Real-Coded Adaptive Range Genetic Algorithm And Its Application to Aerodynamic Design

Akira Oyama*, Shigeru Obayashi* and Kazuhiro Nakahashi*

* Tohoku University, Department of Aeronautics and Space Engineering, Sendai, 980-8579, Japan
{oyama, obayashi, nakahash}@ad.mech.tohoku.ac.jp

ABSTRACT

Real-coded Adaptive Range Genetic Algorithms (ARGAs) have been developed. The real-coded ARGAs possess both advantages of the binary-coded ARGAs and the use of the floating point representation to overcome the problems of having a large search space that requires continuous sampling. First, the efficiency and the robustness of the proposed approach are demonstrated by test functions. Then the proposed approach is applied to an aerodynamic airfoil shape optimization problem. The results confirm that the real-coded ARGAs consistently find better solutions than the conventional real-coded Genetic Algorithms do. The designed airfoil shape is considered to be the global optimal and thus ensures the feasibility of the real-coded ARGAs in aerodynamic designs.

1. INTRODUCTION

Aerodynamic optimization of a wing using Computational Fluid Dynamics (CFD) is a challenging problem due to the following three reasons. First, the objective function distributions are extremely rough as pointed out in [1], originating in nonlinearity of the governing flow equations. Second, the design space is highly multidimensional. For example, a wing shape is usually parameterized by over 100 design variables since a wing consists of an elaborately curved surface. Third, function evaluations are very expensive. An aerodynamic evaluation using a Navier-Stokes calculation, for instance, usually requires 60-90 minutes of CPU time on a vector computer.

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

On the other hand, Genetic Algorithms (GAs) are known to be robust optimization algorithms 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. However, there are limitations in a population size as well as a number of generations allowed.

Finding a global optimum in the continuous domain is challenging for GAs. In traditional GAs, binary representation has been used for chromosomes, which evenly discretizes a real design space. Although such binary-coded GAs have been successfully applied to a wide range of design optimization problems (for example, see [2],[3]), they suffer from disadvantages, when applied to the real-world problems involving a large number of real design variables. Since binary substrings representing each parameter with the desired precision are concatenated to form a chromosome for the GAs, the resulting chromosome encoding a large number of design variables would result in a huge string length. For example, for 100 variables with a precision of six digits, the string length is about 2000. GAs would perform poorly for such design problems. Previous applications have been kept away from this problem by sacrificing precision or narrowing down the search regions prior to the optimization.

However, such approaches might exclude the region that actually has the global optimum.

Another drawback of the binary-coded GAs applied to parameter optimization problems in continuous domains comes from discrepancy between the binary representation space and the actual problem space. For example, two points close to each other in the representation space might be far in the binary represented problem 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 [4]. In these real-coded GAs, a chromosome is coded as a finite-length string of the real numbers corresponding to the design variables. The real-coded GAs are robust, accurate, and efficient because the floating point representation 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 [5], [6]. However, even the real-coded GAs would lead to premature convergence when applied to aerodynamic shape designs with a large number of design variables.

A more sophisticated approach is to alter dynamically the coarseness of the search space referred to as dynamic coding. In [7], Krishnakumar et al. presented Stochastic Genetic Algorithms (Stochastic GAs) to solve efficiently problems with a large number of real design parameters. The key features of Stochastic GAs are:

1) Each binary number represents a region of the real space instead of a single point to maintaining good precision with the small string length.
2) Those regions adapt during the optimization process according to the 1/5th success rule as Evolutionary Strategies (ES) to improve efficiency and robustness.

Actually, the Stochastic GAs have been successfully applied to Integrated Flight Propulsion Controller designs [7] and air combat tactics optimization [8]. As they mentioned, the Stochastic GAs bridge the gap between ES and GAs to handle large design problems.

Adaptive Range Genetic Algorithms (ARGAs) are a quite new approach using dynamic coding proposed by Arakawa and Hagiwara [9] for binary-coded GAs to treat continuous design space. The essence of their idea is to adapt the population toward promising design regions during the optimization process, which enables efficient and robust search in good precision while keeping the string length small. Moreover, ARGAs eliminate prior definition of boundaries of the search regions since ARGAs distribute design candidates according to the normal distributions of the design variables in the present population. In [10], ARGAs have been applied to pressure vessel designs and outperformed other optimization algorithms.

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 the dynamic coding is incorporated with the used of the floating point representation. 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. First, to display advantages of the present approach, the proposed approach will be applied to two test function optimization problems. Then, an aerodynamic airfoil shape optimization will be demonstrated to ensure the feasibility of the proposed approach in aerodynamic design problems.

2. ADAPTIVE RANGE GENETIC ALGORITHMS

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

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

where

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

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

$$\begin{aligned} N'(m_i, s_i^{\,2})(p_i) &= \sqrt{2p}\, s_i \cdot N(m_i, s_i^{\,2})(p_i) \\ &= \exp\left(-\frac{(p_i - m_i)^2}{2 s_i^{\,2}}\right) \qquad (3) \end{aligned}$$

where the average m_i and the standard deviation 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

$$p_i = \begin{cases} m_i - \sqrt{-2s_i^{\,2} \cdot \ln\left(N'_{LB} + (N'_{UB} - N'_{LB})\dfrac{c_i}{2^{sl-1}-1}\right)} & for \quad c_i \le 2^{sl-1} - 1 \\ m_i + \sqrt{-2s_i^{\,2} \cdot \ln\left(N'_{UB} - (N'_{UB} - N'_{LB})\dfrac{c_i - 2^{sl-1}}{2^{sl-1}-1}\right)} & for \quad c_i \ge 2^{sl-1} \end{cases} \qquad (4)$$

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 improvement. 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 coded as a real string r_i ,

$$p_i = r_i \qquad (5)$$

where $p_{i,\min} \le r_i \le p_{i,\max}$

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

$$p_i = (p_{i,\max} - p_{i,\min}) \cdot r_i + p_{i,\min} \qquad (6)$$

where $0 \le r_i \le 1$

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

$$p_i = s_i \cdot pn_i + m_i \qquad (7)$$

$$r_i = \int_{-\infty}^{pn_i} N(0,1)(z)dz \qquad (8)$$

where the average m_i and the standard deviation s_i of each design variable are calculated by the top half of the present population. 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. To prevent inconsistency between the actual and updated population statistics, the present ARGAs update μ and σ every M generations and then the population is reinitialized. Flowchart of the resulting ARGA is shown in Fig. 3.

To improve robustness of the present ARGAs further, relaxation factors w_m and w_s are introduced to update the average and standard deviation as

$$m_{new} = m_{present} + w_m (m_{sampling} - m_{present}) \qquad (9)$$

$$s_{new} = s_{present} + w_s (s_{sampling} - s_{present}) \qquad (10)$$

where w lower than 1 contributes to improve robustness of the ARGAs. $m_{sampling}$ and $s_{sampling}$ are determined by sampling the top half of the present population. Here, w_m , w_s and M are set to 1, 0.5 and 4, respectively. They are determined by a parametric study using a simple test function.

In this study, design variables are coded 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 [11] 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. 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.

3. RESULTS

3.1 Multi-Modal function

To demonstrate how the real-coded ARGAs work, they were first applied to minimization of a high dimensional multi-modal function:

$$F1 = \sum_{i=1}^{20} (x_i^{\,2} + 5(1 - \cos(x_i \cdot p)) \qquad (11)$$

where $x_i \in [-3,3]$. Figure 4 shows one-dimensional version of this function. This function has two local optima near $x_i = \pm 2$. Here, 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 5 compares the performances of the conventional GA and the ARGA. Figure 6 plotting $\boldsymbol{x}_{1sol}$'s which are obtained as the temporary solutions of all design candidates helps to understand why the ARGA worked much better than the conventional GA. This figure shows that the ARGA maintained gene diversity longer than the conventional GA in the initial phase and then adapt to their search space to the local region near the optimal. While the initial gene diversity contributes to the ARGAs' robustness, the adaptive feature of the ARGAs improves their local search capability.

3.2 Dynamic control problem

The next problem is a dynamic control problem in [5]. This problem is hard to optimize since it involves many correlated design parameters like practical design problems. The objective is to minimize the following function:

$$F2 = x_N{}^2 + \sum_{k=0}^{N-1} (x_k{}^2 + u_k{}^2) \qquad (12)$$

subject to: $x_{k+1} = x_k + u_k$, $\quad k = 0,1,..., N-1$,

where, x_k is a state, and $\vec{\boldsymbol{u}} = (\boldsymbol{u}_0,...,\boldsymbol{u}_{N-1})$ is the control vector. The search domain is [-200,200] for each u_k. The initial state x_0 and the size of the control vector are given by 100 and 45, respectively. The population size was set to 50, and the maximum number of generations is set to 500. Figure 7 compares the optimization histories of five trials for both GAs. The bold line indicates the analytically obtained optimum value. While all the trials using the conventional GA lead to premature convergence, the ARGA find the global optimum at every trial.

3.3 Aerodynamic airfoil shape optimization

To demonstrate performance of the real-coded ARGAs in comparison with that of the conventional real-coded GAs, aerodynamic airfoil shape optimizations were carried out. The objective function was the lift-to-drag ratio to be maximized where the free stream Mach number and the angle of attack were set to 0.8 and 2 degrees, respectively. The airfoil thickness was constrained so that the maximum thickness was greater than 12% of the chord length. The aerodynamic performance of each design was evaluated by the two-dimensional Navier-Stokes solver based on a total variation diminishing type upwind differencing [12], the lower-upper symmetric Gauss-Seidel scheme [13] and the multigrid method [14].

As described in [15], aerodynamic performances of the designed airfoils greatly depend on the choice of the shape parameterization techniques. Therefore, the PARSEC airfoil [16] was used, which can represent a wide variety of airfoils with a reasonable number of parameters. This technique parameterizes an airfoil shape using a linear combination of shape functions as

$$Z = \sum_{n=1}^{6} a_n \cdot X^{n-1/2} \qquad (13)$$

An airfoil shape is defined by basic geometric parameters that are directly related to the knowledge of transonic flows around the airfoil, instead of the coefficients of shape functions themselves: leading-edge radius, upper and lower crest location including curvatures, trailing-edge ordinate, thickness, direction and wedge angle as illustrated in Fig. 8. Those parameters are related to the coefficients of the shape functions by solving simple simultaneous equations. Selecting those parameters is considered to help finding the global optimum design by relaxing the complexity of the objective function distribution. In this study, nine design variables are used to give an airfoil shape with both trailing-edge thickness and its ordinate (at X=1) frozen to 0.

Figure 9 compares optimization histories of three trials using the real-coded ARGA and the conventional real-coded GA. Both of the population size and the number of generations were 100. The present ARGA outperformed the conventional GA starting from all initial populations.

Figure 10 shows the shape and the corresponding pressure coefficient distribution of the airfoil designed with the ARGA. The surface pressure distribution is similar to that of NASA supercritical airfoils, such as an approximately uniform distribution (rooftop) on the upper surface, a weak shock wave significantly rear of the midchord, a pressure plateau downstream of the shock wave, a relatively steep pressure recovery on the extreme rearward region, and a trailing edge pressure slightly more positive than ambient pressure [17]. The design result is considered to be the global optimal and thus ensures the feasibility of the present ARGA in aerodynamic designs.

4. CONCLUTIONS

To develop robust and efficient GAs applicable to 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 resulting real-coded ARGAs are expected to possess both advantages of the binary-coded ARGAs and the use of the floating point representation to overcome the problems of having a large search space that requires continuous sampling. First, the efficiency and the robustness of the proposed approach have been demonstrated by using two test functions. Then the proposed approach has been applied to an aerodynamic airfoil shape optimization problem. The results confirm that the real-coded ARGAs consistently find better solutions than the conventional real-coded GAs do. The design result is considered to be the global optimal and thus ensures the feasibility of the real-coded ARGAs in aerodynamic designs.

REFERENCES

[1] Obayashi, S. and Tsukahara, T., Comparison of Optimization Algorithms for Aerodynamic Shape Design, AIAA Paper 96-2394, (1996).

[2] Quagliarella, D., Periauz, J., Poloni, C. and Winter, G. (Eds.), Genetic Algorithms in Engineering and Computer Science, John Wiley and Sons, (1997).

[3] Dasgupta, D. and Michalewicz, Z. (Eds.), Evolutionary Algorithms in Engineering Applications, Springer, (1997).

[4] Michalewicz, Z., Genetic Algorithms + Data Structures = Evolution Programs, 3rd ed., Springer-Verlag, (1996).

[5] 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.

[6] Wright, A. H., Genetic Algorithms for Real Parameter Optimization, in Rawlins, G. (Ed.), Foundations of Genetic Algorithms, Morgan Kaufmann Publishers, (1991), pp.205-218.

[7] 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.

[8] 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.

[9] 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.

[10] 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).

[11] Baker, J. E., “Reducing Bias and Inefficiency in the Selection Algorithm,” Proc. of the 2nd Intl. Conference on Genetic Algorithms, (1987), pp.14-21.

[12] Obayashi, S. and Wada, Y., Practical Formulation of a Positively Conservative Scheme, AIAA J., Vol. 32, No.5(1994), pp. 1093-1095.

[13] Yoon, S., Jameson, A. and Kwak, D., Effect of Artificial Diffusion Scheme on Multigrid Convergence, AIAA Paper 95-1670, (1995).

[14] Jameson, A., Solution of the Euler Equations for Two-Dimensional Transonic Flow by a Multigrid Method, Applied Mathematics and Computation, Vol. 13, Nov. (1983), pp. 327-356.

[15] Oyama, A., Obayashi, S., Nakahashi, K. and Hirose, N., Fractional Factorial Design of Genetic Coding for Aerodynamic Optimization, AIAA Paper 99-3298, (1999).

[16] Li, P., Seebass, A. R., and Sobieczky, H., Manual Aerodynamic Optimization of an Oblique Flying Wing, AIAA Paper 98-0598, (1998).

[17] Harris, C. D., NASA Supercritical Airfoils, NASA TP 2969, (1990).

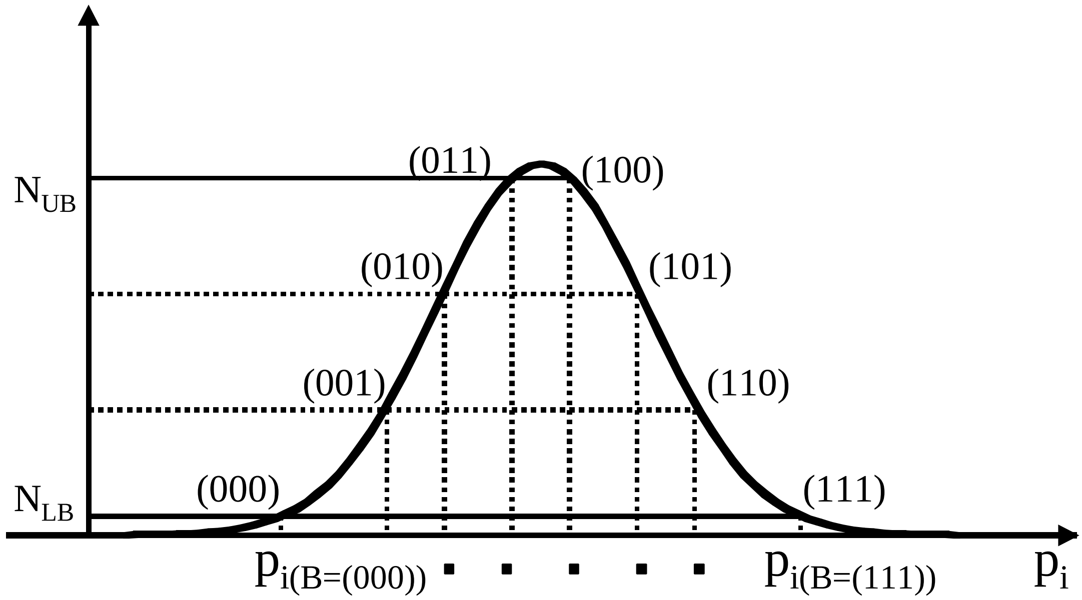


Figure 1. Decoding for binary-coded ARGAs.

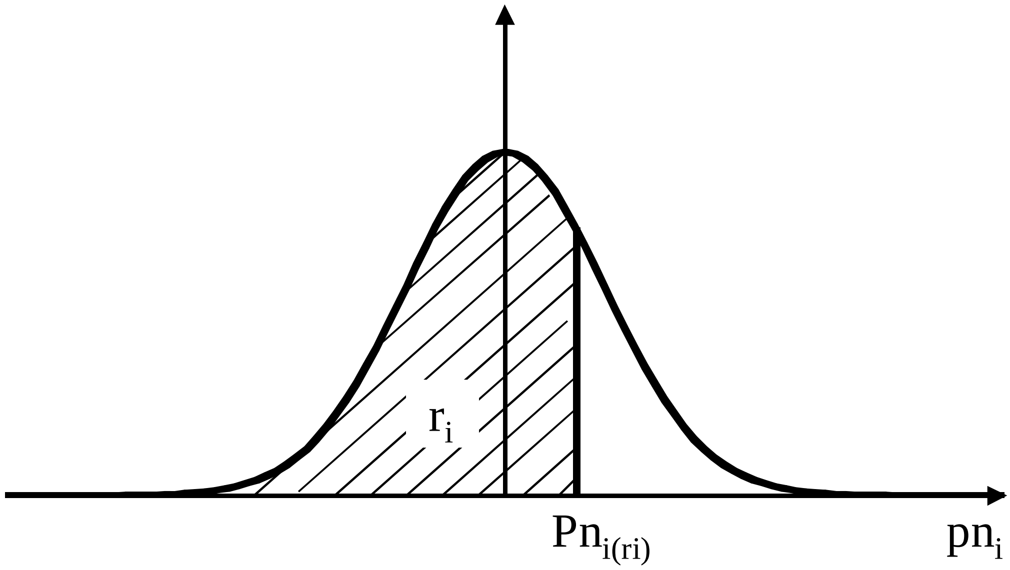


Figure 2. Decoding for real-coded ARGAs.

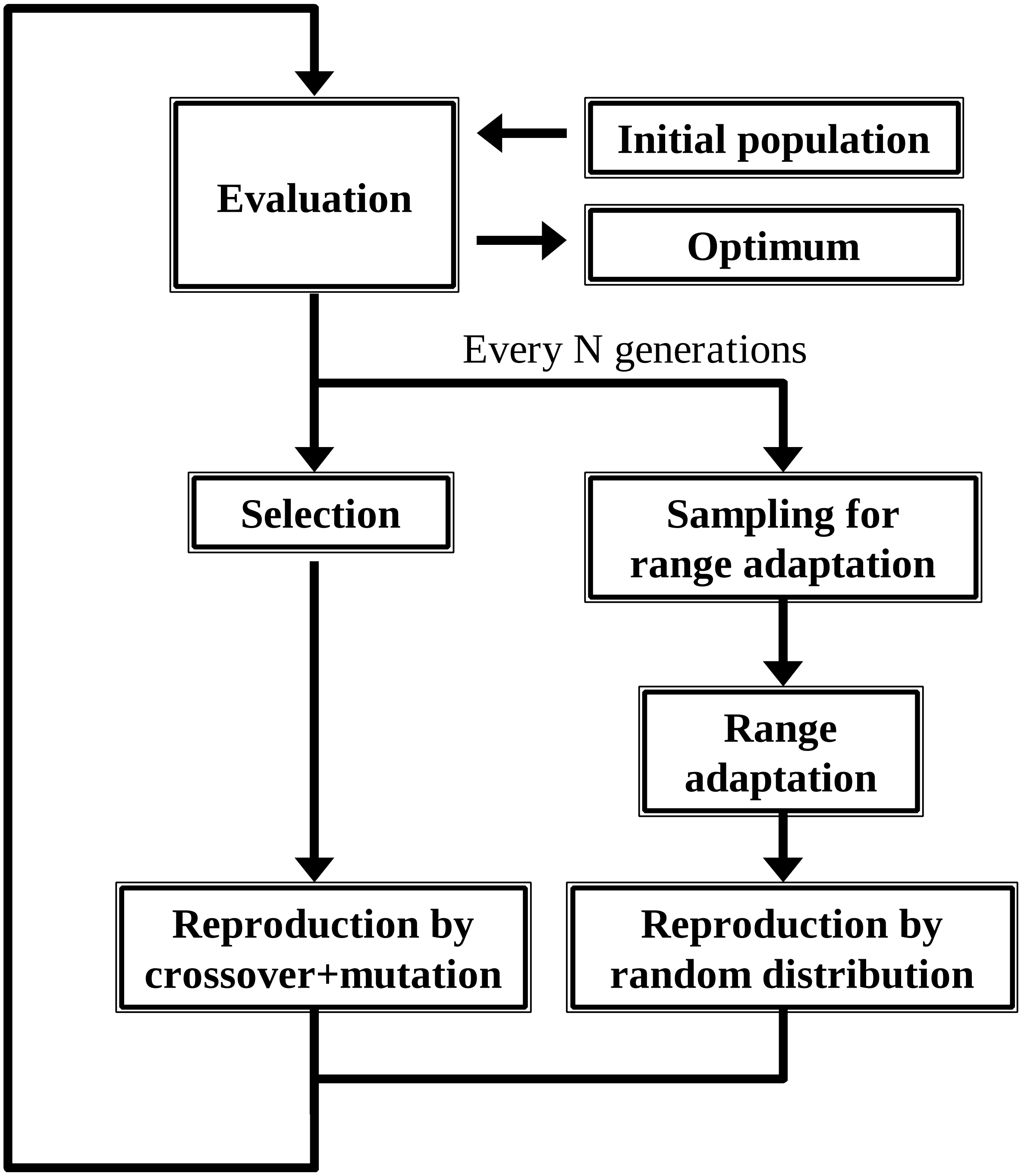


Figure 3. Flow chart of the present ARGA.

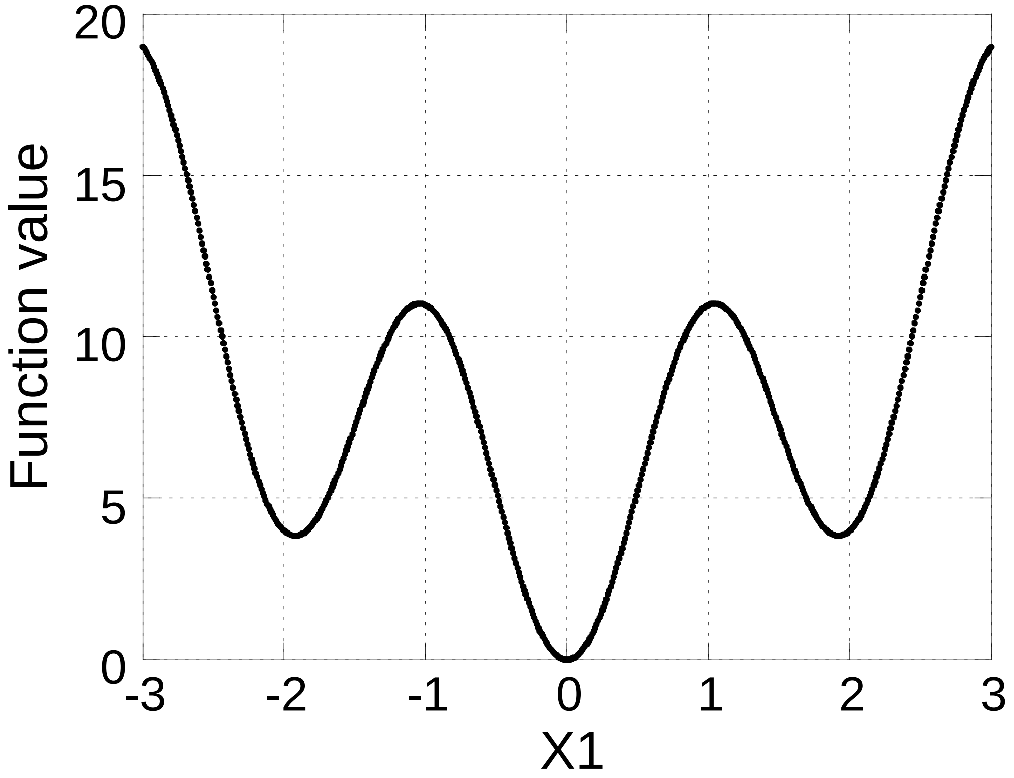
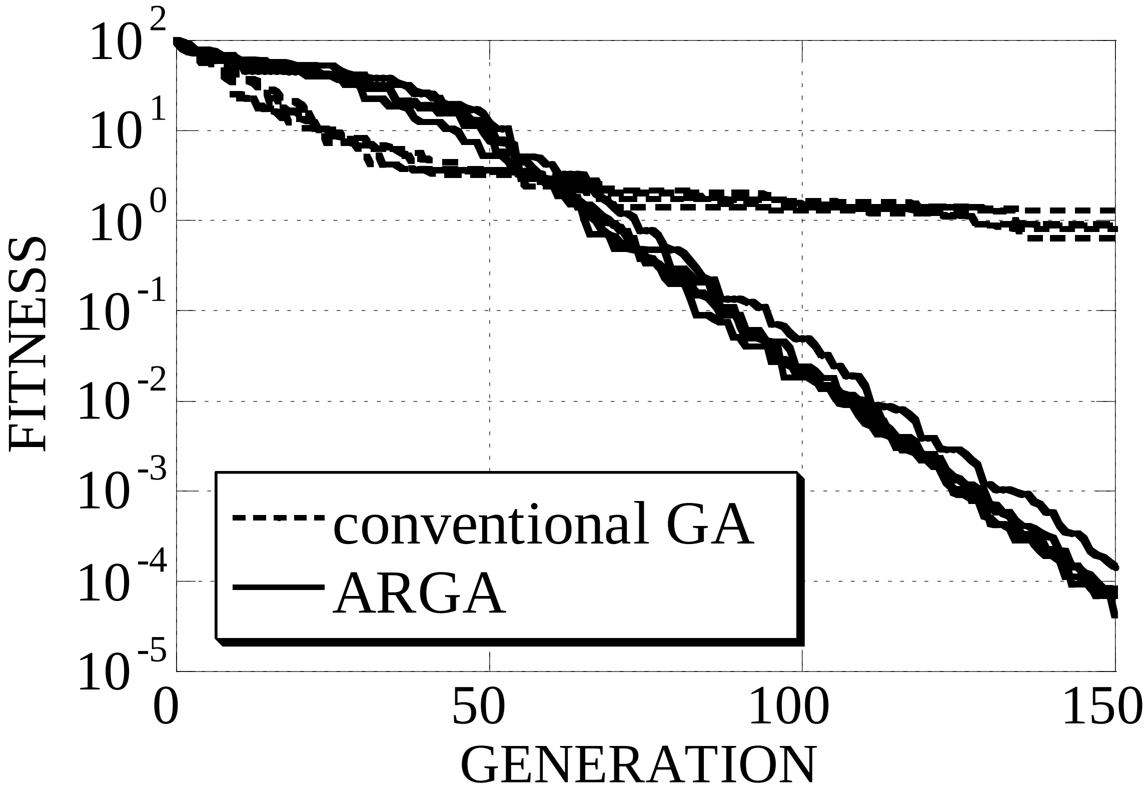


Figure 4. An one-dimensional version of F1.

Figure 5. Fitness history.



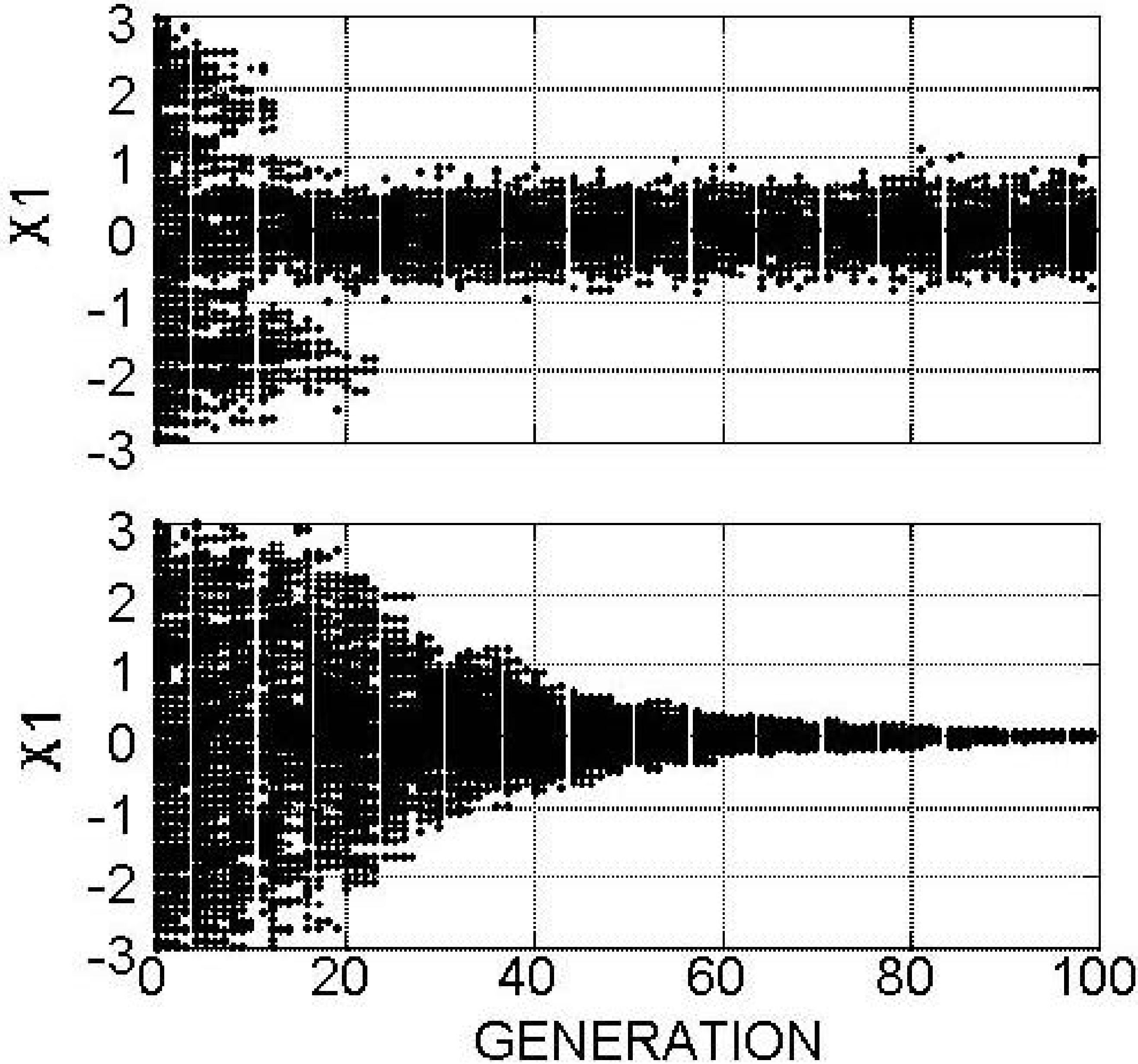


Figure 6. X1 histories of conventional GA(above) and ARGA(bellow).

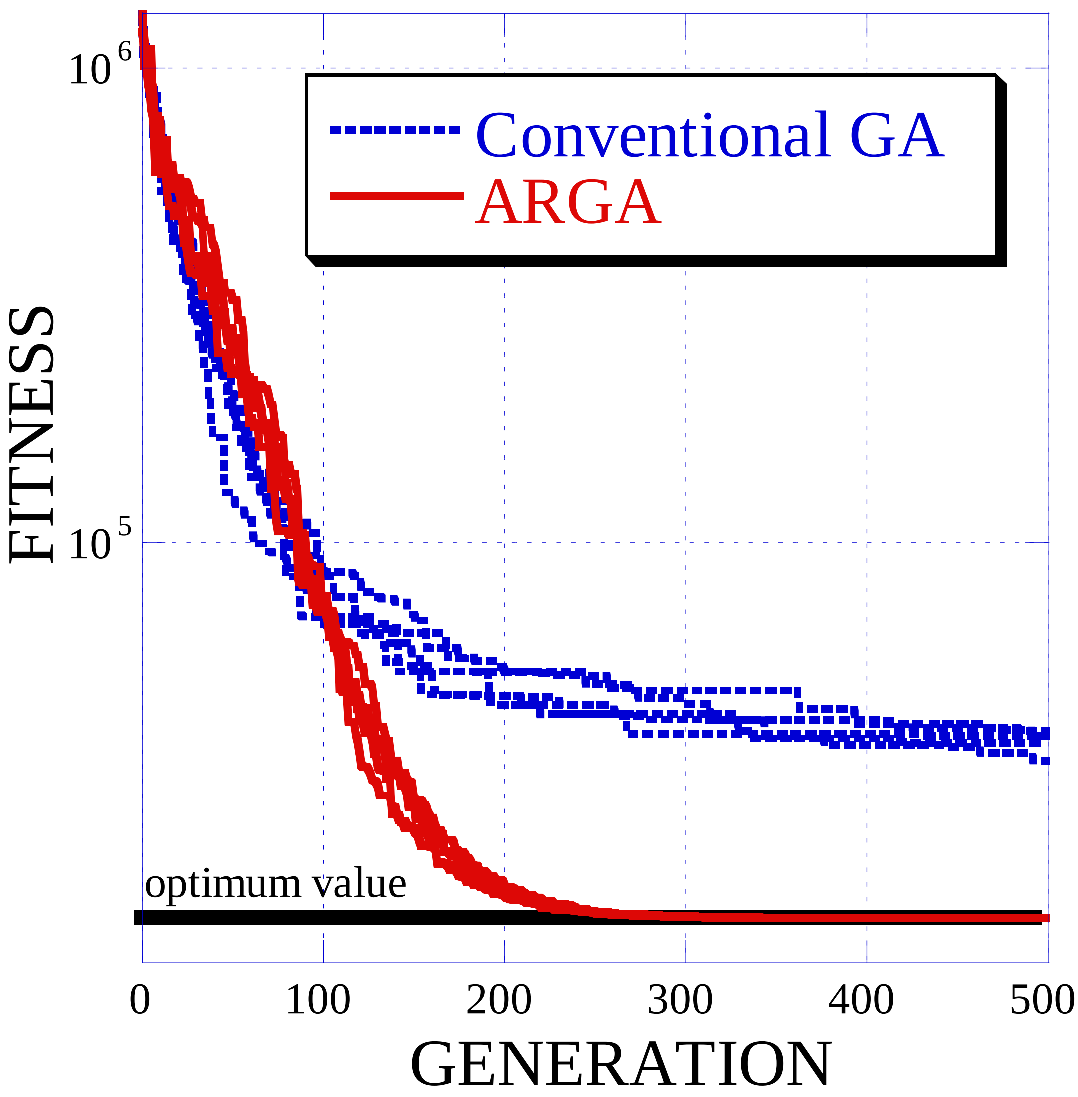


Figure 7. Optimization histories for the Dynamic control problem.

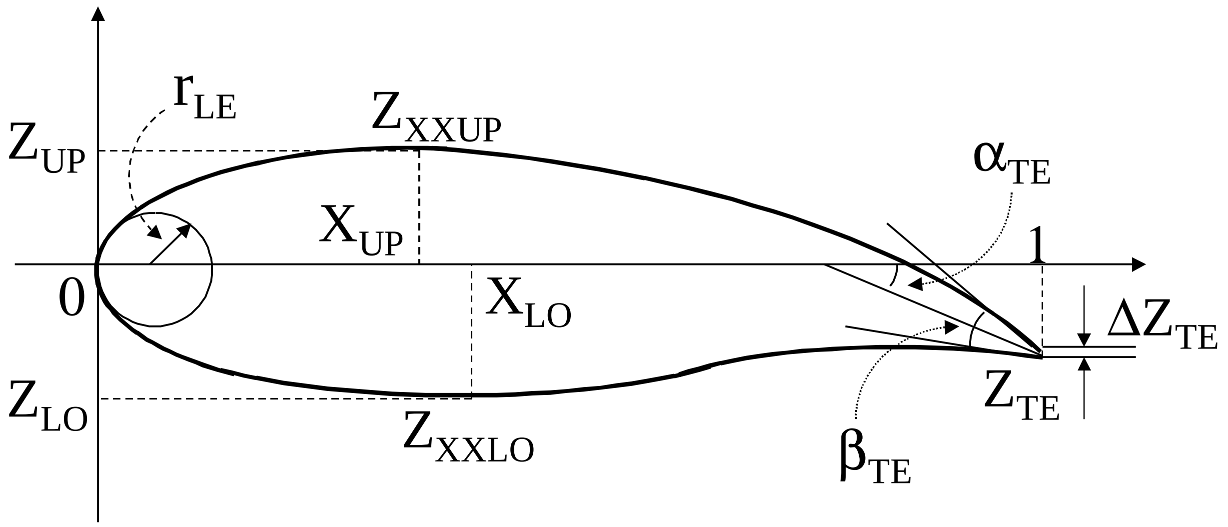


Figure 8. Design parameters for the Sobieczky shape functions.

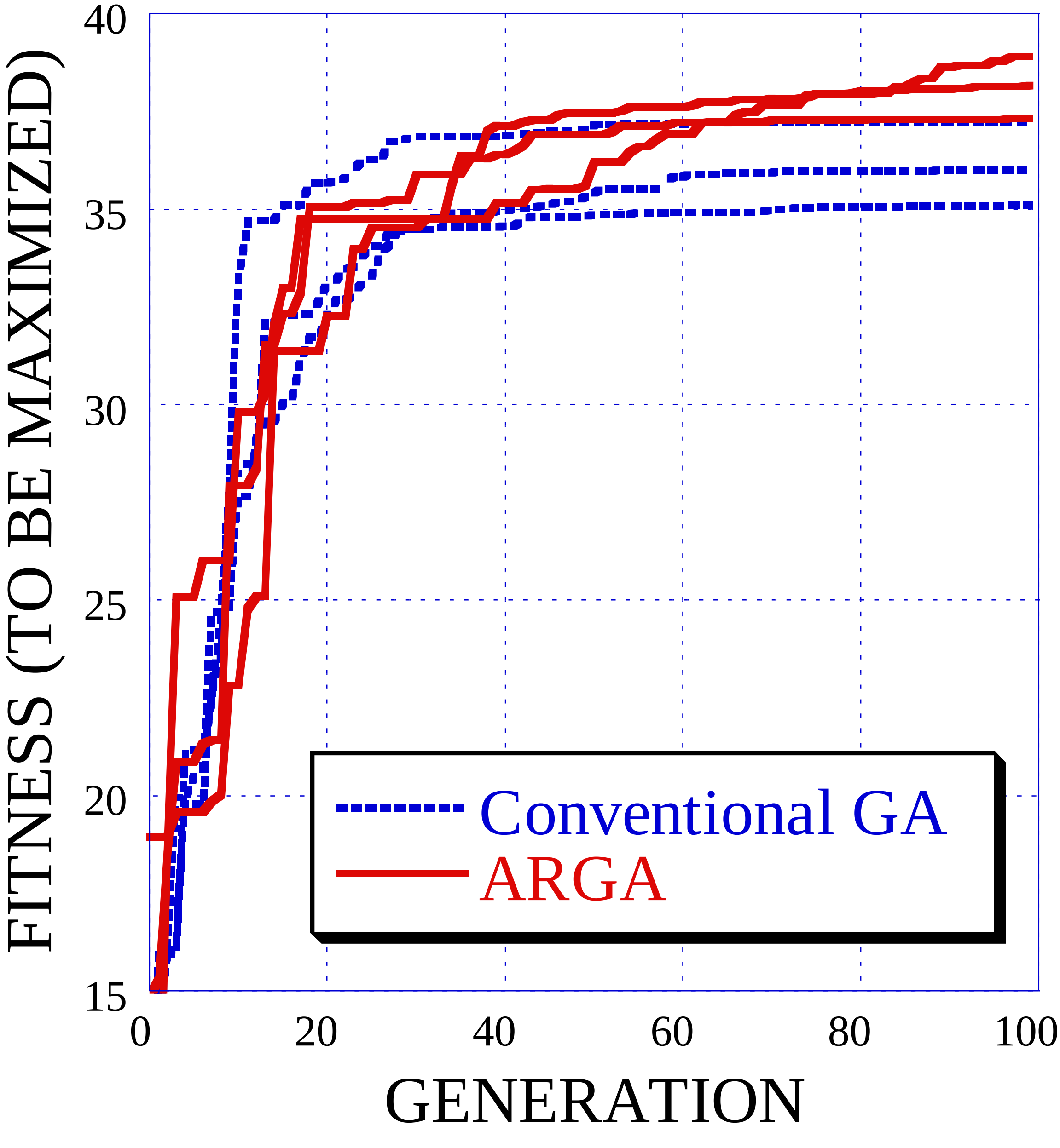


Figure 9. Optimization histories for the aerodynamic airfoil shape optimization.

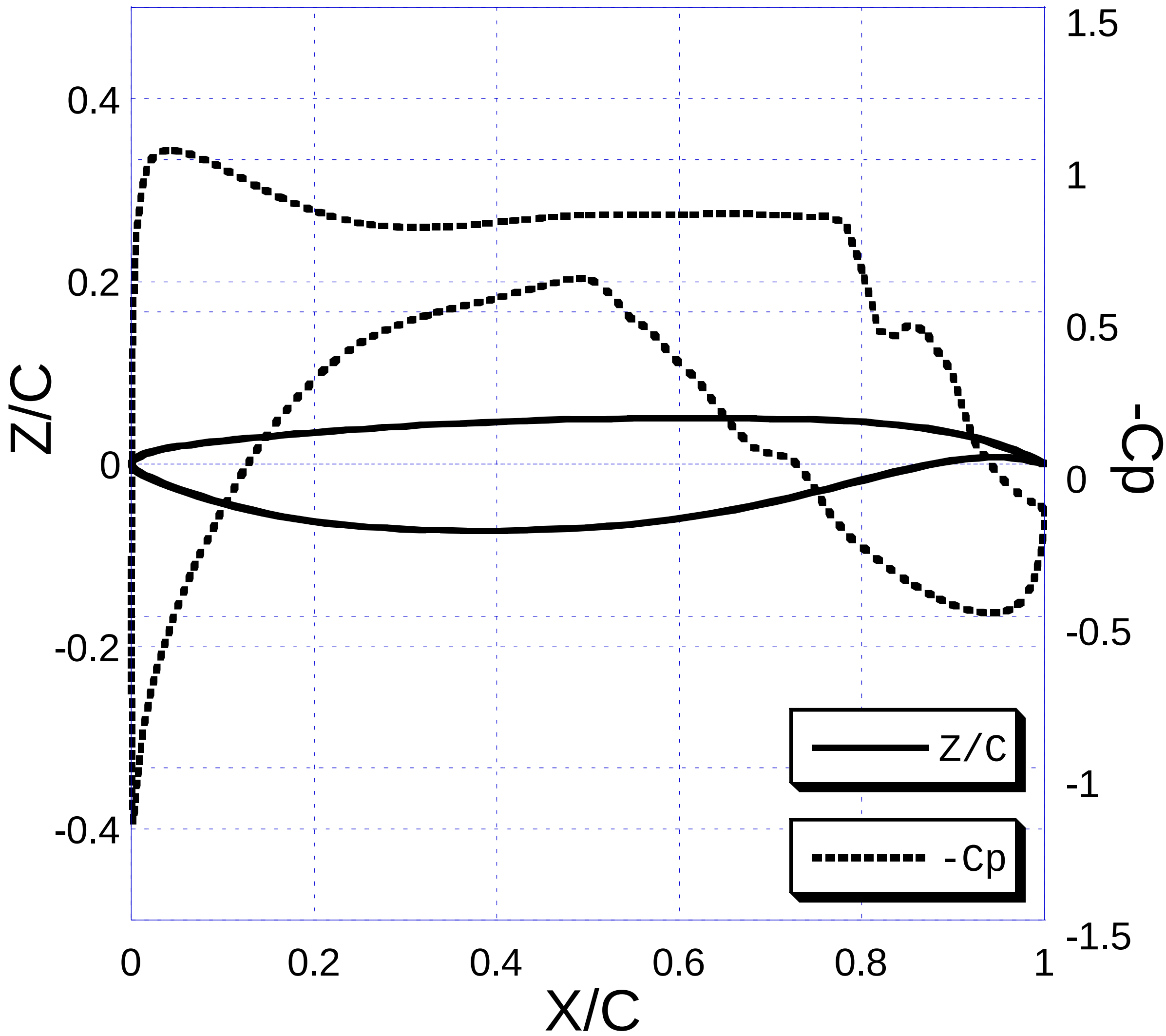


Figure 10. Designed airfoil shape and the corresponding pressure distribution.