

Real-Coded Adaptive Range Genetic Algorithm Applied to Transonic Wing Optimization

Akira Oyama¹, Shigeru Obayashi² and Takashi Nakamura³

¹Tohoku University, Currently, NASA Glenn Research Center, Cleveland OH, USA.

²Tohoku University, Department of Aeronautics and Space Engineering,
Sendai, 980-8579, Japan.
obayashi@ieee.org

³National Aerospace Laboratory, Chofu, Tokyo, 182-8522, Japan.
nakamura@nal.go.jp

Abstract. Real-coded Adaptive Range Genetic Algorithms (ARGAs) have been applied to a practical three-dimensional shape optimization for aerodynamic design of an aircraft wing. The real-coded ARGAs possess both advantages of the binary-coded ARGAs and the floating-point representation to overcome the problems of having a large search space that requires continuous sampling. The results confirm that the real-coded ARGAs consistently find better solutions than the conventional real-coded Genetic Algorithms do.

1 Introduction

Most of commercial aircrafts today, such as B747, B777, and A340 cruise at transonic speeds, that is, just below the speed of sound. During the long duration of cruise, engine thrust is applied to maintain aircraft speed against aerodynamic drag. Since a large part of their maximum takeoff weights is occupied by the fuel weight, the objective of an aerodynamic design optimization of a transonic wing is, in principle, minimization of drag.

Unfortunately, drag minimization has many tradeoffs. There is a tradeoff between drag and lift because one of the drag component called induced drag increases in proportion to the square of the lift. A wing that achieves no induced drag would have no lift. Another tradeoff lies between aerodynamic drag and wing structure weight. An increase in the wing thickness allows the same bending moment to be carried with reduced skin thickness with an accompanying reduction in weight. On the other hand, it will lead to an increase in another component of the drag called wave drag. Therefore, the aerodynamic design of a transonic wing is a challenging problem.

Furthermore, optimization of a transonic wing design is difficult due to the followings. First, aerodynamic performance of a wing is very sensitive to its shape. Very precise definition of the shape is needed and thus its definition usually requires more than 100 design variables. Second, function evaluations are very expensive. An aerodynamic evaluation using a high fidelity model such as the Navier-Stokes equations usually requires 60-90 minutes of CPU time on a vector computer.

Among optimization algorithms, Gradient-based Methods (GMs) are well-known algorithms, which probe the optimum by calculating local gradient information. Although GMs are generally superior to other optimization algorithms in efficiency, the optimum obtained from these methods may not be a global one, especially in the aerodynamic optimization problem.

On the other hand, Genetic Algorithms (GAs) are known to be robust methods modeled on the mechanism of the natural evolution. GAs have capability of finding a global optimum because they don't use any derivative information and they search from multiple design points. Therefore, GAs are a promising approach to aerodynamic optimizations.

Finding a global optimum in the continuous domain for the aerodynamic design is challenging even for GAs. In traditional GAs, binary representation has been used for chromosomes, which evenly discretizes a real design space. Since binary substrings representing each parameter with a desired precision are concatenated to form a chromosome for GAs, the resulting chromosome encoding a large number of design variables for real-world problems would result in a string length too long. In addition, there is discrepancy between the binary representation space and the actual problem space. For example, two points close to each other in the real space might be far away in the binary-represented space. It is still an open question to construct an efficient crossover operator that suits to such a modified problem space.

A simple solution to these problems is the use of floating-point representation of parameters as a chromosome [1]. In these real-coded GAs, a chromosome is coded as a finite-length string of the 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 moreover, the string length reduces to the number of design variables. It has been reported that the real-coded GAs outperformed binary-coded GAs in many design problems [2]. However, even the real-coded GAs would lead to premature convergence when applied to aerodynamic shape designs with a large number of design variables.

The objective of the present work is to develop robust and efficient GAs applicable to aerodynamic shape designs. To achieve this goal, the idea of dynamic coding, in particular Adaptive Range GAs [3,4], is incorporated with the use of the floating-point representation. The resulting approach is then applied to a practical wing design problem as well as a simple test case to examine its performance.

To perform the practical wing design, the computation was processed in parallel using Numerical Wind Tunnel (NWT) at National Aerospace Laboratory, Japan. NWT has 166 vector processing elements at peak performance of 280 GFLOPS. The actual computation took 108 hours with 64 PE's.

2 Adaptive Range Genetic Algorithms

To treat a large search space with GAs more efficiently, sophisticated approaches have been proposed, referred to as dynamic coding, which dynamically alters the coarseness of the search space. In [5], Krishnakumar et al. presented Stochastic Genetic Algorithms (Stochastic GAs) to solve problems with a large number of real

design parameters efficiently. Stochastic GAs have been successfully applied to Flight Propulsion Controller designs [5] and air combat tactics optimization [6].

Adaptive Range Genetic Algorithms (ARGAs) proposed by Arakawa and Hagiwara [3] are a quite new approach, also using dynamic coding for binary-coded GAs to treat continuous design space. The essence of their idea is to adapt the population toward promising regions during the optimization process, which enables efficient and robust search in good precision while keeping the string length small. Moreover, ARGAs eliminate a need of prior definition of search boundaries since ARGAs distribute solution candidates according to the normal distributions of the design variables in the present population. In [4], ARGAs have been applied to pressure vessel designs and outperformed other optimization algorithms.

Since the ideas of the Stochastic GAs and the use of the floating point representation are incompatible, ARGAs for floating point representation are developed. The real-coded ARGAs are expected to possess both advantages of the binary-coded ARGAs and the floating point representation to overcome the problems of having a large search space that requires continuous sampling.

2.1 ARGAs for Binary Representation

When conventional binary-coded GAs are applied to real-number optimization problems, discrete values of real design variables p_i are given by evenly discretizing prior-defined search regions for each design variable $[p_{i,min}, p_{i,max}]$ according to the length of the binary substring $b_{i,l}$ as

$$p_i = (p_{i,max} - p_{i,min}) \frac{c_i}{2^{sl} - 1} + p_{i,min} \quad (1)$$

where sl represents string length and

$$c_i = \sum_{l=1}^{sl} (b_{i,l} \cdot 2^{l-1}).$$

In binary-coded ARGAs, decoding rules for the offspring are given by the following normal distributions,

$$N'(\mathbf{m}, \mathbf{s}_i^2)(p_i) = \sqrt{2\pi}\mathbf{s}_i \cdot N(\mathbf{m}, \mathbf{s}_i^2)(p_i) = \exp\left(-\frac{(p_i - \mathbf{m})^2}{2\mathbf{s}_i^2}\right) \quad (2)$$

where the average \mathbf{m} and the standard deviation \mathbf{s}_i of each design variable are determined by the population statistics. Those values are recomputed in every generation. Then, mapping from a binary string into a real number is given so that the region between N'_{UB} and N'_{LB} in Fig. 1 is divided into equal size regions according to the binary bit size as

$$p_i = \begin{cases} \mathbf{m} - \sqrt{-2\mathbf{s}_i^2 \cdot \ln(N'_{LB} + (N'_{UB} - N'_{LB}) \frac{c_i}{2^{sl-1} - 1})} & \text{for } c_i \leq 2^{sl-1} - 1 \\ \mathbf{m} + \sqrt{-2\mathbf{s}_i^2 \cdot \ln(N'_{UB} - (N'_{UB} - N'_{LB}) \frac{c_i - 2^{sl-1}}{2^{sl-1} - 1})} & \text{for } c_i \geq 2^{sl-1} \end{cases} \quad (3)$$

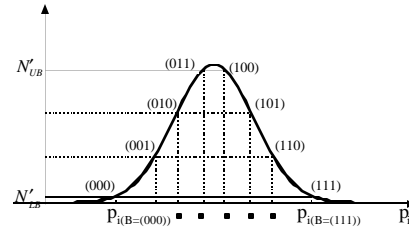


Fig. 1 Decoding for original ARGAs

where N'_{UB} and N'_{LB} are additional system parameters defined in $[0,1]$. In the ARGAs, genes of design candidates represent relative locations in the updated range of the design space. Therefore, the offspring are supposed to represent likely a range of an optimal value of design variables.

Although the original ARGAs have been successfully applied to real parameter optimizations, there is still room for improvements. The first one is how to select the system parameters N'_{UB} and N'_{LB} on which robustness and efficiency of ARGAs largely depend. The second one is the use of constant intervals even near the center of the normal distributions. The last one is that since genes represent relative locations, the offsprings become constantly away from the centers of the normal distributions when the distributions are updated. Therefore, the actual population statistics does not coincide with the updated population statistics.

2.2 ARGAs for Floating-Point Representation

In real-coded GAs, real values of design variable are directly encoded as a real string r_i , $p_i=r_i$ where $p_{i,\min} \leq r_i \leq p_{i,\max}$.

Otherwise, sometimes normalized values of the design variables are used as

$$p_i = (p_{i,\max} - p_{i,\min}) \cdot r_i + p_{i,\min} \quad (4)$$

where $0 \leq r_i \leq 1$.

To employ floating-point representation for ARGAs, the real values of design variables p_i are rewritten here by the real numbers r_i defined in $(0,1)$ so that integral of the probability distribution of the normal distribution from $-\infty$ to pn_i is equal to r_i as

$$p_i = \mathbf{s}_i \cdot pn_i + \mathbf{m} \quad (5)$$

$$r_i = \int_{-\infty}^{pn_i} N(0,1)(z) dz \quad (6)$$

where the average \mathbf{m} and the standard deviation \mathbf{s}_i of each design variable are calculated by sampling the top half of the previous population so that the present population distributes in the hopeful search regions. Schematic view of this coding is illustrated in Fig. 2. It should be noted that the real-coded ARGAs resolve drawbacks of the original ARGAs; no need for selecting N'_{UB} and N'_{LB} as well as arbitrary resolution near the average. Updating \mathbf{m} and \mathbf{s}_i every generation, however,

results in 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

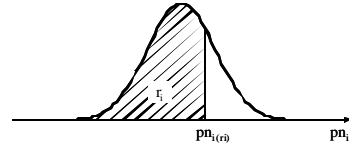


Fig. 2 Decoding for real-coded ARGAs

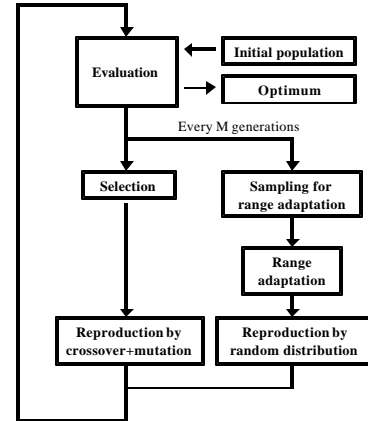


Fig. 3 Flowchart of ARGA

inconsistency, the present ARGAs update \mathbf{m} and \mathbf{s} every M ($M > 1$) generations and then the population is reinitialized. Flowchart of the present ARGAs is shown in Fig. 3.

To improve robustness of the present ARGAs further, relaxation factors, \mathbf{w}_m and \mathbf{w}_s are introduced to update the average and standard deviation as

$$\mathbf{m}_{new} = \mathbf{m}_{present} + \mathbf{w}_m (\mathbf{m}_{sampling} - \mathbf{m}_{present}) \quad (7)$$

$$\mathbf{s}_{new} = \mathbf{s}_{present} + \mathbf{w}_s (\mathbf{s}_{sampling} - \mathbf{s}_{present}) \quad (8)$$

where $\mathbf{m}_{sampling}$ and $\mathbf{s}_{sampling}$ are determined by sampling the top half of the population. Here, \mathbf{w}_m , \mathbf{w}_s and M are set to 1, 0.5 and 4, respectively. They are determined by parametric studies using some simple test functions.

In this study, design variables are encoded in a finite-length string of real numbers. Fitness of a design candidate is determined by its rank among the population based on its objective function value and then selection is performed by the stochastic universal sampling [7] coupled with the elitist strategy. Ranking selection is adopted since it maintains sufficient selection pressure throughout the optimization. One-point crossover is always applied to real-number strings of the selected design candidates. Structured coding [8] is incorporated for the wing design. Mutation takes place at a probability of 0.1, and then a uniform random disturbance is added to the corresponding gene in the amount up to 0.1.

2.3 Test Problem Using a Multi-Modal Function

To demonstrate how the real-coded ARGAs works, it was applied to minimization of a high dimensional multi-modal function:

$$F1 = \sum_{i=1}^{20} (x_i^2 + 5(1 - \cos(x_i \cdot \mathbf{p}))) \quad (9)$$

where $x_i \in [-3, 3]$. This function has a global minimum at $x_i = 0$ and two local optima near $x_i = \pm 2$. In the real-coded ARGAs, x_i correspond to p_i in eq.(5). 150 generations were allowed with a population size of 300. Five trials were run for each GA changing seeds for random numbers to give different initial populations. Figure 4

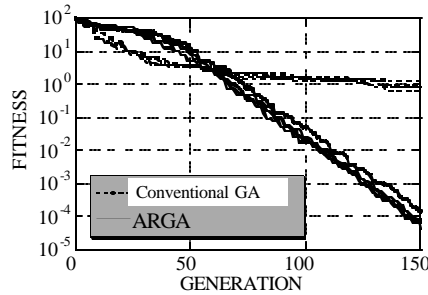


Fig. 4 Comparison of convergence histories

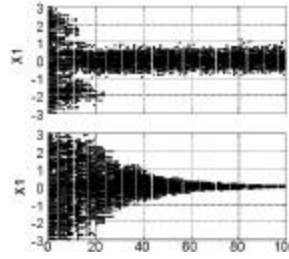


Fig. 5 Comparison of convergence histories of x_j between GA (above) and ARGAs (below)

compares the performances of the conventional GA and the ARGA. Figure 5 plots all x_j 's from the temporary solutions, which helps to understand why the ARGA works better than the conventional GA. This figure shows that the ARGA maintains gene diversity longer than the conventional GA in the initial phase and then adapts to their search space to the local region near the optimal. While the initial gene diversity contributes to the ARGA's robustness, the adaptive feature of the ARGA improves their local search capability. The ARGA also showed its advantages over a real-coded GA on dynamic control problem and aerodynamic airfoil shape optimization [9].

3 Aerodynamic Design of a Transonic Wing

A wide range of approximations can represent the flow physics. Among them, the Navier-Stokes equations provide the state-of-the-art of aerodynamic performance evaluation for engineering purposes. Although the three-dimensional Navier-Stokes calculation requires large computer resources to estimate wing performances within a reasonable time, it is necessary because a flow around a wing involves significant viscous effects, such as potential boundary-layer separations and shock wave/boundary layer interactions in the transonic regime. Here, a three-dimensional Reynolds-averaged Navier-Stokes solver [10] is used to guarantee an accurate model of the flow field and to demonstrate the feasibility of the present algorithm.

The objective of the present wing design problem is maximization of lift-to-drag ratio L/D at the transonic cruise design point, maintaining the minimum wing thickness required for structural integrity against the bending moment due to the lift distribution. The cruising Mach number is set to 0.8. The Reynolds number based on the chord length at the wing root is assumed to 10^7 .

In the present optimization, a planform shape of generic transport was selected as the test configuration (Fig. 6). Wing profiles of design candidates are generated by the PARSEC airfoils as briefly described in the next section. The PARSEC parameters and the sectional angle of attack (in other words, root incident angle and twist angle) are given at seven spanwise sections, of which spanwise locations are also treated as design variables except for the wing root and tip locations. The PARSEC parameters are rearranged from root to tip according to the airfoil thickness so that the resulting wings always have maximum thickness at the wing root. The twist angle parameter is also rearranged into numerical order from tip to root. The wing surface is then interpolated in spanwise direction by using the second-order Spline interpolation.

In total, 87 parameters determine a wing geometry. Parameter ranges of the design space are shown in Table 1. It should be noted that in ARGAs, user-defined design space is used just to seed the initial population. ARGAs can promote the search space outside of the initially defined design space.

To estimate the required thickness distribution to stand the bending moment due to the lift distribution, the wing is modeled by a thin walled box-beam as shown in Fig. 6. The constraint for wing thickness t_j is specified by using the minimum thickness t_{min} calculated from the wing box sustaining the aerodynamic bending moment M as,

$$t_j > \frac{M}{S_{ultimate} \cdot c \cdot t_2} = t_{min} \quad (10)$$

where following assumptions are made: the thickness of the skin panels are 2.5[cm] and its ultimate normal stress $s_{ultimate}$ is 39[ksi]. The length of the chord at wing root c and maximum wingspan $b/2$ are 10[m] and 18.8[m], respectively (for the derivation of Eq. (10), see [11] for example).

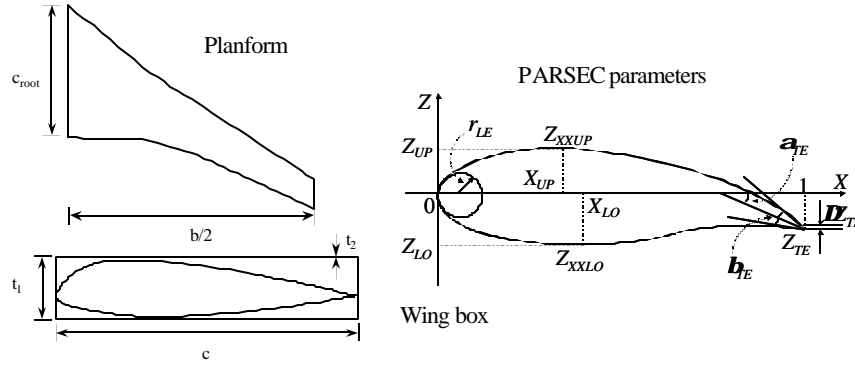


Fig. 6 Wing geometry definition. Planform shape is frozen during the optimization. Wing box is used to estimate its structural strength. PARSEC parameters are the design variables for airfoil shapes defined at seven spanwise sections

Table 1 Parameter ranges of the design space. PARSEC is determined by leading edge radius (r_{LE}), upper and lower crest locations including curvatures (X_{UP} , Z_{UP} , Z_{XXUP} , X_{LO} , Z_{LO} , Z_{XXLO}), trailing-edge ordinate (Z_{TE}) and thickness (ΔZ_{TE}) and direction and wedge angles (α_{TE} , β_{TE})

parameters	r_{LE}	Z_{TE}	α_{TE}	β_{TE}	X_{UP}	Z_{UP}	Z_{XXUP}	X_{LO}	Z_{LO}	Z_{XXLO}	twist angle
Upper bound	0.030	0.01	-3.0	8.0	0.7	0.18	0.0	0.6	0.02	0.9	7 deg
Lower bound	0.002	-0.01	-13.0	4.0	0.3	0.08	-0.3	0.2	-0.04	0.3	-1 deg

3.1 PARSEC Airfoils

An airfoil family ‘‘PARSEC’’ has been recently proposed to parameterize an airfoil shape [12]. A remarkable point is that this technique has been developed aiming to control important aerodynamic features effectively by selecting the design parameters based on the knowledge of transonic flows around an airfoil.

Similar to 4 digit NACA series airfoils, the PARSEC parameterizes upper and lower airfoil surfaces using polynomials in coordinates X , Z as,

$$Z = \sum_{n=1}^6 a_n \cdot X^{n-1/2} \quad (11)$$

where a_n are real coefficients. Instead of taking these coefficients as design parameters, the PARSEC airfoils are defined by basic geometric parameters:

leading-edge radius (r_{LE}), upper and lower crest locations including curvatures (X_{UP} , Z_{UP} , Z_{XXUP} , X_{LO} , Z_{LO} , Z_{XXLO}), trailing-edge ordinate (Z_{TE}), thickness (ΔZ_{TE}) and direction and wedge angles (α_{TE} , β_{TE}) as shown in Fig. 6. These parameters can be expressed by the original coefficients a_n by solving simple simultaneous equations. Eleven design parameters are required for the PARSEC airfoils to define an airfoil shape in total. In the present case, the trailing-edge thickness is frozen to 0. Therefore, ten design variables are used to give each spanwise section of the wing.

3.2 Optimization Using Real-Coded ARGA

Because the objective function distribution of the present optimization is likely to be more complex than the above test function minimization, the relaxation factor w_s is now set to 0.3. The structured coding coupled with one-point crossover proposed in [13] is also incorporated. The present ARGA adopts the elitist strategy where the best and the second best individuals in each generation are transferred into the next generation without any crossover or mutation. The parental selection consists of the stochastic universal sampling and the ranking method using Michalewicz's nonlinear function. Mutation takes place at a probability of 10% and then adds a random disturbance to the corresponding gene in the amount up to $\pm 10\%$ of each parameter range in Table 1. The population size is kept at 64 and the maximum number of generations is set to 65 (based on the CPU time allowed). The initial population is generated randomly over the entire design space.

The main concern related to the use of GAs coupled with a three-dimensional Navier-Stokes solver for aerodynamic designs is the computational cost required. In the present case, each CFD evaluation takes about 100 min. of CPU time even on a vector computer. Because the present optimization evaluates $64 \times 65 = 4160$ design candidates, sequential evolutions would take almost 7000 h (more than nine months!).

Fortunately, parallel vector computers are now available at several institutions and universities. In addition, GAs are intrinsically parallel algorithms and can be easily parallelized. One of such computers is *Numerical Wind Tunnel (NWT)* located at National Aerospace Laboratory in Japan. NWT is a MIMD parallel computer with 166 vector-processing elements (PEs) and its total peak performance and the total main memory capacity are about 280 GFLOPS and 45GB, respectively. For more detail, see [14]. In the present optimization, evaluation process at each generation was parallelized using the master-slave concept. This made the corresponding turnaround time almost $1/64$ because the CPU time used for GA operators are negligible.

To handle the structural constraint with the single-objective GA, the constrained optimization problem was transformed into an unconstrained problem as

$$fitness\ function = \begin{cases} 100 + L/D & \text{if } t \geq t_{min} \\ (100 + L/D) \cdot \exp(t - t_{min}) & \text{otherwise} \end{cases} \quad (12)$$

where t and t_{min} are thickness and minimum thickness at the span station of the maximum local stress. The exponential term penalizes the infeasible solutions by reducing the fitness function value. Because some design candidates can have negative L/D , the summation of 100 and L/D is used.

3.3 Results

The optimization history is shown in Fig. 7 in terms of L/D . During the initial phase of the optimization, some members had a strong shock wave or failed to satisfy the structural constraint. However, they were weeded out from the population because of the resultant penalties to the fitness. The final design has L/D of 18.91 ($C_L = 0.26213$ and $C_D = 0.01386$) satisfying the given structural constraint. Turnaround time of this optimization was about 108 h on NWT.

To examine whether the present optimal design is close to a global optimum, we have checked it against analytically and empirically established design guidelines. In aerodynamics, spanwise lift distribution should be elliptic to minimize the induced drag. However, the structural constraint leads to a tradeoff between induced drag and wave drag. This enforces the spanwise lift distribution to be linear rather than elliptic.

The present solution has a linear distribution. To produce this distribution, a wing is usually twisted in about five degrees. The present wing is twisted in six degrees.

Figure 8 shows the designed airfoil sections and the corresponding pressure distributions at the 0, 33, and 66% spanwise locations. In the pressure distributions, neither any strong shock wave nor any flow separation is found. This ensures that the present wing has very little wave drag and pressure drag. At 33 and 66% spanwise locations, the rooftop, front-loading and rear loading patterns are observed, which are typical for the supercritical airfoils [15] used for advanced transport today. The corresponding airfoil shapes are indeed similar to supercritical airfoils. Overall, these detailed observations of the design confirm that the present design is very close to a global optimum expected by the present knowledge in aerodynamics.

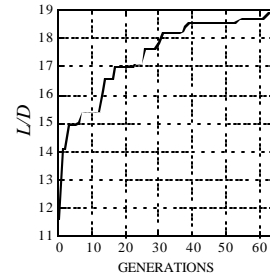


Fig. 7 Optimization history

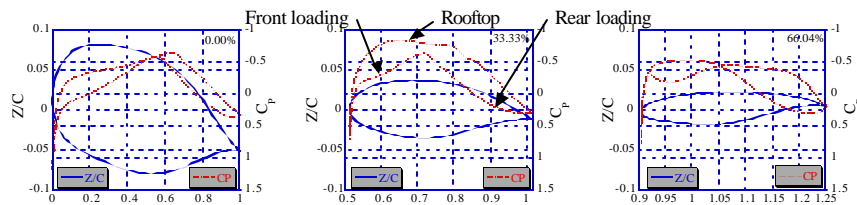


Fig. 8 Designed airfoil sections and corresponding pressure distributions

4 Summary

To develop GAs applicable to practical aerodynamic shape designs, the real-coded ARGAs have been developed by incorporating the idea of the binary-coded ARGAs with the use of the floating-point representation. The real-coded ARGAs have been applied to a practical aerodynamic design optimization of a transonic wing shape for generic transport as well as a simple test case. The test case result confirms the

present GA outperforms the conventional GA.

Aerodynamic optimization was performed with 87 real-number design variables by using the Navier-Stokes code. The realistic structural constraint was imposed. The resulting wing appears very similar to advanced wing designs based on supercritical airfoils. The straight span load distribution of the resulting design represents a compromise between minimizations of induced drag and wave drag. The designed wing also has a fully attached flow and the allowable minimum thickness so that pressure drag and wave drag are minimized under the present structural constraint. These results confirm the feasibility of the present approach for future applications.

5 References

1. Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, third revised edition, Springer-Verlag, (1996).
2. Janikow, C. Z. and Michalewicz, Z., An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithms, Proc. of the 4th Intl. Conference on Genetic Algorithms, (1991), pp.31-36.
3. Arakawa, M. and Hagiwara, I., Development of Adaptive Real Range (ARRange) Genetic Algorithms, *JSME Intl. J.*, Series C, Vol. 41, No. 4 (1998), pp. 969-977.
4. Arakawa, M. and Hagiwara, I., Nonlinear Integer, Discrete and Continuous Optimization Using Adaptive Range Genetic Algorithms, Proc. of 1997 ASME Design Engineering Technical Conferences, (1997).
5. Krishnakumar, K., Swaminathan, R., Garg, S. and Narayanaswamy, S., Solving Large Parameter Optimization Problems Using Genetic Algorithms, Proc. of the Guidance, Navigation, and Control Conference, (1995), pp.449-460.
6. Mulgund, S., Harper, K., Krishnakumar, K. and Zacharias, G., Air Combat Tactics Optimization Using Stochastic Genetic Algorithms, Proc. of 1998 IEEE Intl. Conference on Systems, Man, and Cybernetics, (1998), pp.3136-3141.
7. Baker, J. E., Reducing Bias and Inefficiency in the Selection Algorithm, Proc. of the 2nd Intl. Conference on Genetic Algorithms, (1987), pp.14-21.
8. Oyama, A., Obayashi, S., Nakahashi, K. and Hirose, N., Aerodynamic Wing Optimization via Evolutionary Algorithms Based on Structured coding, *CFD Journal*, Vol. 8, No. 4 (2000), pp. 570-577.
9. Oyama, A., Obayashi, S. and Nakahashi, K., Wing Design Using Real-Coded Adaptive Range Genetic Algorithm, Proc. of 1999 IEEE Intl. Conference on Systems, Man, and Cybernetics [CD-ROM], (1999).
10. Obayashi, S. and Guruswamy, G. P., Convergence Acceleration of an Aeroelastic Navier-Stokes Solver, *AIAA Journal*, Vol. 33, No. 6, 1995, pp.1134-1141.
11. Case, J., Chilver, A. H. and Ross, C. T. F., *Strength of Materials & Structures with an Introduction to Finite Element Methods*, 3rd Edn., Edward Arnold, London, 1993.
12. Sobieczky, H., Parametric Airfoils and Wings, *Recent Development of Aerodynamic Design Methodologies –Inverse Design and Optimization –*, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, (1999), pp.72-74.
13. Oyama, A., Obayashi, S., Nakahashi, K. and Hirose, N., Fractional Factorial Design of Genetic Coding for Aerodynamic Optimization, AIAA Paper 99-3298, (1999).
14. Nakamura, T., Iwamiya, T., Yoshida, M., Matsuo, Y. and Fukuda, M., Simulation of the 3 Dimensional Cascade Flow with Numerical Wind Tunnel (NWT), Proc. of the 1996 ACM/IEEE Supercomputing Conference [CD-ROM], (1996).
15. Harris, C. D., NASA Supercritical Airfoils, NASA TP 2969, (1990).