AEROELASTIC OPTIMIZATION OF COMPOSITE WING BASED ON IMPROVED ADAPTIVE GENETIC ALGORITHM

Qiang Xiong *, Degang Cui **
*** Beijing Aerospace Technology Research Institute, CASIC, Beijing, China**
**** Science and Technology Committee, AVIC, Beijing, China**

Keywords: *adaptive genetic algorithm; dynamic population; composite wing; aeroelastic optimization*

Abstract

Having the constant population size and crossover/mutation probability, standard genetic algorithm (SGA) has such disadvantages as premature convergence, low stability and optimization efficiency for large design variables situation. This paper presents an improved adaptive genetic algorithm (IAGA), which adjusts the population size and the crossover/mutation probability adaptively and linearly, as well as integrating the IAGA running in a high performance parallel computing platform with high efficiency. The IAGA has been tested on an aeroelastic optimization of a composite wing. The case shows that the IAGA has realized improving the premature convergence, stability and optimization efficiency.

1 General Introduction

The aeroelastic optimization of composite wing is a problem with complex and huge scale design variables, including the layers' thickness, angle and stacking sequence in all optimization elements[1]. The genetic algorithm with some advantages like global searching capability and implied parallelism, etc, which could solve the optimization problem with discrete/continuous design variables in the composite aeroelastic tailoring design, shows very good application prospect[2-5].

Having the constant population size and crossover/mutation probability, standard genetic algorithm (SGA) has good robustness in solving the global optimal problem of simple structure, while it has such disadvantages as premature convergence, low robustness and optimization efficiency in complex structure[6]. At present, according to the SGA improvement research, people focus much time on crossover/mutation probability, and little on taking population size into account. Moreover, in the existing effects of improved genetic algorithm (IGA) [6,7], the majority of the IGAs have verified through some classic numerical function, and the rest of IGAs only do some optimization on simple engineering structures with small design variables. They didn't take any consideration on complex engineering optimization with huge scale design variables. It brings about an enormous searching space, finally affecting the genetic algorithm optimal precision, stability/robustness and optimization efficiency, etc.

This paper presents an improved adaptive genetic algorithm(IAGA) based on the above problems, emphasizing on improving the genetic algorithm operators. The IAGA has been tested on an aeroelastic optimization of composite wing with huge scale design variables, and the optimization results have been compared with that of SGA, thus demonstrating the IAGA's validity and rationality.

2 Adaptive Genetic Algorithm

2.1 Operation Process of Algorithm

This paper uses a set of existing genetic algorithm optimization system[8], which integrates the IAGA operators, and running the optimization example. The optimization system has been operated stably for nine years, through a lot of practical examples[1,4,5]. The optimization system process is shown in figure 1 the optimization workflow.

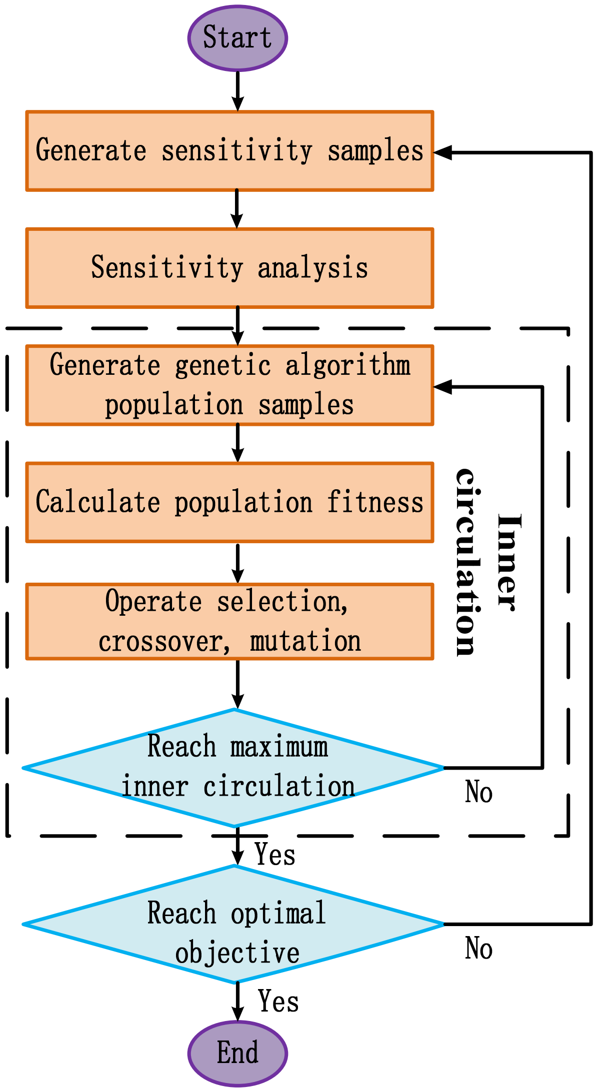

Fig. 1 the optimization workflow

The system implementation steps are as follows:

1) Establish coding. The chromosome constitutes the adding layers' thickness Δt and the adding layers' angle $\alpha(-90° \le \alpha \le 90°)$.
2) Initial the population. The population is generated through producing $M_i(i=1 \sim M)$ individuals randomly.
3) Establish the fitness evaluation formula, i.e. the non-dimensional flutter velocities of wing is greater than 1.
4) Genetic operation, which includes selection, crossover and mutation. Selection operator roulette wheel selection method as well as elitist strategy, ensuring the best individual can be completely inherited to the next generation. Crossover operator adopts single point crossover method. Mutation operator adopts allele's mutation method.
5) Determine whether the optimization reaches the maximum inner circulation number. If yes, complete the genetic evolution, output the best individual's fitness and gene value. If not, repeat from step 3) to step 5).

2.2 Strategy of Dynamically Adjusting Population Size

In a standard genetic algorithm, the population size is set by the user to a fixed value at the beginning of the search and remains constant through the entire run. Due to the size of solution space for different optimization problems being so different, the population size M in each generation of SGA is so difficult to determine. When the value of M is smaller than needed, it's likely to produce large sampling error, reduce the diversity of population and often lead the SGA to premature convergence, although the SGA can get a higher operation speed in optimization. On the contrary, when the value of M is larger than normal situation, it's likely to produce the waste of computing resources, moreover, to reduce the optimization efficiency of SGA[9], especially in engineering problems with large scale design variables. Usually optimization efficiency is the key factor of engineering optimization problems.

Therefore it's a difficult task to find an adequate population size. It has been shown, both theoretically and empirically, that the optimal size is something that differs from problem to problem. A somewhat widely accepted intuition behind population sizing is that it should be set proportionally to the problem's size and difficulty. However, problem difficulty is very hard to estimate for real-world problems, which brings us back to the difficulty of setting the appropriate population size[10]. At present, the value of population size M is often determined by its user subjectively and keeping constant in the evolution, which often produce large deviation in the practical application.

Based on these observations, the researchers have put forward various schemes that try to calculate a proper population size during the SGA running. Goldberg[11] etc gave the estimated formula of population size through studying the problem from SGA in theory, and Harik[12] etc improved that formula. Actually, it's not functional in practical application, because those parameters like the size, number and fitness variance of building blocks should be calculated first. This paper presents a strategy, which can adjust the next generation's population size dynamically according to the change of contemporary evolution algebra. The adjusting formula (1) is as follows:

$$\begin{cases} M_1 = M_{\max} & (i=1) \\ M_i = M_{i-1} - \kappa \times \dfrac{M_{\max} - M_{\min}}{N_{\max}} & \\ M_i = M_{\min} \quad (M_i \le M_{\min}) & (1 < i < N_{\max}) \end{cases} \tag{1}$$

Among the formula (1):

$M_{\max} / M_{\min}$ represent the maximum/minimum population size;

κ is a scale factor, generally taking 0.8~1.0;

$N_{\max}$ represents the maximum evolution generation;

M_i is population size of the i^{th} $(1 < i < N_{\max})$ generation.

From the formula (1), we can see that the population size decreases along with the inner loop generation increases. In the earlier stage of optimization, the individual's fitness difference between the average and the best is much greater; therefore a large scale population size should be kept. With the population evolving, the individual's fitness of the average is more and more close to the best, i.e. the population diversity becomes smaller and therefore a small scale population size should be used, in order to improve the algorithm's efficiency. The maximum/minimum population size should be constrained at the same time in the evolution.

2.3 Strategy of Dynamically Adjusting Crossover/ Mutation Probability

Both crossover probability P_c and mutation probability P_m have a great influence on the performance of genetic algorithm. If the value of P_c and P_m were chosen inappropriately, the good genes would be destroyed or hybridize with relatives, leading the evolution to premature convergence or slow convergence speed. At present, there is an effective method that the individuals' crossover/mutation probability is often determined by their fitness value in the evolution. This paper adopts the strategy of adaptively linear adjusting $P_{c,i}$ and $P_{m,i}$, and adjusting formula (2) and (3) are as follows:

$$P_{c,i} = \begin{cases} P_{c,\min} + (P_{c,\max} - P_{c,\min}) \times \dfrac{F_{\max} - F_i}{F_{\max} - F_{avg}}; & (F_i \ge F_{avg}) \\ P_{c,\max}; & (F_i < F_{avg}) \end{cases} \tag{2}$$

$$P_{m,i} = \begin{cases} P_{m,\min} + (P_{m,\max} - P_{m,\min}) \times \dfrac{F_{\max} - F_i}{F_{\max} - F_{avg}}; & (F_i \ge F_{avg}) \\ P_{m,\max}; & (F_i < F_{avg}) \end{cases} \tag{3}$$

Among the formula (2) and (3):

$P_{c,\max} / P_{c,\min}$ represent the maximum/minimum crossover probability, $0 \le P_{c,\min} \le P_{c,\max} \le 1$;

$P_{m,\max} / P_{m,\min}$ represent the maximum/minimum mutation probability, $0 \le P_{m,\min} \le P_{m,\max} \le 1$;

$F_{\max}, F_{avg}, F_{\min}$ represent the maximum/average/ minimum fitness respectively;

$F_i (i = 1 \sim M)$ represent the i^{th} individual's fitness.

From the formula (2) and (3), we can see that the values of $P_{c,i}$ and $P_{m,i}$ will increase when $F_{\max}$ is close to F_{avg}, in order to ensure the population diversity, so $P_{c,i}$ and $P_{m,i}$ have inverse ratio with $F_{\max} - F_{avg}$. On the contrary, the values of P_c and P_m have direct ratio with $F_{\max} - F_i$.

3 An Aeroelastic Optimization Example of Composite Wing

In order to demonstrate the IAGA's validity, the IAGA has been tested on an aeroelastic optimization of composite wing, and the optimization results have been compared with that of SGA.

The finite element model of composite wing is shown in figure 2. The model contains 1263 nodes and 2821 elements. 558 skin elements have been selected as optimal design variable units from the above and below wing skins, which can be seen from figure 2. The design variables are the adding layers' thickness Δt and the adding layers' angle $\alpha(-90° \le \alpha \le 90°)$ from the 558 skin elements, and the optimal design space is calculated as huge as 180×2^{558}. The optimization objective is to target the wing's flutter speed $V_f \ge 340 m/s$ under the constraint condition of every element layer's thickness being less than 15 millimeter. In the optimal computing, 1% total weight volume was added every optimal step until the result met the design requirements. The table 1 shows the operating parameters of SGA and IAGA.

The SGA operators and IAGA operators were integrated into the high performance parallel computing platform respectively for optimization, and the parallel computers' configurations are as follows: CPU--Intel Core2 Duo E8600(3.33GHz)， HD--320GB， RAM-2GB.

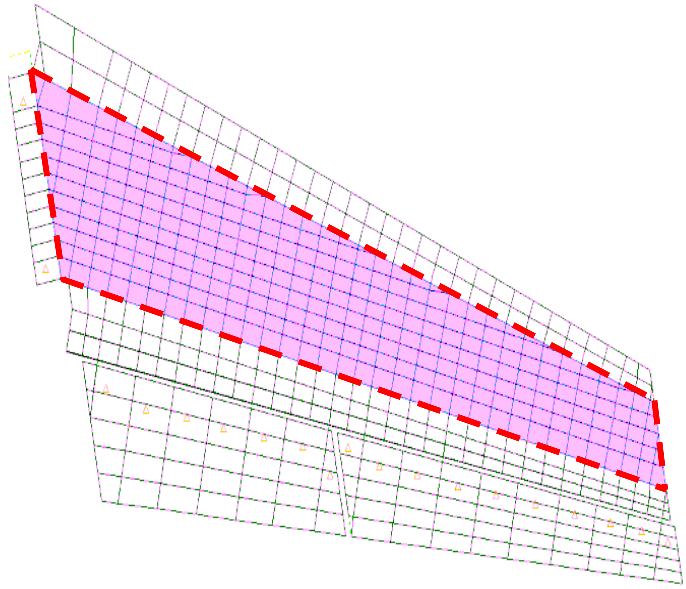

Fig. 2 the wing model and optimal design elements

Table 1 the operating parameters of SGA and IAGA

Algorithm parameter	SGA	IAGA
Population size/M	M=600	$M_{\max} = 600, M_{\min} = 300$
Crossover probability/Pc	P_c=0.8	$P_{c,\max} = 0.8, P_{c,\min} = 0.5$
Mutation probability/Pm	P_m=0.01	$P_{m,\max} = 0.05$ $P_{m,\min} = 0.005$
Inner circulation generation/N	15	10
Others		$\kappa = 1.0$

The figure 3 shows the optimization results of SGA optimization and IAGA optimization, respectively.

Firstly, from the figure 3, we can see that the flutter speed reaches to 340.94m/s after 16 IAGA optimal steps, while the SGA needs 22 steps coming to the same goal. Here a step means $N_{\max}$ evolution generation of inner circulation, shown in figure 1. Therefore it's not difficult to learn that the IAGA's optimization ability exceeds that of the SGA greatly, i.e. the SGA's premature convergence is improved by the IAGA effectively.

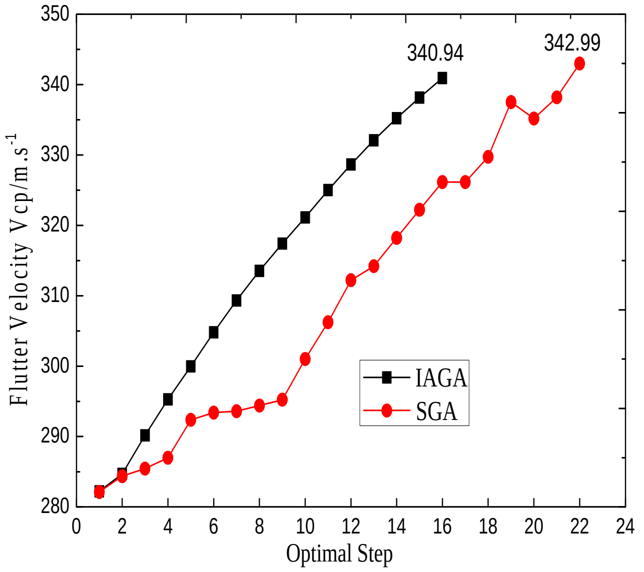

Fig. 3 the optimization results of SGA and IAGA

Secondly, from the optimization process of figure 3, the IAGA optimization curve has been rose steadily step by step, which could be ascribed to the algorithm stability. However, the optimization process seems fluctuate for SGA, which could be due to the unreasonable distribution of local stiffness in the optimization, leading to the change of wing's flutter mode. Therefore the phenomenon could be accounted for the IAGA's better stability than that of SGA in the optimization.

Thirdly, The IAGA optimization spends 1680 minutes for 16 steps i.e. 105 minutes for each step, while the SGA optimization spends 2970 minutes for 22 steps i.e. 135 minutes for each step. Compared with SGA optimization efficiency, the IAGA has increased by 43.43%. Therefore we can see that the IAGA, with the strategy of dynamically adjusting population size, could decrease the computing time in optimization greatly, i.e. increase the optimization efficiency substantially.

4 Conclusions

In summary, the IAGA could improve the premature convergence of SGA as well as the global optimal precision effectively; the optimization process of IAGA appears better stability; the IAGA spends less time than SGA on optimization, yet a same result. The IAGA has more practical application value for that optimization with huge scale design variables and complex structure.

This research is supported by the Major Program of the National Natural Science Foundation of China (Grant No.91330206).

References

[1] Wenyuan, C., Yan, C., etc, "Composite wing optimization design based on distributed computing", Acta Materiae Compositae Sinica, Vol. 24, Issue 1, 2007, pp.167-171

[2] Yingshu, X., Degang, C., "Optimum design of unsymmetrical and unbalanced composite laminate", Acta Aeronautica Et Astronautica Sinica, Vol. 25, Issue 2, 2004, pp.137-139

[3] Yingshu, X., Degang, C., "Optimum design of unsymmetrical and unbalanced composite laminate", Journal of Beijing University of Aeronautics and Astronautics, Vol. 30, Issue 9, 2004, pp.855-858

[4] Yingshu, X., Degang, C., "Optimal design of vertical tail using genetic-neural network algorithm", Journal of Computational Mechanics, Vol. 22, Issue 9, 2005, pp.796-800

[5] Wenyuan, C., Degang, C., "Optimization for composite wing based on Pareto genetic algorithm", Journal of Beijing University of Aeronautics and Astronautics, Vol. 33, Issue 2, 2007, pp.145-148

[6] Hangyu, K., Jing, J, etc, "Improving crossover and mutation for adaptive genetic algorithm", Computer Engineering and Applications, Vol. 12, 2006, pp.93-96

[7] Fei, Y., Weiji, L., "A stacking sequence optimization method of composite laminated panel based on an adaptive genetic algorithm", Journal of Northwestern Polytechnical University, Vol. 19, Issue 1, 2001, pp.156-159

[8] Degang, C., etc, "Grid-based multi-disciplinary optimization technology for aircraft (feature article)", Aeronautical Manufacturing Technology, Vol. 12, 2009, pp.26-31

[9] Ming, Z., Shudong, S., "Genetic algorithm principle and application", Beijing: National defense industry press (4th edition), 2002:90-97

[10] Fernando G. Lobo and Claudio F. Lima. Adaptive Population Sizing Schemes in Genetic Algorithms. Studies in Computatonal Intelligence, 2007,vol. 54.pp:185-204

[11] D. E. Goldberg, K. Deb, and J. H. Clark. Genetic algorithms. noise. and the sizing of populations. Complex System, 1992,vol. 6.pp:333-362

[12] Harik G R , Cantú2 Paz E , Goldberg D E , et al. The gambler's ruin problem: genetic algorithms and the sizing of populations [C]. In : BYCK T, ed.

Proceeding of the 4th International Conference on Evolutionary Computation .New York : IEEE Press , 1997. 7 12

Contact Author Email Address

dgcui@vip.163.com

Copyright Statement