Niching and Elitist Models for MOGAs

Shigeru Obayashi, Shinichi Takahashi and Yukihiro Takeguchi

Department of Aeronautics and Space Engineering
Tohoku University
Sendai, Japan
s.obayashi@computer.org

Abstract. This paper examines several niching and elitist models applied to Multiple-Objective Genetic Algorithms (MOGAs). Test cases consider a simple problem as well as multidisciplinary design optimization of wing planform shape. Numerical results suggest that the combination of the fitness sharing and the best-*N* selection leads to the best performance.

1. Introduction

Aircraft design presents a grand challenge to numerical optimization. It is in nature multidisciplinary among aerodynamics, structure, control and propulsion. Each disciplinary model has to be accurate enough to predict aircraft performance. Especially, aerodynamic calculation is computer intensive and the resulting aerodynamic performance is very sensitive to the geometry. Therefore, a robust optimization algorithm is indispensable to this field.

Evolutionary algorithms, Genetic Algorithms (GAs) in particular, are known to be robust (Goldberg, 1989) and have been enjoying increasing popularity in the field of numerical optimization in recent years. GAs have been applied to aeronautical problems in several ways, including parametric and conceptual design of aircraft (Bramlette et al., 1989), preliminary design of turbines (Powell et al., 1989), topological design of nonplanar wings (Gage et al., 1993) and aerodynamic optimization using Computational Fluid Dynamics (CFD) (for example, Quagliarella et al., 1998).

Furthermore, GAs can search for many Pareto-optimal solutions in parallel, by maintaining a population of solutions (Goldberg, 1989). Most real world problems require the simultaneous optimization of multiple, often competing objectives. Such multiobjective (MO) problems seek to optimize components of a vector valued objective function. Unlike the single-objective optimization, the solution to MO problem is not a single point, but a family of points known as the Pareto-optimal set. Each point in this set is optimal in the sense that no improvement can be achieved in one objective component that doesn't lead to degradation in at least one of the remaining components.

GAs can be very efficient, if they can sample solutions uniformly from the Pareto-optimal set. Since GAs are inherently robust, the combination of efficiency and robustness makes them very attractive for solving MO problems. Several approaches have been proposed (Schaffer, 1985, Fonseca et al., 1993 and Horn et al., 1994) and one of them to be employed here is called Multiple Objective Genetic Algorithms (MOGAs) (Fonseca et al., 1993).

Performance of MOGAs can be measured by variety of Pareto solutions and convergence to Pareto front. To construct a better MOGA, several niching and elitist models are examined in this paper through numerical tests.

2. MOGAs

The first three sections below describe basic GA operators used here. Then the extension to multiobjective optimization problems are discussed. Finally, the niching and elitist models are introduced.

2.1. Coding

In GAs, the natural parameter set of the optimization problem is coded as a finite-length string. Traditionally, GAs use binary numbers to represent such strings: a string has a finite length and each bit of a string can be either 0 or 1. For real function optimization, it is more natural to use real numbers. The length of the real-number string corresponds to the number of design variables.

As a sample problem, let's consider the following optimization:

Maximize: $f(x, y) = x + y$

Subject to: $x^2 + y^2 \leq 1$ and $0 \leq x, y \leq 1$

Let's represent the parameter set by using the polar coordinates here as

$$(x, y) = (r\cos\theta, r\sin\theta) \tag{1}$$

since the representation of the constraints will be simplified. Each point (x, y) in the GA population is encoded by a string (r, θ).

2.2. Crossover and mutation

A simple crossover operator for real number strings is the average crossover (Davis, 1990) which computes the arithmetic average of two real numbers provided by the mated pair. In this paper, a weighted average is used as

$$\begin{aligned} \text{Child1} &= ran1 \cdot \text{Parent1} + (1\text{-}ran1) \cdot \text{Parent2} \\ \text{Child2} &= (1\text{-}ran1) \cdot \text{Parent1} + ran1 \cdot \text{Parent2} \end{aligned} \tag{2}$$

where Child1,2 and Parent1,2 denote encoded design variables of the children (members of the new population) and parents (a mated pair of the old generation), respectively. The uniform random number *ran*1 in [0,1] is regenerated for every design variable. Because of Eq. (2), the number of the initial population is assumed even.

Mutation takes place at a probability of 20% (when a random number satisfies *ran*2 < 0.2). A high mutation rate is applied due to the real number coding. Equations (2) will then be replaced by

$$\begin{aligned} \text{Child1} &= ran1\cdot\text{Parent1} + (1\text{-}ran1)\cdot\text{Parent2} + m\cdot(ran3\text{-}0.5) \\ \text{Child2} &= (1\text{-}ran1)\cdot\text{Parent1} + ran1\cdot\text{Parent2} + m\cdot(ran3\text{-}0.5) \end{aligned} \tag{3}$$

where *ran*2 and *ran*3 are also uniform random numbers in [0,1] and m determines the range of possible mutation. In the following test cases, m was set to 0.4 for the radial coordinate r and $\pi/3$ for the angular coordinate θ.

2.3. Ranking

For a successful evolution, it is necessary to keep appropriate levels of selection pressure throughout a simulation (Goldberg, 1989). Scaling of objective function values has been used widely in practice. However, this leaves the scaling procedures to be determined. To avoid such parametric procedures, a ranking method is often used (Goldberg, 1989). In this method, the population is sorted according to objective function value. Individuals are then assigned an offspring count that is solely a function of their rank. The best individual receives rank 1, the second best receives 2, and so on. The fitness values are reassigned according to rank, for example, as an inverse of their rank values. Then the SUS method (Baker, 1987) takes over with the reassigned values. The method described so far will be hereon referred to as SOGA (Single-Objective Genetic Algorithm).

2.4. Multiobjective Pareto ranking

SOGA assumes that the optimization problem has (or can be reduced to) a single criterion (or objective). Most engineering problems, however, require the simultaneous optimization of multiple, often competing criteria. Solutions to multiobjective problems are often computed by combining multiple criteria into a single criterion according to some utility function. In many cases, however, the utility function is not well known prior to the optimization process. The whole problem should then be treated with non-commensurable objectives. Multiobjective optimization seeks to optimize the components of a vector-valued objective function. Unlike single objective optimization, the solution to this problem is not a single point, but a family of points known as the Pareto-optimal set.

By maintaining a population of solutions, GAs can search for many Pareto-optimal solutions in parallel. This characteristic makes GAs very attractive for solving MO problems. As solvers for MO problems, the following two features are desired: 1) the solutions obtained are Pareto-optimal and 2) they are uniformly sampled from the Pareto-optimal set. To achieve these with GAs, the following two techniques are successfully combined into MOGAs (Fonseca et al., 1993).

To search Pareto-optimal solutions by using MOGA, the ranking selection method described above for SOGA can be extended to identify the near-Pareto-optimal set within the population of GA. To do this, the following definitions are used: suppose

$\mathbf{x}_i$ and $\mathbf{x}_j$ are in the current population and $\mathbf{f} = (f_1, f_2, \cdots, f_q)$ is the set of objective functions to be maximized,

1. $\mathbf{x}_i$ is said to be dominated by (or inferior to) $\mathbf{x}_j$, if $\mathbf{f}(\mathbf{x}_i)$ is partially less than $\mathbf{f}(\mathbf{x}_j)$, i.e., $f_1(\mathbf{x}_i) \le f_1(\mathbf{x}_j) \wedge f_2(\mathbf{x}_i) \le f_2(\mathbf{x}_j) \wedge \cdots \wedge f_q(\mathbf{x}_i) \le f_q(\mathbf{x}_j)$ and $\mathbf{f}(\mathbf{x}_i) \ne \mathbf{f}(\mathbf{x}_j)$.
2. $\mathbf{x}_i$ is said to be non-dominated if there doesn't exist any $\mathbf{x}_j$ in the population that dominates $\mathbf{x}_i$.

Non-dominated solutions within the feasible region in the objective function space give the Pareto-optimal set.

As the first test case examined later in this paper, let's consider the following optimization:

Maximize: $f_1 = x$, $f_2 = y$

Subject to: $x^2 + y^2 \le 1$ and $0 \le x, y \le 1$

The Pareto front of the present test case becomes a quarter arc of the circle $x^2 + y^2 = 1$ at $0 \le x, y \le 1$.

Consider an individual $\mathbf{x}_i$ at generation t (Fig. 1) which is dominated by p_i^t individuals in the current population. Following Fonseca et al. (1993), its current position in the individuals' rank can be given by

$$\text{rank}(\mathbf{x}_i, t) = 1 + p_i^t \qquad (4)$$

All non-dominated individuals are assigned rank 1 as shown in Fig. 1. The fitness assignment according to rank can be done similar to that in SOGA.

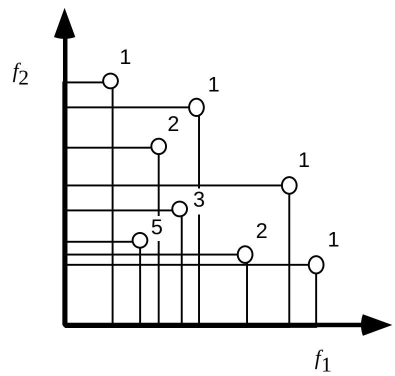

Fig. 1. Pareto ranking method.

2.5. Fitness sharing

To sample Pareto-optimal solutions from the Pareto-optimal set uniformly, it is important to maintain genetic diversity. It is known that the genetic diversity of the population can be lost due to the stochastic selection process. This phenomenon is called the random genetic drift. To avoid such phenomena, the niching method has been introduced (Goldberg, 1989). In this paper, two specific niching models are examined for MOGAs.

The first model is called fitness sharing (FS). A typical sharing function is given by Goldberg (1989). The sharing function depends on the distance between individuals. The distance can be measured with respect to a metric in either genotypic or phenotypic space. A genotypic sharing measures the interchromosomal Hamming distance. A phenotypic sharing, on the other hand, measures the distance between the designs' objective function values. In MOGAs, a phenotypic sharing is usually preferred since we seek a global tradeoff surface in the objective function space.

This scheme introduces new GA parameters, the niche size σ_{share}. The choice of σ_{share} has a significant impact on the performance of MOGAs. In our experiences, it is very difficult to determine its value on the trial-and-error basis. Fonseca et al. (1993) gave a simple estimation of σ_{share} in the objective function space as

$$N\sigma_{share}^{q-1} - \frac{\prod_{i=1}^{q}\left(M_i - m_i + \sigma_{share}\right) - \prod_{i=1}^{q}\left(M_i - m_i\right)}{\sigma_{share}} = 0 \tag{5}$$

where N is a population size, q is a dimension of the objective vector, and M_i and m_i are maximum and minimum values of each objective, respectively. This formula has been successfully adapted here. Since this formula is applied at every generation, the resulting σ_{share} is *adaptive* to the population during the evolution process. Niche counts can be consistently incorporated into the fitness assignment according to rank by using them to scale individual fitness within each rank.

2.6. Coevolutionary shared niching

Coevolutionary shared niching (CSN) is an alternate, new niching method proposed in Goldberg et al. (1998). The technique is loosely inspired by the economic model of monopolistic competition, in which businessmen locate themselves among geographically distributed populations – businessmen and customers – where individuals in each population seek to maximize their separate interests thereby creating appropriately spaced niches containing the most highly fit individuals.

The customer population may be viewed as a modification to the original sharing scheme, in which the sharing function and σ_{share} are replaced by requiring customers to share within the closest businessman's service area. In other words, a customer is supposed to be served by the nearest businessman. The number of customers a businessman serves becomes the niche count. Then, a customer's raw fitness is divided by the niche count similar to the original sharing scheme.

The evolution of the businessman population is conducted in a way that promotes the independent establishment of the most highly fit regions or niches in the search space. The businessman population is created by an *imprint* operator that carries the best of one population over the other. Simply stated, businessmen are chosen from the best of the customer population.

This model introduces a new GA parameter d_{min} that determines the minimum distance between the businessmen. In the following test cases, this parameter d_{min} was tuned by the trial-and-error basis and kept constant during the evolution. Niche counts were incorporated into the fitness assignment according to rank similar to the fitness sharing.

2.7. Generational models

To examine effects of generational models, two elitist models are considered here. The first one is the elitist recombination (ER) model that selects two best individuals among two parents and their two offsprings. The other model is the so-called best-*N* (BN) model that selects the best *N* individuals among *N* parents and *N* children similar to CHC (Eshelman, 1991). These models are compared with the simple generational (SG) model that replaces *N* parents simply with *N* children. The population size was kept to 100 in all test cases.

3. Comparison of Niching and Elitist Models

From the techniques described above, five optimization results are shown here for the first test case to maximize $f_1 = x$ and $f_2 = y$, subject to $x^2 + y^2 \leq 1$ and $0 \leq x, y \leq 1$. Figures 2 to 4 show the results obtained from the simple generational model with the fitness sharing (SG + FS), the elitist recombination with the fitness sharing (ER + FS) and the best-*N* with the fitness sharing (BN + FS), respectively. The GA population is represented by dots and the Pareto front is indicated by a gray arc. When FS was used, the results were improved by the stronger elitist model. Among the three generational models examined here, the best-*N* selection BN was the best.

Figure 5 shows the results obtained from SG + CSN in gray dots and from BN + CSN in black dots (d_{min} = 0.028). The distribution of the gray dots are almost as good as that of the black dots. It also indicates that the coevolutionary shared niching CSN provides a significant improvement over FS when combined with SG. The result obtained from ER + CSN did not show any further improvement and thus it is not plotted here. Only minor improvements were obtained by using the elitist models.

The use of the adaptive σ_{share}, Eq. (5), seems to give better performance than the use of the hand-tuned, constant d_{min} when combined with BN. To confirm this observation, the results are compared in terms of average values of (r, θ) after 30 generations over five different runs as shown in Table 1. Better solutions should have r closer to 1 and θ closer to 45 deg (the uniform distribution in θ should give the average of 45 deg). For comparison, σ_{share} was also tuned by trail and error (σ_{share} = 0.11).

Table 1. Performance comparison of the niching parameters

	constant σ_{share}	constant d_{min}	adaptive σ_{share}
average r	0.9948	0.9914	0.9958
average θ	44.27 deg	44.78 deg	45.25 deg

This confirms that BN + FS with the adaptive σ_{share} gives the best performance. It further leads to a speculation: "adaptive σ_{share} (FS) < adaptive d_{min} (CSN) ?" CSN is very promising but further investigations will be needed, especially in the area of how to determine its parameter d_{min}.

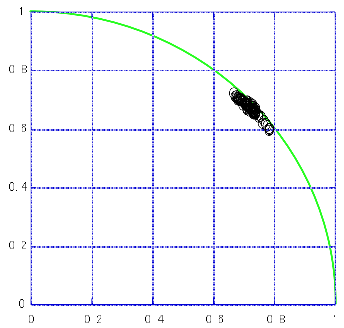

Fig. 2. Pareto solutions by SG + FS.

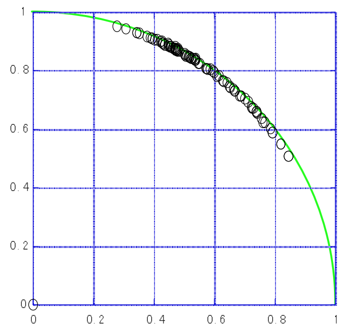

Fig. 3. Pareto solutions by ER + FS.

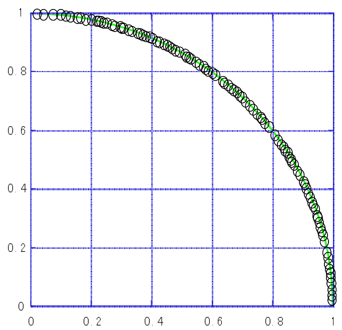

Fig. 4. Pareto solutions by BN + FS.

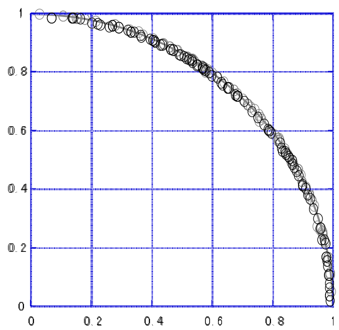

Fig. 5. Pareto solutions by SG + CSN (gray dots) and BN + CSN (black dots).

4. Multidisciplinary Optimization of Wing Planform Design

An application of MOGA to multidisciplinary optimization (MDO) of wing planform design (Takahashi et al., 1998) is examined in this section. The present multiobjective optimization problem is described as follows:

1. Minimize aerodynamic drag (induced + wave drag)
2. Minimize wing weight
3. Maximize fuel weight (tank volume) stored in wing

under these constraints:

1. Lift to be greater than given aircraft weight
2. Structural strength to be greater than aerodynamic loads

Since the purpose of the present design is to examine the performance of MOGAs as a system-level optimizer, the design variables for wing geometry are greatly reduced. First, aircraft sizes were assumed as wing area of 525 ft^2 and total maximum takeoff weight of 45,000 *lb* at cruise Mach number of 0.75. Next, as a baseline geometry, a transonic wing was taken from a previous research (Fujii et al., 1987). The original wing has an aspect ratio of 9.42, a taper ratio of 0.246 and a sweep angle at the quarter chord line of 23.7 deg. Its airfoil sections are supercritical and their thickness and twist angle distributions are reduced toward the tip. Then, only two parameters are chosen as design variables: sweep angle and taper ratio.

The objective functions and constraints are computed as follows. First, drag is evaluated, using a potential flow solver called FLO27 (Jameson et al, 1977). The code can solve subsonic and transonic flows. From the flow field solution, lift and drag can be postprocessed. Since the flow is assumed inviscid, only a sum of the induced and wave drag is obtained. Second, wing weight is calculated, using an algebraic weight equation as described in Torenbeek (1982). Third, the fuel weight is calculated directly from the tank volume given by the wing geometry. Finally, the structural model is taken from Wakayama et al., (1995). In this research, the wing box is modeled only for calculating skin thickness. Then the wing is treated as a thin-walled, single cell monocoque beam to calculate stiffness. Flexibility of the wing is ignored. The objective function values and constraints' violations are now passed on to the system-level optimizer. MOGA is employed as the system-level optimizer here. When any constraint is violated, the rank of a particular design is lowered by adding 10.

In this section, the elitist model was frozen to BN and the results were compared between two niching models, FS and CSN. Figure 6 shows the resulting Pareto front obtained from BN + FS. BN + CSN gave a similar Pareto front and thus the result is not presented here. The major difference of the two, however, appears in the convergence history. As shown in Fig. 7, FS was able to converge the population to the Pareto front, but CSN was not. This is probably because of the adaptive σ_{share} used in FS. This result again suggests a need of an adaptive d_{min}. Figure 8 shows wing planform shapes of the resulting Pareto solutions.

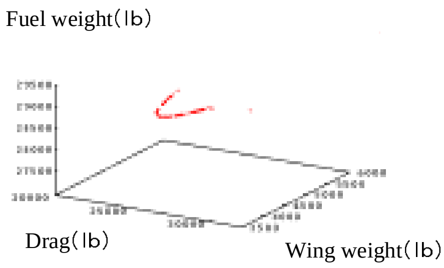

Fig. 6. Pareto solutions in objective function space.

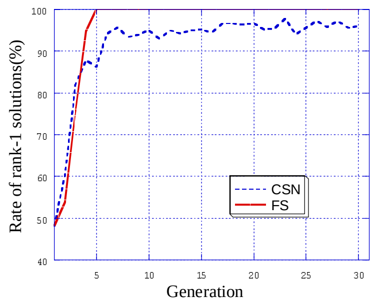

Fig. 7. Convergence history (average of five different runs).

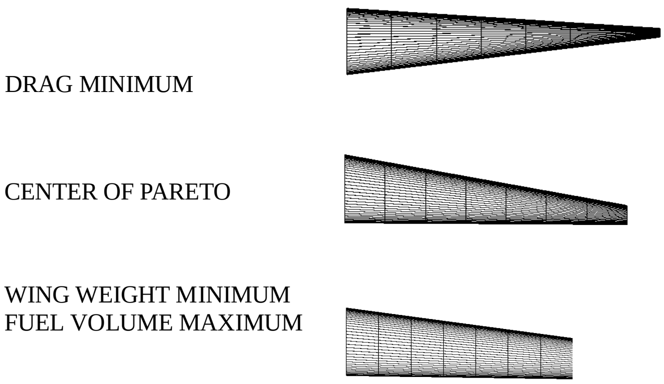

Fig. 8. Planform shapes of Pareto solutions.

5. Conclusion

Niching and elitist models have been examined for multiobjective Genetic Algorithms (MOGAs). The fitness sharing and coevolutionary shared niching models were considered for niching. Two elitist models, the elitist recombination and the best-N selection were compared with the simple generational model. The test cases indicate that the combination of the fitness sharing and the best-*N* selection provides the best performance for MOGAs so far. The results also suggest a need of an adaptive formula for d_{min} in the coevolutionary shared niching scheme.

6. Acknowledgment

This work has been supported by Bombardier Aerospace, Toronto, Canada.

References

1 Baker, J. E.: Reducing bias and inefficiency in the selection algorithm, Proceedings of the Second International Conference on Genetic Algorithms, Erlbaum, Hillsdale (1987) 14-21.
2 Bramlette, M. F. and Cusic, R.: A comparative evaluation of search methods applied to the parametric design of aircraft, Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, Inc., San Mateo (1989) 213-218.
3 Davis, L.: *Handbook of Genetic Algorithms*, Van Nostrand Reinhold (1990).
4 Fonseca C. M., and Fleming, P. J.: Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, Proceedings of the 5th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, Inc., San Mateo (1993) 416-423.
5 Fujii, K. and Obayashi, S.: Navier-Stokes simulations of transonic flows over a practical wing configuration, *AIAA Journal*, 25 (3) (1987) 369-370.
6 Gage, P. and Kroo, I.: A role for genetic algorithms in a preliminary design environment, AIAA Paper 93-3933 (1993).
7 Goldberg, D. E.: *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley Publishing Company, Inc., Reading (1989).
8 Goldberg, D. E. and Wang, L.: Adaptive niching via coevolutionary sharing. In Quagliarella, D., Periaux, J., Poloni, C. and Winter, G. (Eds.), *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, John Wiley and Sons, Chichester (1998) 21-38.
9 Horn, J., Nafplitois, N. and Goldberg, D., E.: A niched Pareto genetic algorithm for multiobjective optimization, Proceedings of the 1st IEEE Conference on Evolutionary Computation (1994) 82-87.
10 Jameson, A. and Caughey, D. A.: Numerical calculation of the transonic flow past a swept wing, COO-3077-140, New York University, July (1977) (also NASA-CR 153297).
11 Powell, D. J., Tong, S. S. and Skolnick, M. M.: EnGENEous domain independent, machine learning for design optimization, Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, Inc., San Mateo (1989) 151-159.
12 Quagliarella, D., Periaux, J., Poloni, C. and Winter, G. (Eds.): *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, John Wiley and Sons, Chichester (1998). See Chapters 12-14.
13 Schaffer, J. D.: Multiple objective optimization with vector evaluated genetic algorithm, Proceedings of the 1st International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, Inc., San Mateo (1985) 93-100.
14 Takahashi, S., Obayashi S. and Nakahashi, K.: Inverse optimization of transonic wing shape for mid-size regional aircraft, AIAA Paper 98-0601, AIAA Aerospace Sciences Meeting & Exhibit, Reno NV (January 12-15, 1998).
15 Torenbeek, E.: *Synthesis of Subsonic Airplane Design*, Kluwer Academic Publishers, Dordrecht (1982).
16 Wakayama, S. and Kroo, I.: Subsonic wing planform design using multidisciplinary optimization, *Journal of Aircraft*, 32 (4) July-August (1995) 746-753.