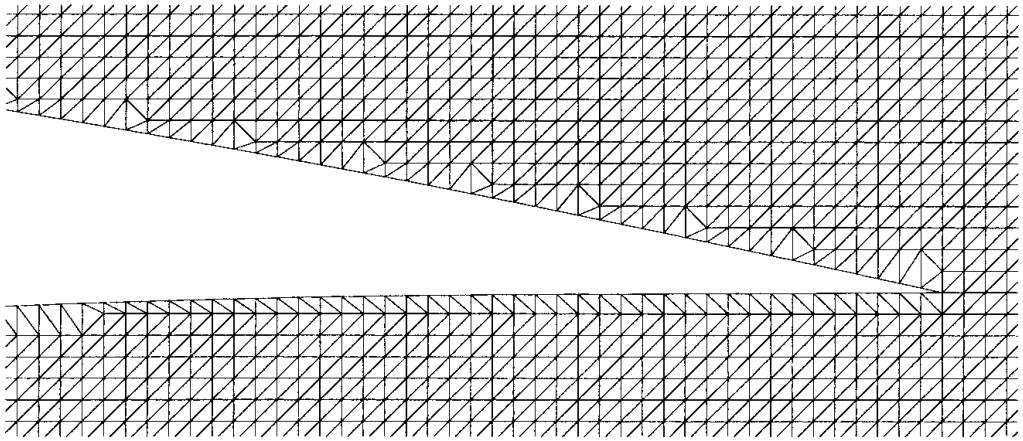

Figure 3. A part of the mesh for the Helmholtz solver near the NACA64A410 airfoil.

solver are made using the local fitting, the number of nodes and elements in the mesh might vary according to design. Therefore, the objective function J computed using the finite element approximation of w is discontinuous.

The lower limit for the lift coefficient $C_l^{\min}$ is set to be 0.5. The backscatter is measured in the sector $\Theta = [180, 200]°$ in (4.2). The direction of the incident wave is 10°. The penalty parameter ε is 10^{-4}. The GA parameters are shown in Table I. In the previous sections, the sharing distance and the mutation exponent are denoted by σ_{share} and p, respectively. From these parameters it is possible to calculate that 6016 fitness function values are computed in one optimization run. The initial population contained the NACA64A410 airfoil and 63 randomly chosen designs.

The computations are made on an IBM SP2 parallel computer using the high performance switch and the MPICH message passing library. We have employed four processors (model 390). The computation of one solution of the Euler equations and the Helmholtz equation required roughly 180 and 30 CPU s, respectively. The total wall clock time for one optimization run was approximately 85 h.

The cost functions of some of the designs in the generations 1, 24, 72 and 94 can be seen in Figure 4. The cost function values $(C_d^\varepsilon, J^\varepsilon)$ for the initial design NACA64A410 are (0.0164, 0.00167). After 94 generations we obtained 18 non-dominated designs. These designs are sorted according to their C_d^ε values and then they are referred using their ordinal number. The corresponding cost function values for the designs 1, 14 and 18 are (0.0025, 0.0157), (0.0046, 0.00129) and (0.0143, 0.00127). The cost functions for these designs, and some of the corresponding airfoils are shown in Figure 7.

In the remaining figures, we have examined three designs, namely the NACA64A410, and the design 1 and 14 from the 18 non-dominated designs in the last generation. The airfoils for

Table I. The parameters in GA

Population size	64
Generations	94
Tournament size	3
Sharing distance	0.25
Crossover probability	0.8
Mutation probability	0.2
Mutation exponent	4

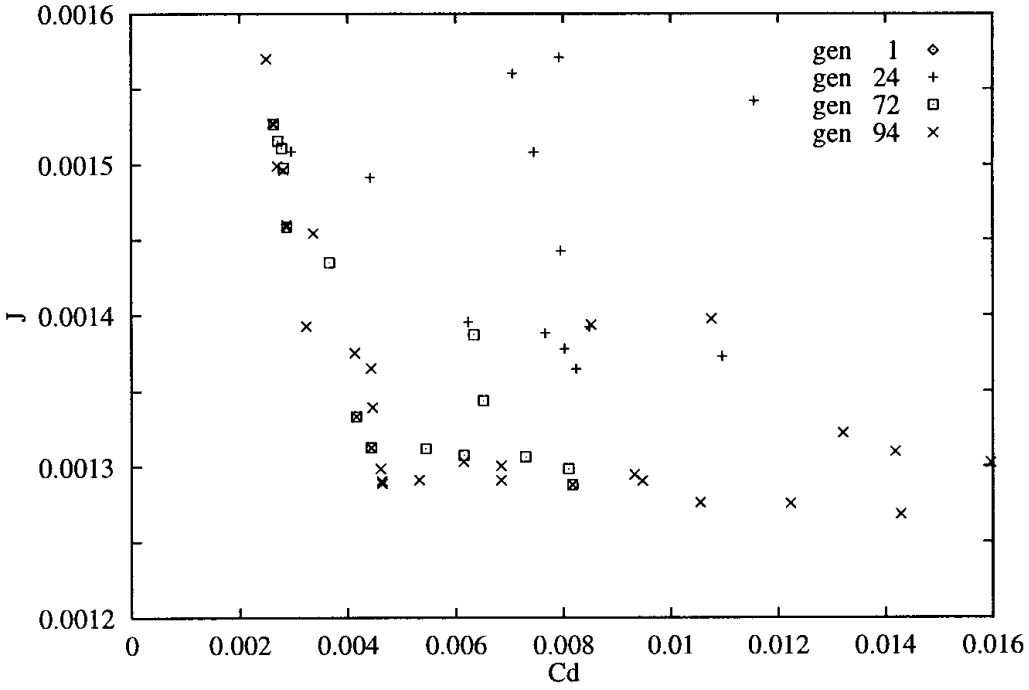

Figure 4. A part of the individuals in the generations 1, 24, 72 and 94.

the design 1 and 14 can be seen in Figure 7. The pressure coefficients C_p are shown in Figures 5 and 6, the radar cross sections (RCSs) are given in the sector where the backscatter is minimized.

The final generation is probably not fully convergent, i.e. there is still a small gap between the non-dominated individuals in the last generation and the set of Pareto optimal solutions. Hence, more generations would probably improve the non-dominated designs. Also due to the elitism mechanism in our GA the number of non-dominated designs should start to grow when

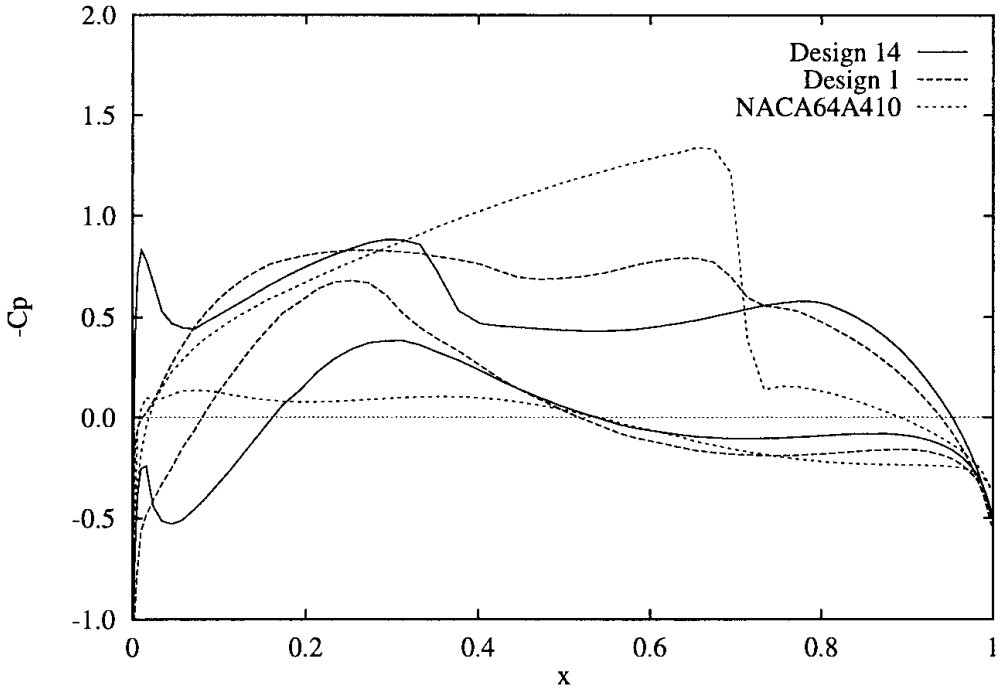

Figure 5. The pressure coefficients for some designs.

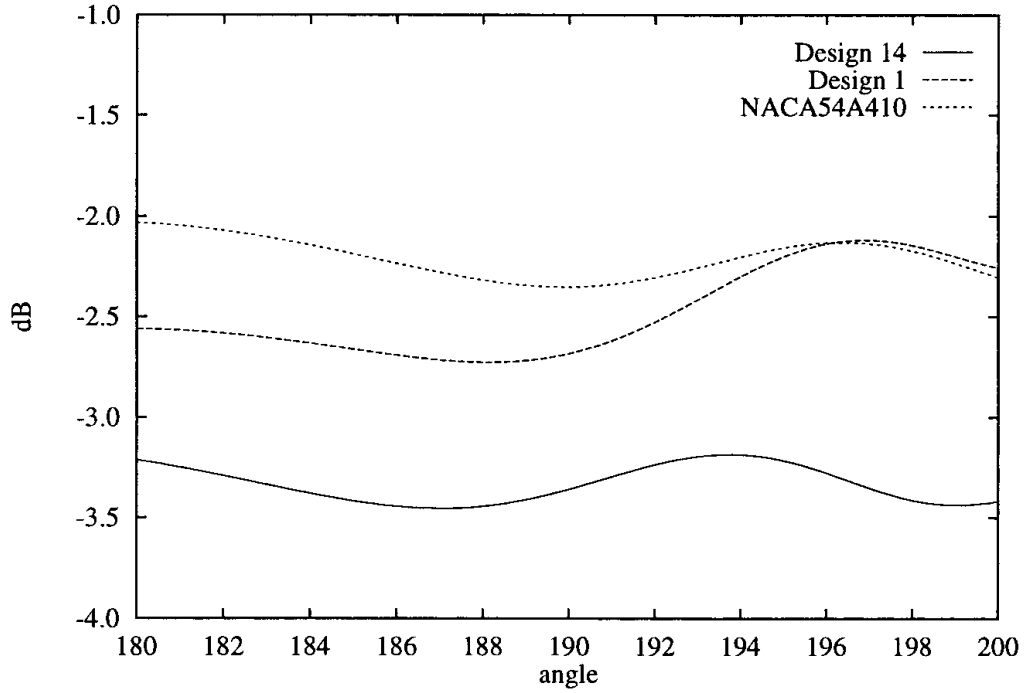

Figure 6. The RCSs for some designs.

the individuals are reaching the Pareto set. Probably the tuning of the GA parameters are likely to accelerate the convergence. Unfortunately, the tuning is rather difficult, since each GA run requires extensive computational time. A cheaper way would be to find the characteristic properties of the cost functions and then make the tuning using similar but simpler functions.

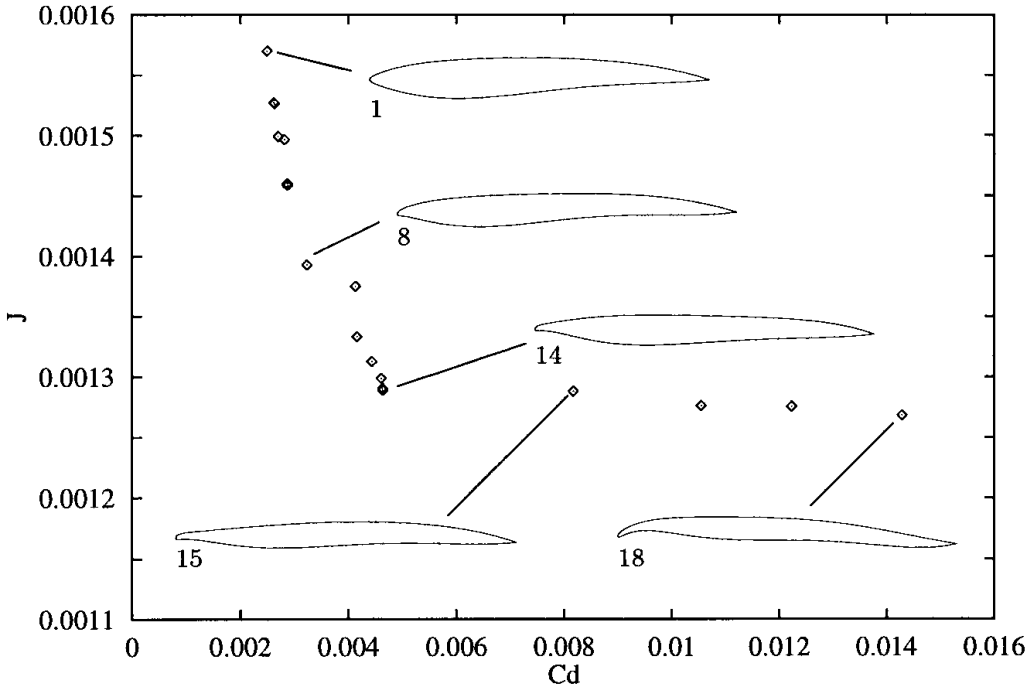

Figure 7. The non-dominated designs from the last generation.

6. CONCLUSIONS

As the numerical results showed, we were able to obtain several non-dominated designs for the decision maker to choose from. Hence, in the future, the design cycle can be reduced using this kind of tool. Since gradients are not required and the cost functions do not have to be continuous, we can use any standard state solvers for shape optimization with our GA. Also, we succeed to get a good parallel efficiency with standard sequential state solvers. The number of performed cost function evaluations was rather high and therefore, the optimization was computationally expensive. In order to reduce the amount of computations, the convergence towards the set of Pareto optimal solutions should be improved. Moreover, a more realistic flow analysis would require the solution of the full Navier–Stokes equations.

ACKNOWLEDGMENTS

This research was supported by the Academy of Finland, grant 34063.

REFERENCES

1. A. Jameson, 'Aerodynamic design via control theory', *J. Sci. Comput.*, **3**, 233–260 (1988).
2. O. Pironneau, *Optimal Shape Design for Elliptic Systems*, Springer, Berlin, 1984.
3. F.J. Baron and O. Pironneau, 'Multidisciplinary optimal design of a wing profile', in J. Herskovits (ed.), *Structural Optimization 93, The World Congress on Optimal Design of Structural Systems*, Vol. II, 1993, pp. 61–68.
4. R.A.E. Mäkinen and J. Toivanen, 'Optimal shape design for Helmholtz/potential flow problem using fictitious domain method', *AIAA Paper 94-4307-CP*, The 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1994, pp. 529–536.
5. R.A.E. Mäkinen and J. Toivanen, 'Parallel solution of optimal shape design problem governed by Helmholtz/potential flow equations', in D.H. Bailey *et al.* (eds), *The 7th SIAM Conference on Parallel Processing for Scientific Computing*, SIAM, Philadelphia, PA, 1995, pp. 102–103.
6. H. Eschenauer, J. Koski and A. Osyczka (eds), *Multicriteria Design Optimization. Procedures and Application*, Springer, Berlin, 1990.
7. R.A.E. Mäkinen, J. Periaux and J. Toivanen, 'Shape design optimization in 2D aerodynamics using genetic algorithms on parallel computers', in S. Taylor, A. Ecer, J. Periaux and N. Satofuka (eds), *Parallel CFD'95*, Elsevier, Amsterdam, 1996, pp. 395–402.
8. J.H. Holland, *Adaptation in Neural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
9. D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
10. J.D. Schaffer, 'Some experiments in machine learning using vector evaluated genetic algorithms', TCGA File No. 00314, *Ph.D Thesis*, Vanderbilt University, Nashville, TN, 1984.
11. N. Srinivas and K. Dep, 'Multiobjective optimization using nondominated sorting in genetic algorithms', *Evol. Comput.*, **3**, 221–248 (1995).
12. C.M. Fonseca and P.J. Fleming, 'Multiobjective optimizers', in H.-M. Voigt, W. Ebeling, I. Rechenberg and H.-P. Schwefel (eds), *Parallel Problem Solving from Nature—PPSN IV, International Conference on Evolutionary Computation*, Vol. 1141, Lecture Notes in Computer Science, Springer, Berlin, 1996, pp. 584–593.
13. C.K. Oei, D.E. Goldberg and S.-J. Chang, 'Tournament selection, niching, and the preservation of diversity', *IlliGAL Report No. 91011*, Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, IL, 1991.
14. Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin, 1992.
15. A. Jameson and S. Yoon, 'Multigrid solution of the Euler equations using implicit schemes', *AIAA Paper 85-0293*, AIAA 23rd Aerospace Sciences Meeting, 1985.
16. J. Hoffren and T. Siikonen, 'FINF2A: a multi-block Navier–Stokes solver for steady two dimensional and axisymmetric flows', *Report B-38*, Laboratory of Aerodynamics, Helsinki University of Technology, 1992.
17. A. Bamberger, P. Joly and J.E. Roberts, 'Second order absorbing boundary conditions for the wave equation: a solution for the corner problem', *SIAM J. Numer. Anal.*, **27**, 323–352 (1990).
18. Yu.A. Kuznetsov and K.N. Lipnikov, 'Fictitious domain method for solving the Helmholtz wave equation for unbounded domain', in Yu.A. Kuznetsov (ed.), *Numerical Methods and Mathematical Modelling*, Institute of Numerical Mathematics of Russian Academy of Sciences, 1992, pp. 56–69 (in Russian).
19. R.H. Bartels, J.C. Beatty and B.A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*, Morgan Kaufmann, Los Altos, 1987.

