JOURNAL OF PROPULSION AND POWER
Vol. 20, No. 4, July–August 2004

# Transonic Axial-Flow Blade Optimization: Evolutionary Algorithms/Three-Dimensional Navier–Stokes Solver

Akira Oyama* and Meng-Sing Liou†
*NASA Glenn Research Center, Cleveland, Ohio 44135*
and
Shigeru Obayashi‡
*Tohoku University, Sendai 980-8577, Japan*

**The development of a high-fidelity aerodynamic design optimization tool based on evolutionary algorithms for turbomachinery is attempted. A three-dimensional Navier–Stokes solver was used for aerodynamic analysis, so that flowfields would be represented accurately and so that realistic and reliable designs would be produced. For efficient and robust design optimization, the real-coded adaptive range genetic algorithm was adopted, and the computation was parallelized and performed on an SGI Origin 2000 cluster to reduce turnaround time. The aerodynamic redesign of the NASA rotor 67 blade demonstrated the superiority of the present method over the conventional design approach, increasing adiabatic efficiency by 2% over the original design. This increase is achieved not only at the design condition, but over the entire operating range. This design optimization method has proven to be suitable for parallel computing. This promising tool is shown to help turbomachinery designers to design higher-performance machines while shortening the design cycle and reducing design costs.**

## Nomenclature

| | | |
|---|---|---|
| $M$ | = | number of generations between population initializations |
| $m_{\text{design}}$ | = | mass flow rate of a design candidate, kg/s |
| $m_{\text{rotor67}}$ | = | mass flow rate of the NASA rotor 67, kg/s |
| $N(0,1)$ | = | probability distribution function |
| $P_{\text{ratio,design}}$ | = | total pressure ratio of a design candidate |
| $P_{\text{ratio,rotor67}}$ | = | total pressure ratio of the NASA rotor 67 |
| $p$ | = | dimensionless pressure |
| $p_i$ | = | $i$th design variable of a design candidate |
| $p_{i,\max}$ | = | upper search boundary of the $i$th design variable |
| $p_{i,\min}$ | = | lower search boundary of the $i$th design variable |
| $pn_i$ | = | $i$th normalized design variable of a design candidate |
| $r_i$ | = | $i$th string of a design candidate |
| $s$ | = | dimensionless entropy |
| $\gamma$ | = | gas specific heat ratio, 1.4 |
| $\mu_i$ | = | average of the $i$th design variable |
| $\rho$ | = | dimensionless density |
| $\sigma_i$ | = | standard deviation of the $i$th design variable |
| $\sigma_{\text{present}}$ | = | standard deviation of the present generation |
| $\sigma_{\text{sampled}}$ | = | standard deviation of the sampled design candidates |
| $\sigma_{\text{updated}}$ | = | standard deviation of the next generation |
| $\omega$ | = | relaxation factor |

Presented as Paper 2002-5642 at the AIAA/ISSMO 9th Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, 4 September 2002; received 7 May 2003; revision received 8 February 2004; accepted for publication 11 February 2004. 

*National Research Council Research Associate, Turbomachinery and Propulsion System Division; currently Research Associate, Division for Space Transportation Engineering, Japan Aerospace Exploration Agency Institute of Space and Astronautical Science, 3-1-1 Yoshinodai, Sagamihara, Kanagawa 229-8510, Japan; oyama@flab.eng.isas.jaxa.jp. Member AIAA.

†Senior Scientist, Turbomachinery and Propulsion System Division, MS 5-11, 21000 Brookpark Road; meng-sing.liou-1@nasa.gov. Associate Fellow AIAA.

‡Professor, Institute of Fluid Science, 2-1-1 Katahira, Aoba, obayashi@ieee.org. Associate Fellow AIAA.

## Introduction

THE compressor is a critical part in developing a new aeroengine because a small improvement in efficiency can result in significant savings in the annual fuel costs of an aircraft fleet. Although today's aeroengine compressors have achieved very high performance, there is an increasing demand for new compressor designs with even higher performance.

One approach to further improve compressor performance is to develop a computer-based design system based on a high-fidelity flow solver such as three-dimensional Navier–Stokes analysis and a numerical design optimization method. Currently, state-of-the-art blade design systems depend on the axisymmetric through-flow method in the initial stage of the blade shape design. High-fidelity computational fluid dynamics (CFD) may also be used, but often CFD is used only for validation purposes or for evaluating the loss coefficient to be used for the next through-flow calculation. Then, design experts rely on their experience and intuition to optimize the blade design manually through a trial-and-error process.

Such a conventional approach is, however, approaching its limits. One reason for this is that the through-flow method cannot capture complicated flow structures within a compressor blade pass, such as secondary flow and shock/boundary-layer interactions that may produce substantial flow losses. Another reason is that it is very difficult to solve a compressor blade design by trial and error because it involves a large number of design parameters and constraints and because the objective functions are inherently nonlinear and multimodal in nature. There is, therefore, a demand for a computational design optimization system based on three-dimensional Navier–Stokes computations coupled with a numerical design optimization method.

Among the many numerical optimization methods, the gradient-based methods, which are classified as local optimization methods, are perhaps most widely used. (For example, see Ref. 1) These methods start with a single design point and use the local gradient of the objective function with respect to changes in the design variables to determine a search direction by using methods involving gradients, such as the steepest descent, conjugate gradient, quasi-Newton techniques, or adjoint formulations. These methods are efficient and can find a true optimum as long as the objective function is differentiable and convex (Kuhn–Tucker condition). However, aerodynamic

design optimization problems usually have nondifferentiable and multimodal objective functions.[2] Optimizations depending on such methods could lead to a local, but not necessarily a global, optimum close to the starting point. Furthermore, such computations easily get bogged down when many constraints are considered.

On the other hand, evolutionary algorithms [(EAs) which is the collective name of evolution-based optimization algorithms such as genetic algorithms, evolutionary strategies, and evolutionary programming] are classified as global optimization methods. They start with multiple points sprinkled over the entire design space, and they search for true optimums according to the objective function values instead of the local gradient information by using the unique operations of EAs, that is, selection, recombination, and mutation. These features lead to robustness, suitability for parallel computing, and simplicity in coupling various evaluation codes. In addition, EAs sample various Pareto-optimal solutions in parallel when applied to a multiobjective design optimization problem. Comparison of various optimization methods has shown that EAs lead to a better result in general.[2–4] Because of these advantages, EAs have become increasingly popular in a broad class of design problems.[5,6] EAs also have been applied successfully to turbomachinery design optimization problems such as compressor cascade design,[7] turbine cascade design,[8] blade design,[9] centrifugal compressor design,[10,11] multistage compressor design,[12] and rocket engine pumps.[13]

For the previous applications of EAs to turbomachinery design optimizations, however, either the optimizations have been restricted to a two-dimensional analysis, or simplified geometry modeling has been used to reduce the number of design parameters. To represent the flowfield in a compressor accurately and produce reliable designs, designers must use a three-dimensional Navier–Stokes computation for aerodynamic performance evaluations. For the present method, we adopted parallel computing to reduce the significant computational time that was expected, especially the turnaround time. The parallel computing was performed on the SGI Origin 2000 cluster (Silicon Graphics, Inc.) at the Institute of Fluid Science, Tohoku University, in Japan. To reduce the number of required aerodynamic evaluations, we adopted a real-coded adaptive range genetic algorithm (ARGA). The aerodynamic redesign of the NASA rotor 67 demonstrates the superiority of the EA-based high-fidelity design optimization approach.

## Design Optimization Problem

The optimization problem considered was a redesign of the NASA rotor 67 (Ref. 14). This low-aspect-ratio, transonic axial-flow fan rotor is the first-stage rotor of a two-stage fan. The fan was designed and tested in the 1970s to help develop efficient, lightweight engines for short-haul aircraft. Rotor 67 was designed by the use of a streamline-analysis computational procedure, which provides an axisymmetric, compressible-flow solution to the continuity, energy, and radial equilibrium equations.

The rotor design pressure ratio is 1.63 at a mass flow of 33.25 kg/s, and the design rotational speed is 16,043 rpm, which yields a tip speed of 429 m/s and an inlet tip relative Mach number of 1.38. The rotor has 22 blades and an aspect ratio of 1.56 (based on the average span/root axial chord). The rotor solidity varies from 3.11 at the hub to 1.29 at the tip, and the inlet and exit hub-to-tip-radius ratios are 0.375 and 0.478, respectively. The Reynolds number is $1.797 \times 10^6$ on the basis of the blade axial chord at the hub.

The objective of the rotor shape design optimization problem was to minimize flow losses. Hence, the objective function was the mass-averaged entropy production from the inlet to the exit at the design point of rotor 67. Local entropy in the computing domain was computed from the local pressure and density:

$$s = p/\rho^{\gamma} \tag{1}$$

Because an optimized rotor design should meet the required mass flow rate and pressure ratio, these parameters were maintained by the specification of constraints on them:

$$|(m_{\text{design}} - m_{\text{rotor67}})/m_{\text{rotor67}}| \leq 0.005 \tag{2}$$

$$|(P_{\text{ratio,design}} - P_{\text{ratio,rotor67}})/P_{\text{ratio,rotor67}}| \leq 0.01 \tag{3}$$

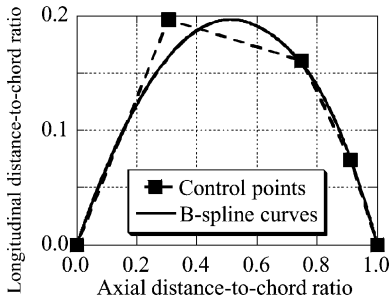


**Fig. 1 Control points and B-spline curves for a mean camber line.**

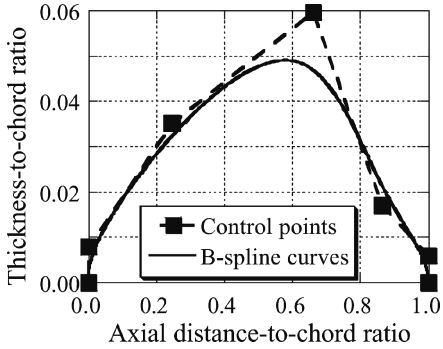


**Fig. 2 Control points and B-spline curves for a thickness distribution.**

## Blade Shape Parameterization

The rotor blade shape under investigation was represented by four blade profiles, at spanwise stations of 0, 31, 62, and 100%. (All spanwise locations discussed here were measured from the hub.) The shape was linearly interpolated between the profiles from hub to tip. Each of these sectional profiles can be uniquely defined by the use of a mean camber line, together with a thickness distribution described at a set of controlled points. These control points were used as parameters in the third-order B-spline curves to yield a smooth representation of the blade camber and thickness. Parameterization with B-spline curves is one of the most popular approaches for airfoil designs.

When B-spline curves are used for shape parameterization, the positions of the control points of the curves are often considered as the design parameters. Here, five control points were used for the mean camber line, as shown in Fig. 1. For the thickness distribution, two more control points were added at the leading and trailing edges, so that these points would represent the leading- and trailing-edge radii (Fig. 2). Chordwise locations of the control points at the leading and trailing edges were frozen at 0 and 1, respectively. As a result, 14 design parameters (six for the mean camber and eight for the thickness distributions) were required to represent a sectional shape, and each blade shape was then represented by four sections, resulting in a total of 56 design parameters.

## Flow Solver

Three-dimensional grids were obtained by stacking two-dimensional grids generated on the blade-to-blade surface. These were C-type two-dimensional grids that were generated by solution of an elliptical equation with a prescribed grid spacing and orientation at the wall. To reduce grid skewness due to high stagger or large camber significantly, we chose to adopt the approach proposed in Ref. 15 by allowing the grid to be nonperiodic at the wake (Fig. 3d). By adding lines near the wall, we could obtain viscous grids from the inviscid grids. The wall normal spacing scaled with the axial chord was $10^{-4}$. In the spanwise direction, a standard H-type structure was adopted. Near the hub and tip walls, geometric stretching was used for a specified number of grid points, after which the spanwise spacing remained uniform. There were 201 chordwise, 53 tangential, and 57 spanwise grid points. Among the 201 chordwise grid points, 149 grid points were distributed along the blade surface. The computational grid for the NASA rotor 67 is shown in Fig. 3.

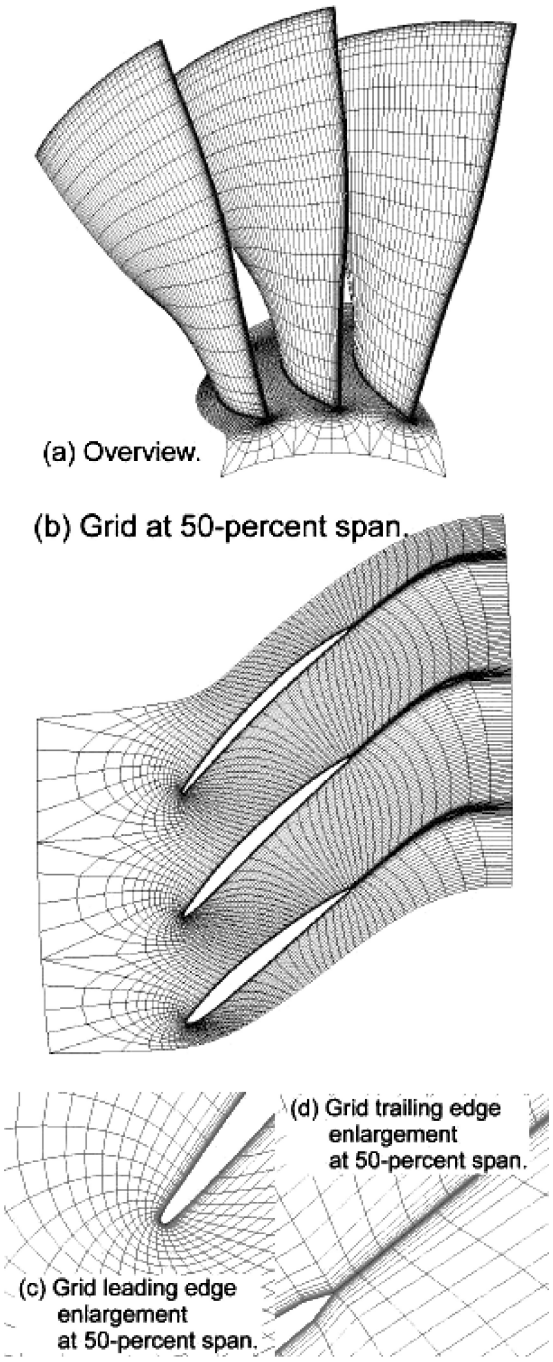


**Fig. 3 Computational grid over NASA rotor 67 (every other line is shown).**

The three-dimensional Navier–Stokes code used in this research was TRAF3D (Refs. 16 and 17), which solved the three-dimensional, full Reynolds-averaged Navier–Stokes equations. TRAF3D uses a central-differencing scheme, including the artificial dissipation terms introduced by Jameson et al.[18] to maintain stability and to prevent oscillations near shocks or stagnation points. The eigenvalue scalings of Martinelli and Jameson[19] and of Swanson and Turkel[20] were incorporated to minimize the amount of artificial diffusion inside the shear layer. The two-layer eddy-viscosity model of Baldwin and Lomax was adopted for the turbulence closure. The system of differential equations uses an explicit, four-stage Runge–Kutta scheme to advance in time. To accelerate the convergence of calculations, the code adopts strategies like local time stepping, implicit residual smoothing,[21] and the full approximation storage multigrid technique.[22] Details about the TRAF3D code and its validations, especially for turbomachinery flows, have been well documented elsewhere, for example, Refs. 15 and 16, and, hence, are not repeated here.

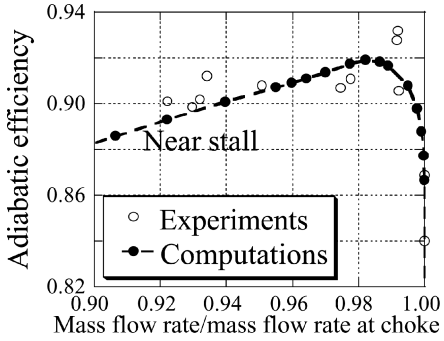


**Fig. 4 Comparison of computed and measured adiabatic efficiency.**

There are four types of boundaries in cascade calculations: inlet, outlet, solid walls, and periodicity. At the inlet, the boundary layers on the hub and tip end walls were accounted for by giving a total pressure and a total temperature profile to mimic the experimental situation. According to the theory of characteristics, the flow angles, total pressure, total temperature, and isentropic relations were used at the subsonic axial inlet, whereas the outgoing Riemann invariant was taken from the interior. At the subsonic axial outlet, the static pressure at the hub was prescribed, and the density and components of velocity were extrapolated together with the circumferential distribution of pressure. The radial equilibrium equation was used to determine the spanwise distribution of the static pressure. On the solid surfaces, the momentum equation, the no-slip condition, and the adiabatic-wall condition were used to find pressure, velocity components, and temperature. The periodicity from blade passage to blade passage was imposed at periodic phantom cells. At the wake, where the grid was not periodic, the phantom cells overlapped the real ones (Fig. 3d). Linear interpolation was then used to compute the value of the dependent variables in the phantom cell.

The capabilities of the present code were validated by a comparison of the computed results with measurements such as those in the Goldman annular vane with and without end wall contouring, the low-speed Langston linear cascade (see Ref. 16) and the NASA rotor 67 (Ref. 17). Figure 4 compares the computed adiabatic efficiency of the NASA rotor 67 with the experimental data. The computed and experimental mass flows were normalized by their respective choking mass flows, as suggested by Pierzga and Wood.[23] This normalization removes any uncertainties in the experimental mass flows. The measured choking mass flow was 34.96 kg/s, whereas the computed value was 34.39 kg/s, a difference of 1.6%. The computed pressure ratios and efficiencies agree very well with the experimental data.

## EAs

In this study, the real-coded ARGA by Oyama et al.[24] was used to improve the robustness and efficiency of the EAs. The real-coded ARGA incorporates the idea of dynamic coding with floating-point representation.

In standard real-coded genetic algorithms, the $i$th real design variable $p_i$ of a design candidate defined in a region $[p_{i,\min}, p_{i,\max}]$ is encoded as a real string $r_i$:

$$r_i = p_i \tag{4}$$

where $p_{i,\min} \le r_i \le p_{i,\max}$. Alternatively, normalized values of the design variables can be used:

$$r_i = (p_i - p_{i,\min})/(p_{i,\max} - p_{i,\min}) \tag{5}$$

where $0 \le r_i \le 1$.

In the real-coded ARGA, the design variable $p_i$ is rewritten implicitly in terms of the real numbers $r_i$ defined in (0,1), so that the integral of the probability distribution of the normal distribution

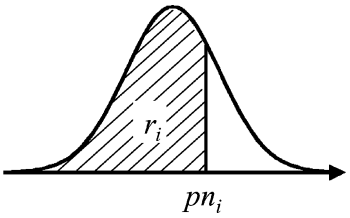


**Fig. 5 Schematic view of decoding of the real-coded ARGA.**

from negative infinity to $pn_i$ is equal to $r_i$:

$$r_i = \int_{-\infty}^{pn_i} N(0,1)(z)\,\mathrm{d}z \tag{6}$$

$$p_i = \sigma_i \cdot pn_i + \mu_i \tag{7}$$

where the probability distribution $N(0,1)$ is scaled, so that the integral from negative infinity to infinity equals 1. The average $\mu_i$ and the standard deviation $\sigma_i$ of each design variable are calculated when the top one-half of the previous population is sampled, so that the present population distributes in the promising search regions. A schematic view of this coding is given in Fig. 5. The update of $\mu_i$ and $\sigma_i$ for every generation, however, results in an inconsistency between the actual and updated population statistics in the next generation because the selection operator picks up the genes that correspond to the promising region according to the old population statistics. To prevent this inconsistency, the present real-coded ARGA updates $\mu$ and $\sigma$ every $M$ generations, and then the population is reinitialized. The real-coded ARGA maintains diversity in the population by reinitializing the genes of the design candidates. To avoid premature convergence and to improve the robustness of the present real-coded ARGA further, we introduced a relaxation factor $\omega_\sigma$ to update the standard deviation:

$$\sigma_{\text{updated}} = \sigma_{\text{present}} + \omega_\sigma(\sigma_{\text{sampled}} - \sigma_{\text{present}}) \tag{8}$$

where the relaxation factor is usually chosen to be between 0.1 and 0.5.

In the real-coded ARGA, the advantages of dynamic coding and floating point representation are both used to overcome the problems of having a large search space that requires continuous sampling. A chromosome is coded as a finite length string of real numbers corresponding to the design variables. The floating-point representation is robust, accurate, and efficient because it is conceptually closest to the real design space, and the string length reduces to the number of design variables. In addition, the real-coded ARGA enables an efficient and robust search while keeping the string length small by adapting the population toward promising regions during the optimization process. Moreover, the real-coded ARGA eliminates the need for the earlier definition of search boundaries because it distributes solution candidates according to the normal distribution of the design variables in the present population. The real-coded ARGA has been successfully applied to airfoil design optimization,[24] wing design optimization,[25] and multiobjective supersonic transport design optimizations.[26]

In the following paragraphs, we describe the steps used in this EA. Figure 6 shows a flowchart of the present real-coded ARGA.

First, we generated the initial population by randomly spreading solutions over the entire initial design space.

Second, the fitness (objective function) of each design candidate was evaluated by its rank as suggested by Michalewicz.[27] The constrained domination approach[28] was used to determine the rank among the population of design candidates according to the objective function value and the extent of the constraint violation. The constrained domination approach handles design constraints without using the penalty function. In addition, this approach maintains sufficient selection pressure throughout the optimization.

Third, the potential parents of the next-generation design candidates were selected via the stochastic universal sampling method,[29] by which the selection probability of each individual was assigned on the basis of its fitness. Also, the two potential parents were selected at the same time to avoid a stochastic error, the so-called genetic drift.[30]

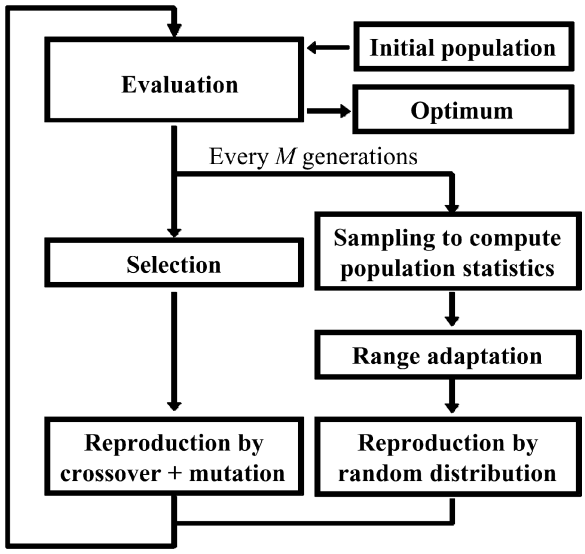


**Fig. 6 Flowchart of the real-coded ARGA.**

Finally, blended crossover (BLX-$\alpha$) and uniform mutation were used to produce the next generation of design candidates. BLX-$\alpha$ is the most common approach for the recombination of two parents represented by a vector of real numbers, as proposed by Eshelman and Schaffer.[31]

In addition, the present EA adopts the elitist strategy,[32] in which the best and second-best individuals in each generation are transferred into the next generation without any recombination or mutation. The elitist strategy guarantees a monotonic improvement in the objective function value.

## Parallelization

The main concern about the use of a three-dimensional Navier–Stokes solver to achieve an optimal aerodynamic compressor design is the required computational effort. Fortunately, this concern is diminishing rapidly because powerful parallel computers are advancing quickly and are becoming increasingly affordable for many institutions and universities. Moreover, personal computer clusters are emerging as a powerful and affordable alternative. This trend especially favors the use of EAs because they are intrinsically amenable to parallel computations. Hence, although the issue of computational cost is diminishing, the ability of the application of EAs to complex problems is increasing.

In this study, all computations were performed on the SGI Origin 2000 cluster at Tohoku University, which consists of 640 processing elements (PEs). The cluster's total scalar performance and total memory size are 384 giga-floating point operations per second and 640 GB, respectively.

In each generation, the aerodynamic evaluation of design candidates was parallelized with the simple master–slave concept: The grid generations and the flow calculations associated with the design candidates of a generation were distributed into the PEs of the SGI cluster. The corresponding parallel efficiency was expected to be almost 100% because of the following factors: 1) Grid generation and the Navier–Stokes computation of each design candidate of each generation were computed independently. 2) The time for the CFD computation of each design candidate was essentially identical. 3) Each Navier–Stokes computation took about 12 h of computational time on an SGI Origin 2000 PE, whereas the real-coded ARGA used only a few seconds for each generation.

## Results

The first step of this EA-based design optimization was to determine the input parameters such as population size and number

**Table 1 Input parameters for the current design optimization**

| Input parameters | Input parameter values |
|---|---|
| Population size | 64 |
| Number of generations | 100 |
| Relaxation factor $\omega$ | 0.2 |
| Generations to adaptation $M$ | 4 |
| Crossover | BLX-0.5 |
| Mutation rate, % | 10 |

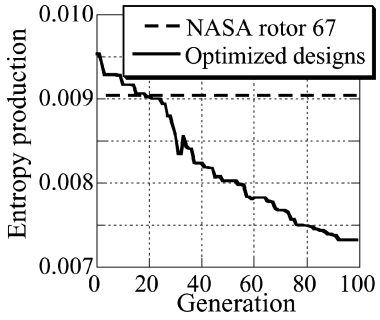


**Fig. 7 Optimization history in terms of entropy production.**

of generations. These parameters (Table 1) were chosen judiciously according to our previous experience.

The next step was to define the design space properly. Even though the real-coded ARGA does not require definition of the design space, it still needs the initial distribution of the design candidates. In this study, the NASA rotor 67 was used as a baseline around which the initial candidates were populated. Specifically, the central values of the initial design space were made to correspond to the design parameter values representing the rotor 67 geometry. These values were found by the minimization of the geometry difference from rotor 67 by the use of the EA without any flow computation.

In this optimization, both the grid generation and Navier–Stokes computation for each design candidate of each generation were distributed into 64 PEs of the SGI cluster and computed in parallel. Although the parallel efficiency was expected to be almost 100%, actual parallel efficiency varied from 67 to 92% (an average of 77.5%), depending on the occupation rate of the machine during the computation.

The discrepancy between the expected and actual parallel efficiencies was attributed to the structure of the SGI Origin 2000 cluster in which each node of the cluster has two PEs and one shared memory. When a CFD computation was submitted to a node with both PEs free, the process was done very fast because the process occupied its memory. On the other hand, when a CFD computation was submitted to a node with one PE free and one occupied by the other process, the computation became slow because it had to compete with the existing process for access to the memory. As a result, the computational time of each CFD computation varied significantly among different PEs, causing the parallel efficiency of the computation to degrade.

Nevertheless, the parallel efficiency was still very high for this parallel computation that used 64 PEs, and the parallel efficiency would be higher than 92% if a parallel computer with one memory for each PE were used. This result proves that the EA-based high-fidelity (Navier–Stokes) design optimization is extremely suitable for parallel computation. In the present study, the average computational time for each generation was $15\frac{1}{2}$ h on the SGI cluster, resulting in the total turnaround time reduced to about two months by the utilization of parallel computing.

Figure 7 presents an optimization history in terms of an objective function represented by entropy production, clearly showing a monotonic reduction. Thus, the design was continually improved with each additional generation. We note that the entropy production of the best design at the 33rd generation increased because the SGI cluster accidentally shutdown, and the EA misinterpreted the shutdown to mean that the flow computation of the best design had not converged. However, even with this glitch, the entropy production continued to decrease afterward, and the optimized design successfully cut its value by more than 19% compared with the NASA rotor 67 after 100 generations. This is a significant improvement for a compressor blade design. In addition, we anticipate that better designs may be obtained if the computation is continued further.

**Table 2 Computed performance of NASA rotor 67 and the optimized design**

| | Mass flow rate, kg/s | Isentropic efficiency | Pressure ratio | Entropy production |
|---|---|---|---|---|
| Rotor 67 | 33.774 | 0.91890 | 1.6758 | 0.0090467 |
| Optimum | 33.929 | 0.93528 | 1.6859 | 0.0073263 |

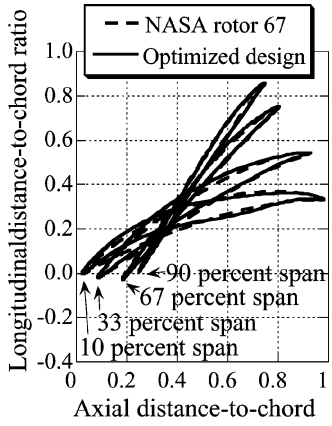


**Fig. 8 Blade profiles of the optimized design and NASA rotor 67.**

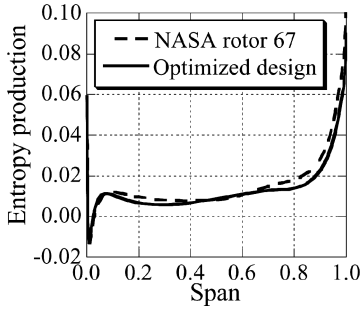


**Fig. 9 Comparison of spanwise entropy production distributions.**

Table 2 compares the performance of the optimized design with that of rotor 67. The constraints on the mass flow rate and pressure ratio were satisfied, and the isentropic efficiency was improved by 1.783% at the design condition, resulting in a higher pressure ratio across the rotor than with the baseline design. The blade profiles of the optimized design and of rotor 67 are shown in Fig. 8.

Figure 9 compares the spanwise distributions of entropy production for rotor 67 and the optimized design. Figure 9 shows that the optimized design reduced entropy production in the regions between the hub and the midspan and near the tip.

Figures 10 and 11 show the blade profiles and the corresponding surface static pressure distributions at spanwise stations of 33% (near the hub) and 90% (near the tip), respectively. The excessive flow acceleration near the leading edge at the 33% span station has diminished because of the decrease of the incidence angle. In addition, at the 90% span station, the shock on the suction side has moved aft and has been weakened considerably because of the aft movement of the maximum camber position. Note that the optimized design gives rise to a double-hump blade shape, especially obvious on the pressure side.

Figures 12 and 13 present the corresponding relative Mach number contours. The optimized design minimized the extent of the supersonic bubble on the suction side near the leading edge at the 33% span station. This explains the reduction in entropy production between the hub and the midspan. At the 90% span station, both the bow shock that impinges the blade suction side and the reflected shock have become more oblique, thus reducing entropy production

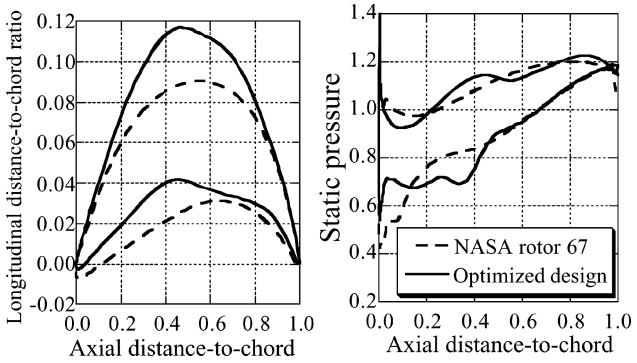


**Fig. 10 Comparison between the optimized design and rotor 67 at 33% span.**

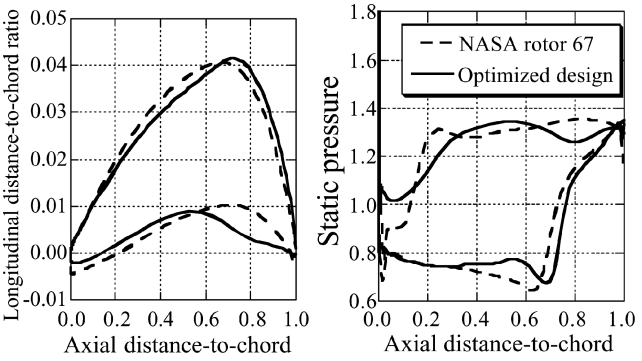


**Fig. 11 Comparison between the optimized design and rotor 67 at 90% span.**

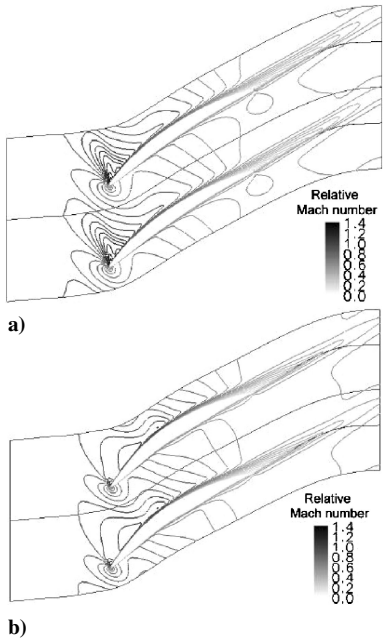


**Fig. 12 Relative Mach number contours of the optimized design and rotor 67 at 33% span: a) NASA rotor 67 and b) optimized design (maximum relative Mach numbers 1.39 for NASA rotor 67 and 1.19 for optimized design, respectively.)**

through the shocks significantly. Flow separation also has decreased because of the weakened shock, further reducing entropy generation.

Figure 14 shows oil flow patterns and static pressure contours on the suction surfaces of the NASA rotor 67 and the optimized design. The shock wave on the optimized design exists only partially on the blade between the stations at 19 and 100% span, whereas that on the NASA rotor 67 exists from the hub to the tip. Figure 14 also shows that the shock wave on the optimized design is significantly weakened. In addition, the shock wave on the suction side of the

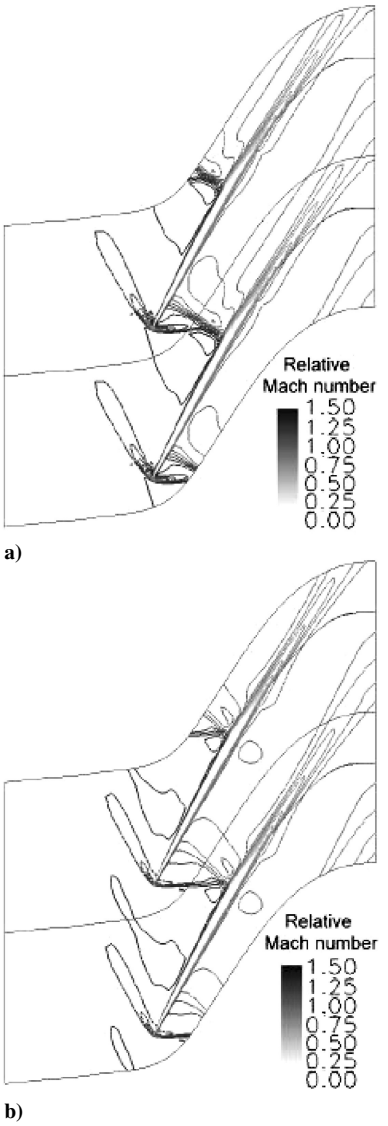


**Fig. 13 Relative Mach number contours of the optimized design and rotor 67 at 90% span: a) NASA rotor 67 and b) optimized design (maximum relative Mach numbers 1.47 for NASA rotor 67 and 1.42 for optimized design, respectively).**

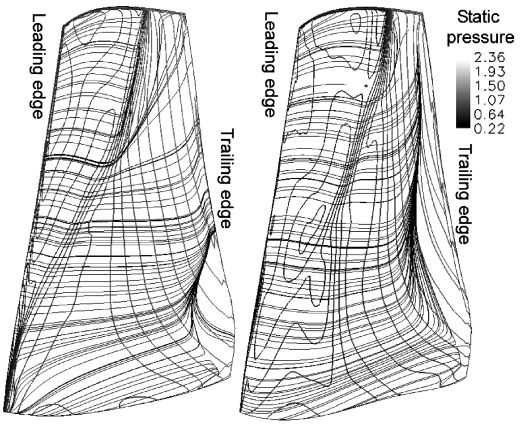


**Fig. 14 Oil flow patterns and static pressure contours on suction surfaces: a) NASA rotor 67 and b) optimized design.**

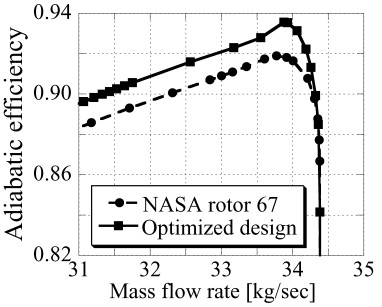


**Fig. 15 Performance map comparison between the optimized design and rotor 67.**

optimized design slants toward downstream from 60 to 90% span to reduce the shock-generated entropy.

Figure 15 compares the performance maps of the optimized design and the NASA rotor 67 over the entire operating range. Although the optimization was carried out only at the design operating condition, it is of considerable significance that the optimized design maintained higher isentropic efficiency over the entire operating range, from choke to stall limits, achieving a remarkable, better than 2% improvement in adiabatic efficiency.

## Conclusions

An EA-based high-fidelity aerodynamic design optimization tool for a transonic compressor blade has been successfully developed. To represent flowfields accurately and so that reliable designs would be produced, a three-dimensional Navier–Stokes solver was used for the aerodynamic analysis. The computation was parallelized and performed on the SGI cluster at the Institute of Fluid Science, Tohoku University in Japan to reduce the turnaround time for the study. For an efficient and robust design optimization, the real-coded ARGA was adopted.

The aerodynamic redesign of the NASA rotor 67 demonstrates the superiority of the present method over conventional through-flow analysis and the trial- and-error design approach commonly used in aeroengine designs. The present method finds a revolutionary compressor blade shape that improves the isentropic efficiency considerably (by 1.783%) while it maintains a total pressure ratio and mass flow of the original design.

This study proves the suitability of the EA-based high-fidelity compressor design optimization to parallel computing. No substantial modifications were made in the CFD code or in the EA code for parallelization because CFD computations of different design candidates were simply distributed onto different PEs. In addition, the parallel efficiency was found to be extremely high even though the number of PEs used for the present optimization is large. The efficiency reached as high as 92% when a simple master–slave concept was used and could have been improved further if the cache hit rate of the CFD solver had been increased.

The present design optimization also reveals three philosophies for compressor blade design:

1) The design of the leading-edge shape is very important in the minimization of the size of the supersonic bubble on the suction side near the hub.

2) The shock wave inside the passage should be as oblique as possible in both the meridional and tangential planes to reduce the shock-generated entropy.

3) An optimum design with significantly higher isentropic efficiency over the entire range of operating conditions can be maintained only if the design is optimized at the operating condition.

According to Moore's speech at the 2003 International Solid-State Circuits Conference, Moore's law will continue for at least another 10 years. Hence, although EA-based high-fidelity compressor design optimization is expensive today, we believe that it will be easily demonstrated on a personal computer cluster in 10 years. Application of the present method to multidisciplinary design optimization problems is also straightforward because EAs make tradeoffs to find optimum designs in parallel when applied to a multiobjective optimization problem. Finally, this study has shown that the present approach (EAs) offers a promising tool for turbomachinery designers to design better machines while the design cycle is shortened and design costs reduced in the near future.

## Acknowledgments

This work was performed while the first author held a National Research Council Research Associateship Award at the NASA Glenn Research Center. The generosity of the Institute of Fluid Science, Tohoku University for providing computational time on the SGI Origin 2000 cluster is greatly appreciated. The authors thank Louis M. Larosiliere and Rodrick V. Chima at the NASA Glenn Research Center for their valuable advice in defining the present design optimization problem and thank Eric R. McFarland for managing the project. The authors also thank Michele Marconcini at the University of Florence for his help in using the TRAF3D code.

## References

[1]Newman, J. C., III, Taylor, A. C., III, Barnwel, R. W., Newman, P. A., and Hou, G. J.-W., "Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations," *Journal of Aircraft*, Vol. 36, No. 2, 1999, pp. 87–96.

[2]Obayashi, S., and Tsukahara, T., "Comparison of Optimization Algorithms for Aerodynamic Shape Design," *AIAA Journal*, Vol. 35, No. 8, 1997, pp. 1413–1415.

[3]Periaux, J., Sefrioui, M., Stoufflet, B., Mantel, B., and Laporte, E., "Robust Genetic Algorithms for Optimization Problems in Aerodynamic Design," *Genetic Algorithms in Engineering and Computer Science*, Wiley, Chichester, England, U.K., 1995, pp. 371–395.

[4]Sasaki, D., Obayashi, S., and Kim, H.-J., "Evolutionary Algorithm vs. Adjoint Method Applied to SST Shape Optimization," *Proceedings of the Annual Conference of CFD Society of Canada*, Computational Fluid Dynamics Society of Canada, Waterloo, ON, Canada, 2001 [CD-ROM].

[5]Miettinen, K., Makela, M. M., Neittaanmaki, P., and Periaux, J. (eds.), *Evolutionary Algorithms in Engineering and Computer Science*, Willey, Chichester, England, U.K., 1999, Chaps. 17–24.

[6]Dasgupta, D., and Michalewicz, Z. (eds.), *Evolutionary Algorithms in Engineering Applications*, Springer-Verlag, Berlin, 1997, Chaps. 2–3.

[7]Benini, E., and Toffolo, A., "Development of High-Performance Airfoils for Axial Flow Compressors Using Evolutionary Computation," *Journal of Propulsion and Power*, Vol. 18, No. 3, 2002, pp. 544–554.

[8]Oksuz, O., Akmandor, I. S., and Kavsaoglu, M. S., "Aerodynamic Optimization of Turbomachinery Cascades Using Euler/Boundary-Layer Coupled Genetic Algorithms," *Journal of Propulsion and Power*, Vol. 18, No. 3, 2002, pp. 652–657.

[9]Pierret, S., "Three-Dimensional Blade Design by Means of an Artificial Neural Network and Navier–Stokes Solver," *Turbomachinery Blade Design Systems*, von Kármán Institute for Fluid Dynamics, Rhode Saint Genese, Belgium, 1999, Chap. 10.

[10]Bonaiuti, D., and Pediroda, V., "Aerodynamic Optimization of an Industrial Centrifugal Compressor Impeller Using Genetic Algorithms," *Evolutionary Methods for Design, Optimization and Control*, edited by K. C. Giannakoglou, D. T. Tsahalis, J. Periaux, and T. Fogarty, International Center for Numerical Methods in Engineering, Barcelona, Spain, 2001, pp. 467–472.

[11]Benini, E., "Design of Centrifugal Compressor Impellers for Maximum Pressure Ratio and Maximum Efficiency," *Proceedings of the 5th European Conference on Turbomachinery*, The European Organising Committee, Prague, Czech Republic, 2003, pp. 641–651.

[12]Oyama, A., and Liou, M.-S., "Multiobjective Optimization of a Multi-Stage Compressor Using Evolutionary Algorithm," AIAA Paper 2002-3535, July 2002.

[13]Oyama, A., and Liou, M.-S., "Multiobjective Optimization of Rocket Engine Pumps Using Evolutionary Algorithm," *Journal of Propulsion and Power*, Vol. 18, No. 3, 2002, pp. 528–535.

[14]Walter, S. C., William, S., and Donald, C. U., "Design and Performance of a 427-Meter-Per-Second-Tip-Speed Two-Stage Fan Having a 2.40 Pressure Ratio," NASA TP-1314, 1978.

[15]Arnone, A., Liou, M.-S., and Povinelli, L. A., "Navier–Stokes Solution of Transonic Cascade Flow Using Nonperiodic C-Type Grids," *Journal of Propulsion and Power*, Vol. 8, No. 2, 1992, pp. 410–417.

[16]Arnone, A., Liou, M.-S., and Povinelli, L. A., "Multigrid Calculation of Three-Dimensional Viscous Cascade Flows," NASA TM-105257, 1991.

[17]Arnone, A., "Viscous Analysis of Three-Dimensional Rotor Flow Using a Multigrid Method," *Journal of Turbomachinery*, Vol. 116, No. 3, 1994, pp. 435–445.

[18]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time-Stepping Schemes," AIAA Paper 81-1259, June 1981.

[19]Martinelli, L., and Jameson, A., "Validation of a Multigrid Method for the Reynolds Averaged Equations," AIAA Paper 88-0414, Jan. 1988.

[20]Swanson, R. C., and Turkel, E., "Artificial Dissipation and Central Difference Schemes for the Euler and Navier–Stokes Equations," AIAA Paper 87-1107, June 1987.

[21]Jameson, A., "The Evolution of Computational Methods in Aerodynamics," *Journal of Applied Mechanics*, Vol. 50, No. 4b, 1983, pp. 1052–1069.

[22]Jameson, A., "Transonic Flow Calculations," *Numerical Methods in Fluid Dynamics*, edited by F. Brezzi, Vol. 1127, Springer-Verlag, New York, June 1985, pp. 156–242.

[23]Pierzga, J. J., and Wood, J. R., "Investigation of the Three-Dimensional Flow Field Within a Transonic Fan Rotor: Experiment and Analysis," *Journal of Engineering for Gas Turbines and Power*, Vol. 107, No. 2, 1985, pp. 436–449.

[24]Oyama, A., Obayashi, S., and Nakahashi, K, "Real-Coded Adaptive Range Genetic Algorithm and Its Application to Aerodynamic Design," *JSME International Journal Series A*, Vol. 43, No. 2, 2000, pp. 124–129.

[25]Oyama, A., Obayashi, S., and Nakamura, T., "Real-Coded Adaptive Range Genetic Algorithm Applied to Transonic Wing Optimization," *Applied Soft Computing*, Vol. 1, No. 3, 2001, pp. 179–187.

[26]Sasaki, D., and Obayashi, S., "Low-Boom Design Optimization for SST Canard–Wing–Fuselage Configuration," AIAA Paper 2003-3432, June 2003.

[27]Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd rev. ed., Springer-Verlag, Berlin, Germany, 1996, pp. 56–62.

[28]Deb, K., Pratap, A., and Moitra, S., "Mechanical Component Design for Multiple Objectives Using Elitist Non-Dominated Sorting GA," *Lecture Notes in Computer Science 1917 Parallel Problem Solving from Nature–PPSN VI*, Springer, Berlin, 2000, pp. 859–868.

[29]Baker, J. E., "Reducing Bias and Inefficiency in the Selection Algorithm," *Proceedings of the Second International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, 1987, pp. 14–21.

[30]Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Longman, Reading, MA, 1989, pp. 185, 186.

[31]Eshelman, L. J., and Schaffer, J. D., "Real-Coded Genetic Algorithms and Interval Schemata," *Foundations of Genetic Algorithms 2*, Morgan Kaufmann, San Mateo, CA, 1993, pp. 187–202.

[32]De Jong, K. A., "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," Ph.D. Dissertation, Dept. of Computer and Communication Science, Univ. of Michigan, Ann Arbor, MI, 1975.