Parallel Genetic Solution for Multiobjective MDO

R.A.E. Mäkinen[a], P. Neittaanmäki[a], J. Periaux[b], M. Sefrioui[bc] and J. 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

[c]LAFORIA-IBP-CNRS University Paris VI, 4 place Jussieu, 75552 Paris Cedex 05, France

A multiobjective multidisciplinary design optimization (MDO) of two-dimensional airfoil designs is presented. In multiobjective problems, one interest is to find the set of Pareto optimal solutions. In this paper, an approximation for the Pareto set is obtained by using a genetic algorithm (GA). Since GAs are naturally parallel, it is convenient to use parallel computers on the solution of this very laborious problem. 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 inviscid Euler equations. The discretization is done using a finite volume method and the steady state solution is obtained by an implicit pseudo-time integration with a multigrid acceleration. The second objective function is equivalent to the integral of the transverse magnetic (TM) radar cross section (RCS) over a given sector. The computational electromagnetics (CEM) wave field analysis solver is based on the two-dimensional time-harmonic Maxwell equations. In this case, these equations can be reduced to the Helmholtz equation, which is solved using a fictitious domain method. Numerical experiments illustrate the above evolutionary methodology on a IBM SP2 parallel computer.

1. Introduction

In classical optimal shape design problems, only one objective function is mimized [4], [10]. For many real life decision making problems, there is a need for simultaneous optimization of multiple objectives in order to reduce the design cycle. This kind of problems with several objective functions are called multiobjective optimization problems. In aerodynamics, for example, the drag of an airfoil or an airplane should be minimized and the lift should be maximized [11]. Such multiobjective optimization problems require tools different from the standard optimization techniques for single objective optimization.

Usually, one discipline only such as aerodynamics, electromagnetics, etc., is 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 is minimized [1], [9].

When working with multiobjective optimization problems, it is convenient to define the concept of *Pareto optimality* [13]. 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 $f_1(\alpha_2)$, $f_2(\alpha_2)$. 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 $f_i(\alpha_1) < f_i(\alpha_2)$. A design is said to be *nondominated* if there is no feasible design in the entire solution space which dominates it. The *Pareto set* is the set of all such nondominated 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 nondominated solutions is computationally a very laborious task. Parallel computers offer a considerable computing power. Genetic algorithms (GA) are naturally parallel and they can be adapted to this kind of problems. The key motivation of this paper is to consider parallel genetic optimization tools implemented on a *distributed parallel architecture.*

This paper is a continuation of development of more realistic shape optimization problems in CFD and CEM. In [1] and [9], 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. The nonlinear lift constraint increases the difficulty of our problem. On the other hand, we are carrying on the investigation and the development of genetic algorithms on parallel computers for optimal shape design problems [8].

The following section describes GAs, their modification for the problem under consideration and a parallel implementation of GAs. The following two sections introduce the CFD and CEM analysis solvers. Then, the actual multiobjective MDO problem for the airfoil design is given. In the last sections, the numerical experiments computed on a IBM SP2 parallel computer and the conclusions are presented.

2. Parallel computation with genetic algorithm

Genetic algorithms (GAs) are stochastic processes designed to mimic the natural selection base on the Darwin's principle of survival of the fittest. The basic idea is to represent solutions to a problem in terms of a population of digital chromosomes that can be modified by the random operations of crossover and mutation. The vitality of a solution is given by a fitness function. To simulate the breeding of a new generation, the following steps are used: (1) reproduction according to fitness: the more fit strings are more likely to be chosen as a parent; (2) recombination: the parent strings are paired, crossed and mutated to produce two offsprings; (3) the replacement: the new offsprings form a new generation. Without any gradient information, GAs can explore the search space with the population of individuals and exchange the best findings through crossover [3].

In GAs, the objective functions can be evaluated at each generations independently. 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 is rather small; see Figure 1. Hence, a parallel implementation based on the master-slave prototype can be efficient [8].

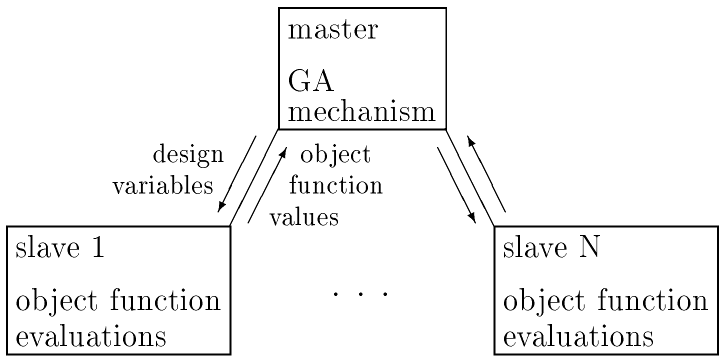

Figure 1. Communication in a parallel implementation of a GA.

For multiobjective optimization problems, it is necessary to make some modifications to the basic GA. One variant of GAs for this kind of problem is Nondominated Sorting GA (NSGA) [12]. In NSGA, the fitness values are computed using the procedure: (1) choose a large dummy fitness value; (2) find the nondominated individuals among the individuals whose fitness value are not set and set the fitness value of those individuals to be the dummy fitness value; (3) decrease the dummy fitness value; (4) if there are still individuals whose fitness value is not set then go to step (2). An example of the fitness values obtained using the previous procedure is shown in Figure 2.

Since we want to employ the *tournament selection*, it is necessary to make some modifications to NSGA, which uses the roulette wheel selection. 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 tournament, i.e., is selected to be a parent in the breeding.

Unfortunately, if there are no modifications to the previous tournament selection, the population would usually converge towards one point on the set of Pareto optimal solutions. The aim was to obtain several points from the Pareto set. We propose the following new way to preserve the diversity of the population. We shall call this approach a tournament slot sharing. The classical sharing function is defined by

$$Sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{\text{share}}}\right)^2, & \text{if } d_{ij} < \sigma_{\text{share}} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where d_{ij} is the genotypic distance between the individuals i and j, i.e., the euclidean distance between the vectors defining the designs i and j in our case. The parameter σ_{share} is the maximum sharing distance for a tournament slot. The probability to the individual i to enter a tournament is computed using the formula

$$p_i = \frac{1/\sum_{j=1}^{n} Sh(d_{ij})}{\sum_{k=1}^{n}(1/\sum_{j=1}^{n} Sh(d_{kj}))}, \tag{2}$$

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.

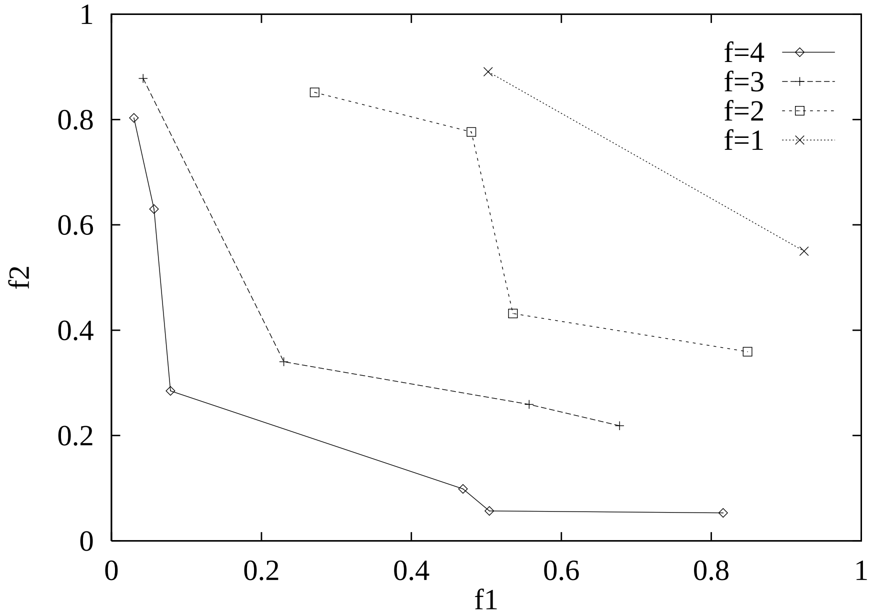

Figure 2. An example of how the fitness values are assigned.

An elitist mechanism is added to our algorithm, since it guarantees a monotonic convergence and it usually accelerates the convergence. This is implemented by copying from the old population to the new population all individuals which would be nondominated in the new population. We have used the floating point coding [7]. This is a natural choice, since the design parameters are floating point numbers. The crossover is done using one crossover site. The mutation utilizes a special distribution promoting small mutations.

3. Euler flow analysis solver

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

$$\frac{\partial W}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0, \tag{3}$$

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

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \quad F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u H \end{pmatrix} \quad \text{and} \quad G = \begin{pmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ \rho v H \end{pmatrix}. \tag{4}$$

In (4), the following notations have been used: ρ density, u and v Cartesian velocity components, p pressure, E total energy and H total enthalpy.

The state equation (3) is discretized by using the finite volume method. The steady state solution is obtained by an implicit pseudo-time integration and the convergence is accelerated using a multigrid algorithm. For more detailed description, see [6].

4. Maxwell field analysis solver

The wave scattering is modeled by the two-dimensional time-harmonic Maxwell equations, which can be reduced to be 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 Π and an absorbing boundary condition $L(w) = 0$ is introduced. The problem then reads

$$\begin{cases} \Delta w + \omega^2 w = 0 & \text{in } \Pi \setminus \text{airfoil} \\ w = -z & \text{on } S \\ L(w) = 0 & \text{on } \partial\Pi, \end{cases} \tag{5}$$

where the constant ω is 2π times the frequency, z is the incident wave and L is a second-order absorbing boundary condition. The discretization is done using linear finite elements. A fictitious domain method is used to solve the system of linear equations [5].

5. Drag and backscatter reduction

In this shape optimization problem, we minimize the drag coefficient and the strength of the backscattered wave while the lift coefficient must be larger or equal than a given value. This problem can formulated as a multiobjective minimization problem

$$\min_{\alpha \in U_{ad}} \{C_d(\alpha),\ J(\alpha)\}, \tag{6}$$

where α is a vector in R^n containing design variables defining the shape of airfoil, C_d is the drag coefficient and J measures the strength of backscattered electromagnetical wave. The function J is defined by the integral

$$J(\alpha) = \int_\Theta |w_\infty(\alpha)|^2 \, ds, \tag{7}$$

where Θ is the sector where the backscatter is minimized and w_∞ is the far field pattern computed from the solution w. In (6), U_{ad} is the set of admissible designs defined by

$$U_{ad} = \left\{ \alpha \in R^n \mid C_l(\alpha) \geq C_l^l \text{ and some geometrical constraints for } \alpha \right\}, \tag{8}$$

where C_l is the lift coefficient and C_l^l is the lower bound for the lift coefficient.

The nonlinear lift constraint is handled by adding a quadratic penalty function to both object functions. Let ε be a small penalty parameter. The penalized object functions are

$$C_d^p(\alpha) = C_d(\alpha) + \frac{1}{\varepsilon} \left(\min\{C_l(\alpha) - C_l^l,\ 0\} \right)^2 \tag{9}$$

and

$$J^p(\alpha) = J(\alpha) + \frac{1}{\varepsilon} \left(\min\{C_l(\alpha) - C_l^l,\ 0\} \right)^2. \tag{10}$$

We assume that the geometrical constraints for designs are box constraints for α, i.e., they are of the form $\alpha_l \leq \alpha \leq \alpha_u$. The penalized multiobjective minimization problem reads

$$\min_{\alpha \in U_{ad}^p} \{C_d^p(\alpha),\ J^p(\alpha)\}, \quad \text{where} \quad U_{ad}^p = \{\alpha \in R^n \mid \alpha_l \leq \alpha \leq \alpha_u\}. \tag{11}$$

6. Numerical results

The parametrization of an airfoil shape is defined using two Bezier curves [2] with nine control points. One curve for the upper half and another for the lower one. The control points on leading and trailing edges remain fixed and the y-coordinates of the other control points are allowed to change. One design variable is the angle of attack. Hence, the total number of design variables is 15.

The airfoils are operating at transonic mach number $M_\infty = 0.75$. The discretization for the Euler solver is done using C-type grid with 192×48 grid nodes. In the Helmholtz problem, the computational domain is $[-0.2, 1.2] \times [-0.4, 0.4]$ excluding the airfoil. The leading edge of airfoil is on origin and the length of airfoil is one which is exactly 10 wave lengths. The mesh is 281×161 rectangular mesh with local fitting to airfoil.

During the optimization process, the grid for the Euler solver is depending continuously and smoothly on the design parameters. Since the meshes for the Helmholtz solver are done using the local fitting, the number of nodes and elements in the mesh might vary according to design. Therefore, the objective function J is discontinuous.

The lower limit for the lift coefficient C_l^l is set to be 0.5. The backscatter is measured in the lower left quadrant, i.e., $\Theta = [180, 270]^\circ$ in (7). The direction of the incident wave is 45°. The penalty parameter ε is 10^{-4}. The GA parameters are shown in Table 1. In one optimization run, 8192 fitness function values are computed. The initial population contained the NACA64A410 airfoil and 63 randomly chosen designs.

Table 1
The parameters in GA.

Population size	64
Generations	128
Tournament size	3
Sharing distance	0.25
Crossover probability	0.8
Mutation probability	0.2

The computations are made on IBM SP2 parallel computer using the high performance switch and the MPICH message passing library. We have employed 8 processors (model 390). Roughly, the computation of one solution of the Euler equations and the Helmholtz equation required respectively 160 and 8 CPU seconds. The total wall clock time for one optimization run was approximately 52 hours.

The cost functions of some of the designs in the generations 1, 32, 64, 96 and 128 can be seen in Figure 3. The cost function values (C_d^p, J^p) for the initial design NACA64A410 are (0.0164, 0.0178). After 128 generations, we obtained 25 nondominated designs. These designs are sorted according to their C_d^p values and then they are referred using their ordinal number. The cost function values for the designs 1, 13 and 25 are respectively (0.0021, 0.0109), (0.0029, 0.0055) and (0.0231, 0.0046). The cost functions for these designs and some of the corresponding airfoils are shown in Figure 4.

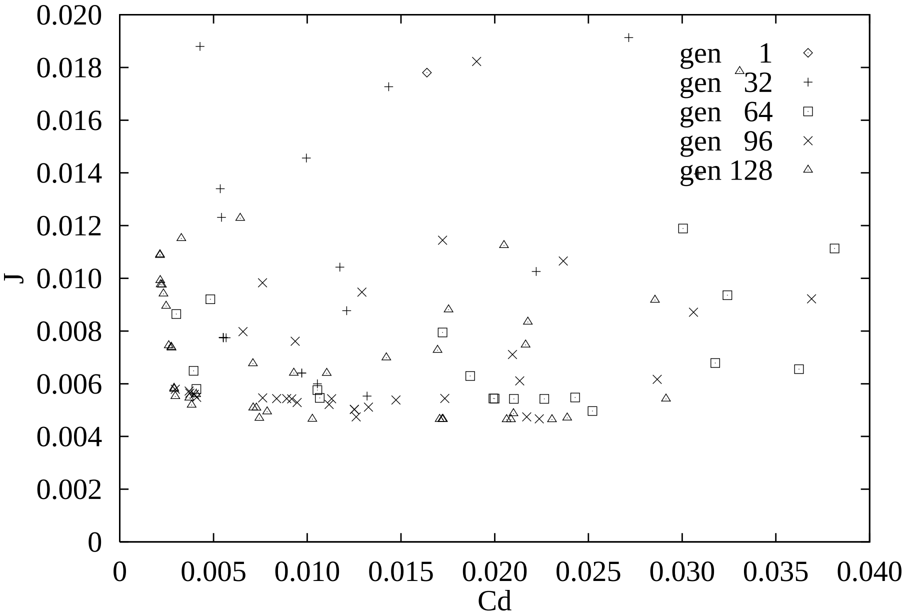

Figure 3. A part of individuals in the generations 1, 32, 64, 96 and 128.

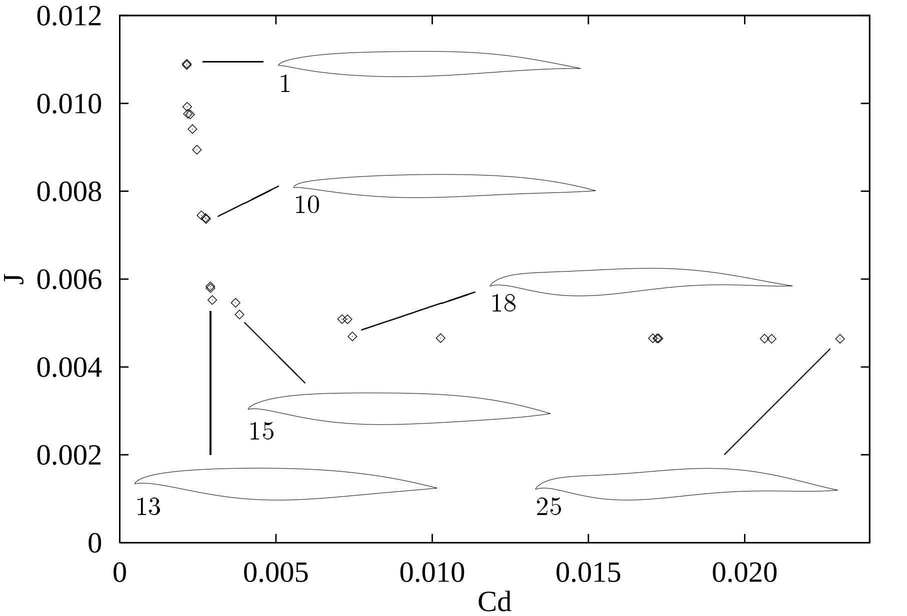

Figure 4. The nondominated designs from the last generation.

7. Conclusions

In the numerical experiments, we were able to obtain several designs for the decision maker to choose from. 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 rather 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. We are aiming to address this in near future.

REFERENCES

1. F. J. Baron and O. Pironneau, *Multidisciplinary optimal design of a wing profile*, Structural optimization 93, The World Congress on Optimal Design of Structural Systems, Vol. II, J. Herskovits (Ed.), 61–68, 1993.
2. 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.
3. D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, 1989.
4. J. Haslinger and P. Neittaanmäki, *Finite element approximation for optimal shape, material and topology design design*, second edition, John Wiley & Sons, 1996.
5. E. Heikkola, Yu. Kuznetsov, P. Neittaanmäki and J. Toivanen, *Numerical studies of different approximations to the Sommerfeld condition: part II*, Report 11, Laboratory of scientific computing, University of Jyväkylä, 1995.
6. J. Hoffren and T. Siikonen, *FINF2A: a multi-block Navier-Stokes solver for steady two-dimensional and axisymmetric flows*, Report B-38, Series B, Laboratory of aerodynamics, Helsinki University of Technology, 1992.
7. Z. Michalewicz, *Genetic algorithms + data structures = evolution programs*, Springer-Verlag, 1992.
8. R. A. E. Mäkinen, J. Periaux and J. Toivanen, *Shape design optimization in 2D aerodynamics using genetic algorithms on parallel computers*, Parallel CFD'95, S. Taylor, A. Ecer, J. Periaux, N. Satofuka (Eds.), Elsevier, 395–402, 1996.
9. 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, 529–536, 1994.
10. O. Pironneau, *Optimal shape design for elliptic systems*, Springer-Verlag, 1984.
11. C. Poloni, *Hybrid GA for multiobjective aerodynamic shape optimization*, Genetic algorithms in engineering and computer science, G. Winter, J. Periaux, M. Galan and P. Cuesta (Eds.), John Wiley & Sons, 397–415, 1995.
12. N. Srinivas and K. Dep, *Multiobjective optimization using nondominated sorting in genetic algorithms*, Evolutionary Computing **3**, 221–248, 1995.
13. R. Steuer, *Multiple criterion optimization: theory, computation, and applications*, John Wiley & Sons, 1986.