Supersonic Transport Design Optimization Based on Evolutionary Algorithms

Shigeru Obayashi

Department of Aeronautics and Space Engineering
Tohoku University
Sendai, 980-8579, JAPAN
s.obayashi@computer.org
http://www.ad.mech.tohoku.ac.jp/~obayashi

Introduction

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 [1]. 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.

Pareto-optimal solutions might be obtained by solving appropriately formulated single-objective optimization problems on a one-by-one basis. To solve the single-objective optimization, gradient-based methods have been the most widely used. The optimum obtained from these methods will be a global optimum, if the objective and constraints are differentiable and convex [2]. In practice, however, it is very difficult to prove differentiability and convexness. One could only hope for a local optimum neighboring the initial point, provided the gradient is well-defined. Therefore, to determine the global optimum, one must optimize from a number of initial points and check for consistency in the optima obtained. In this sense, the gradient-based methods are not robust.

Evolutionary algorithms, Genetic Algorithms (GAs) in particular, are known to be robust [3] and have been enjoying increasing popularity in the field of numerical optimization in recent years. GAs are search algorithms based on the mechanics of natural selection and natural genetics. One of the key features of GAs is that they search from a population of points and not from a single point. In addition, they use objective function information (fitness value) instead of derivatives or other auxiliary knowledge. These features make GAs robust and thus attractive to practical engineering applications. GAs have been applied to aeronautical problems in several ways, including parametric and conceptual design of aircraft [4,5], preliminary design of turbines [6], topological design of nonplanar wings [7] and aerodynamic optimization using Computational Fluid Dynamics (CFD) [8-14].

furthermore, GAs can search for many Pareto-optimal solutions in parallel, by maintaining a population of solutions [3]. In general, GAs are not considered efficient since they require a large number of function evaluations. However, if GAs can sample solutions uniformly from the Pareto-optimal set, then they will turn out to be very efficient. In contrast, when solving the single-objective optimization problem formulated appropriately from multiple objectives, Pareto-optimal solutions have to be sought on a one-by-one basis.

Since GAs are inherently robust, the combination of efficiency and robustness makes them very attractive for solving MO problems. Several approaches have been proposed [14-18] and one of them to be employed here is called Multiple Objective Genetic Algorithms (MOGAs) [16].

In this paper, MOGA is applied to multidisciplinary optimization problems of transonic and supersonic wings. Similar studies have been reported, for example, in [19-22]. This paper will discuss the feasibility of MOGAs applied to the aerodynamic and structural optimization of wing planform shapes using relatively simple disciplinary models.

Genetic Algorithms

GAs are search algorithms based on the mechanics of natural selection and natural genetics. GAs work from a rich database of points (a population of strings), simultaneously climbing many peaks in parallel. Thus, the probability of finding a false peak is reduced with GAs as compared with the conventional methods that go from point to point like the gradient-based methods. In this section, various techniques used in GAs will be reviewed briefly.

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 simple test case, 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). In the following test case, a population of 100 strings will be used where the initial strings are generated randomly.

2. SOGA

At each generation (iteration) of GA's process, fitness value (objective function value) of every individual is evaluated and used to specify its probability of reproduction. A new population is generated from selected parents by performing specific operators on their genes. These operators are briefly explained here.

Simple GA is composed of three operators: 1. Reproduction, 2. Crossover, 3. Mutation

[3]. Reproduction is a process in which individual strings are copied according to their fitness values. This implies that strings with a higher fitness value have a higher probability of contributing one or more offsprings in the next generation. A typical reproduction operator is the roulette-wheel method described in [3]. The reproduction process produces a mating pool. Then crossover proceeds in two steps. First, members in the mating pool are mated at random. Second, each pair of strings undergoes partial exchange of their strings at a random crossing site. This results in a pair of strings of a new generation. Mutation is a bit change of a string that occurs during the crossover process at a given mutation rate. Mutation implies a random walk through the string space and plays a secondary role in simple GA. A flowchart of the present GA is illustrated in Fig. 1.

A simple crossover operator for real number strings is the average crossover [23] 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 $ran1$ 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 $ran2 <$ 0.1). 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 $ran2$ and $ran3$ 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.

3. Ranking

For a successful evolution, it is necessary to keep appropriate levels of selection pressure throughout a simulation [3]. Starting from a randomly created population, there is a tendency for a few superindividuals to dominate early on in the selection process. In this case, objective function values must be scaled back to prevent takeover of the population by these superstrings. When the population is largely converged, competition among population members is less strong and the simulation tends to wander. In this case objective function values must be scaled up to accentuate differences between population members to continue rewarding the best performers.

Scaling of objective function values has been used widely in practice [3]. However, this leaves the scaling procedures to be determined. To avoid such parametric procedures, a ranking method is often used [3]. 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 usual roulette-wheel method takes over with the reassigned values. The method described so far will be hereon referred to as SOGA (Single-Objective Genetic Algorithm).

4. MOGA

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. Such problems are called multiobjective (MO) or multicriteria problems. These problems have long attracted the attention of researchers using traditional techniques of optimization. More recently, GAs have been applied to the search for multicriteria optima [1].

The solution to multiobjective problems is 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. As a second test case in this section, let's consider the following optimization:

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

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

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. Each point in this set is optimal in the sense that no improvement can be achieved in one objective component that does not lead to degradation in at least one of the remaining components. The Pareto front of the present test case becomes a quarter arc of the circle $x^2 + y^2 = 1$ at $0 \le x,\ y \le 1$.

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 [16].

5. Pareto ranking

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 $\boldsymbol{x}_i = (x_i, y_i)$ and $\boldsymbol{x}_j = (x_j, y_j)$ are in the current population and $\boldsymbol{f} = (f_1, f_2)$ is the set of objective functions to be maximized,

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

Non-dominated solutions within the feasible region in the objective function space give the

Pareto-optimal set.

Consider an individual x_i at generation t (Fig. 2) which is dominated by p_i^t individuals in the current population. Its current position in the individuals' rank can be given by

$$\mathrm{rank}(\boldsymbol{x}_i,\, t) = 1 + p_i^t \tag{4}$$

All non-dominated individuals are assigned rank 1 as shown in Fig. 2. The fitness assignment according to rank can be done similar to that in SOGA.

6. 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 their stochastic selection process. This phenomenon is called the random genetic drift [3].

Suppose that SOGA is applied to the objective function with equal peaks as explained in [4]. In an initial population chosen uniformly at random, a relatively even spread of points across the function domain is obtained. As generations pass, the population climbs the hills, ultimately distributing most of the strings near the top of one hill among the others. The population converges to one peak or another without differential advantage. This is caused by the genetic drift, stochastic errors in sampling caused by small population sizes. To reduce the effect of these errors and to form stable subpopulations around each peak, the inducement of niche and species is considered.

To form the niche, the strings on a particular peak are forced to share the fitness value. If five strings are on a peak, each string receives only one fifth of the fitness value corresponding to the particular peak. The incorporation of forced sharing causes the formation of stable subpopulations (species) on different peaks (niches) in the problem.

A practical scheme is given by taking the raw fitness and dividing through by the accumulated number of shares

$$\boldsymbol{f}_s(\boldsymbol{x}_i) = \frac{\boldsymbol{f}(\boldsymbol{x}_i)}{\sum_j s\left(d(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)} \tag{5}$$

where $s(d)$ is a sharing function that determines the neighborhood and degree of sharing. The distance $d = d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 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.

A typical sharing function is given by

$$s(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{share}}\right)^{\alpha} & d < \sigma_{share} \\ 0 & d \geq \sigma_{share} \end{cases} \tag{6}$$

This scheme introduces new GA parameters, the niche size σ_{share} and the power α. In the following, α is set to 0.1 to emphasize the niche count. Reference 16 gives a simple estimation of σ_{share} in the objective function space. For the following test cases, it will be 0.02.

Niche counts can be consistently incorporated into the fitness assignment according to rank by using them to scale individual fitness within each rank. By implementing fitness sharing in MOGAs, one can expect to evolve a uniformly distributed representation of the global tradeoff surface.

7. Comparison of SOGA and MOGA

From the techniques described above, four optimization results are shown here for demonstration purposes. Figure 3 shows the result obtained from SOGA with real number coding. The GA population is represented by dots and the Pareto front of the MO test case is indicated by a solid arc. Since the solution of the present single-objective problem is $(1/\sqrt{2}, 1/\sqrt{2})$, many dots crowd near that point.

Figure 4 shows the result of the Pareto ranking. The Pareto front is recognized better by the population although the points still cluster in the middle of the front surface. Figure 5 shows the result of MOGA that combines the Pareto ranking and the fitness sharing. The population is distributed more uniformly along the Pareto front. On the other hand, the Pareto front receded slightly. This result suggests the use of elitism to preserve the Pareto solutions.

In Fig. 6, the best-N selection [24] is incorporated further, where the best N individuals are selected for the next generation among N parents and N children. The Pareto solutions will be kept once they are formed. As a result, the population grew to the Pareto set successfully as shown in the figure.

Multidisciplinary Optimization of Wing Planform Design

Aerodynamic optimization often has to account for constraints, for example, structural strength. Such structural constraint might be derived from design optimization in the structural discipline. However, a simple sequential optimization that executes each disciplinary optimization task in sequence cannot take advantage of beneficial cross-disciplinary tradeoffs. Therefore, multidisciplinary design optimization (MDO) approach is desired. Formulation of such approach presents organizational challenges for coupling analysis codes in each discipline [19]. Furthermore, MDO requires multiobjective, system-level optimization.

The conventional system-level optimization requires system sensitivity analysis. Although the techniques for sensitivity analysis of disciplinary subproblems are well established, they require expertise in each discipline, especially in CFD. When an analysis code for a discipline is updated, the system sensitivity analysis code must also be changed. This is not cost-effective in terms of code development, since analysis codes in subproblems may be updated with more sophisticated codes frequently. A system-level optimizer is thus desired to be blind to the auxiliary information of subproblems. GAs use only objective function information, not derivatives or other auxiliary knowledge, and thus they are blind to specific problems. This feature also makes GAs attractive for solving system-level optimization.

Another advantage of GAs is their suitability to parallel processing. Since the majority of computational time will be consumed by function evaluations (aerodynamic computations), the simple *master-slave* scheme [3] can be used to improve computational efficiency of the present computation. The master process controls selection, mating, and the performance of genetic operators. The slaves simply perform function evaluations. Since GAs can be parallelized more effectively than the conventional optimization methods, they will be more efficient in parallel computing environments.

Therefore, MOGAs that can seek multiple Pareto solutions in parallel are very attractive for solving MDO problems with parallel architecture. An application of MOGA to multidisciplinary optimization (MDO) of wing planform design is presented in this section.

1. Transonic Wing Design

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 demonstration of MOGA as a system-level optimizer, the design variables for wing geometry are greatly reduced. First, aircraft sizes were assumed as span length of 94.9 *ft* and total weight of 95,000 *lb* with 156 passenger seats at cruise Mach number of 0.82. Next, as a baseline geometry, a transonic wing was taken from a previous research [25]. 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. It has a trailing edge kink at 37.7% semispan location. Its airfoil sections are supercritical and their thickness and twist angle distributions are reduced toward the tip. Then, only three parameters are chosen as design variables: sweep angle, chord length at the kink, and chord length at the tip. These parameters can produce a wide variety of aspect and taper ratios at various sweep angles. In the following, these three real numbers are regarded as strings of genetic codes of design candidates.

The objective functions and constraints are computed as follows. First, drag is evaluated, using a potential flow solver called FLO22 [26]. 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 [27]. Third, the fuel weight is calculated directly from the tank volume given by the wing geometry. Finally, the structural model is taken from [20]. 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. Furthermore, by implementing the fitness sharing technique, one can expect to evolve a uniformly distributed representation of the global tradeoff surface. The arithmetical crossover and non-uniform mutation operators [28] are used here. As the elite strategy [3], best two strings in the old generation are copied into the next generation without crossover or mutation.

The present method has been implemented on Numerical Wind Tunnel (NWT) at National Aerospace Laboratory, a parallel vector machine with 166 processing elements (PEs) at peak performance of 279 GFLOPS. A typical FLO22 calculation takes roughly 30 sec on a single PE of NWT. Computational costs for GA operators were negligible. In the following case, 50 generations are computed with a population size of 100. By using 100 PEs, the following case took roughly 5 min on NWT. Currently, CPU charge for NWT is ¥2,000 per PE per hour. Thus, the computational cost for the present run was about ¥16,000, that is, roughly $130. Since each Pareto solution is meaningful for tradeoff information, the cost for a single Pareto solution comes down to only $1.30. (In contrast, a single optimal solution would cost $130 and the rest of solutions would be discarded, if SOGA was used.)

2. Result of transonic wing planform optimization

Figure 7 shows the locus of Pareto solutions in the objective space. All three axes rearranged so that the left lower corner becomes the desired direction. Figure 8 shows the optimized wing planform shapes. The minimum-drag wing has a slightly larger taper ratio than expected since the original wing is twisted down. The shapes corresponding to the minimum wing weight and the maximum fuel weight can be distinguished easily. The wing design taken from the center of the Pareto surface has the most reasonable shape and can be considered as the best compromise between the multiple objectives here.

3. Extension to supersonic wing planform optimization

To show the applicability of the present approach to the supersonic wing planform design, the next multiobjective optimization problem considers to

1. Minimize aerodynamic drag
2. Minimize wing weight
3. Minimize aspect ratio

under the geometric constraint of the semispan-to-length ratio.

The definition of the supersonic wing geometry is further simplified. The planform parameters were assumed similar to those of the NAL scaled experimental aircraft as the semispan-to-length (lifting length of the wing) ratio of 0.45 and the root chord of 14.3 *ft* at cruise Mach number of 2.0. The flat-plate wing was assumed . Then, only four parameters are chosen as design variables: inboard and outboard sweep angles, chord length of the kink, and spanwise location of the kink. The tip chord length can be calculated from the specified parameters. These parameters can still produce a wide variety of planform shapes.

The objective functions and constraint are computed as follows. First, drag is evaluated, using the linearized theory for supersonic flows [29]. Second, wing weight is calculated, using the transonic algebraic weight equation. The formula will be upgraded for supersonic wings in future. Third, the aspect ratio is used instead of evaluating the structure, since lower aspect ratio is considered to provide stronger stiffness. The supersonic MDO code takes only a few minutes of computational time on the SGI Indy workstation because of those simplified disciplinary models.

In this research, an average rank of the population due to the Pareto ranking and fitness sharing methods was monitored for convergence as shown in Fig. 9. After several generations, all 100 individuals became Rank-1 because the best N selection was used here. The actual rank number still fluctuated afterwards due to the sharing. Since the present problem does not have a scalar objective function, it is difficult to illustrate convergence in terms of the objective function value. Thus in future, a better index for convergence is required.

Figure 10 shows the Pareto front in the objective function space and the planform shapes of the extreme Pareto solutions. The planform shape which gives the minimum drag has the largest aspect ratio. It also has the smallest wing area, and thus it gives the minimum wing weight. One of the compromised solutions is given by the center of the Pareto front. It tries to minimize the drag as well as to maximize the aspect ratio. Although the present disciplinary models are too simple to produce realistic designs, the extreme Pareto solutions are physically reasonable. This confirms the feasibility of the present approach for solving MDO problems of supersonic wing planform shapes.

Conclusion

Description of GA and its extension to multiobjective optimization are given briefly. Multiple objective GAs (MOGAs) based on the Pareto ranking and fitness sharing are introduced. MOGAs are inherently robust, suitable for parallel processing and blind to subproblems. These features make MOGAs very attractive as a system level optimizer for multidisciplinary optimization (MDO).

MOGA has been applied to MDO problems of transonic and supersonic wing planform shapes. In the transonic case, planform shapes with minimum drag, minimum weight and maximum fuel weight (tank volume) under constraints of lift and structural strength were sought by using the analysis model that consisted of potential flow, algebraic weight and wing-box structure equations. A wide variety of Pareto solutions were obtained, including a well compromised solution. The computational time was about 25 min on NAL's NWT and its computational cost was only $7 per Pareto solution.

In the supersonic case, more simplified approach was taken for the feasibility study. Preliminary results confirm that the present approach produce physically reasonable Pareto solutions. These results indicate the feasibility of the present approach for MDO of transonic and supersonic wings.

Acknowledgement

NWT computing time was provided by National Aerospace Laboratory, Japan.

References

[1] Tamaki, H., Kita H. and Kobayashi, S.: Multi-objective optimization by genetic algorithms: a review, Proceedings of 1996 IEEE International Conference on Evolutionary Computation, pp. 517-522, (1996).

[2] Vanderplaats, G. N.: *Numerical Optimization Techniques for Engineering Design: with applications*, McGraw-Hill, Inc., New York, (1984).

[3] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley Publishing Company, Inc., Reading, (1989).

[4] 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, pp. 213-218, (1989).

[5] Crispin, Y.: Aircraft conceptual optimization using simulated evolution, AIAA Paper

94-0092, (1994).
[6] 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, pp. 151-159, (1989).
[7] Gage, P. and Kroo, I.: A role for genetic algorithms in a preliminary design environment, AIAA Paper 93-3933, (1993).
[8] Gregg, R. D. and Misegades, K. P.: Transonic wing optimization using evolution theory, AIAA Paper 87-0520, (1987).
[9] Quagliarella, D. and Cioppa, A. D.: Genetic algorithms applied to the aerodynamic design of transonic airfoils, AIAA Paper 94-1896, (1994).
[10] Yamamoto, K. and Inoue, O.: Applications of genetic algorithm to aerodynamic shape optimization, AIAA Paper 85-1650-CP, A collection of technical papers, 12th AIAA Computational Fluid Dynamics Conference, CP956, San Diego, CA, pp. 43-51, (1995).
[11] Obayashi S. and Takanashi, S.: Genetic optimization of target pressure distributions for inverse design methods, *AIAA Journal*, vol. 34, no. 5, pp. 881-886, (1996).
[12] Doorly, D.: Parallel genetic algorithms for optimization in CFD, *Genetic Algorithms in Engineering and Computer Science*, Winter, G., *et al.* (ed.), John Wiley & Sons, Chichester, pp. 251-270, (1995).
[13] Périaux, J., *et al.*: Robust genetic algorithms for optimization problems in aerodynamic design, *Genetic Algorithms in Engineering and Computer Science*, Winter, G., *et al.* (ed.), John Wiley & Sons, Chichester, pp. 371-396, (1995).
[14] Poloni, C.: Hybrid GA for multi objective aerodynamic shape optimization, *Genetic Algorithms in Engineering and Computer Science*, Winter, *et al.* (ed.), John Wiley & Sons, Chichester, pp. 397-416, (1995).
[15] 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, pp. 93-100, (1985).
[16] 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, pp. 416-423, (1993).
[17] 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, pp. 82-87, (1994).
[18] Michielessen, E. and Weile D. S.: Electromagnetic system design using genetic algorithms, *Genetic Algorithms in Engineering and Computer Science*, Winter, *et al.* (ed.), John Wiley & Sons, Chichester, pp. 345-370, (1995).
[19] Sobieszczanski-Sobieski, J. and Haftka, R. T.: Multidisciplinary aerospace design optimization: survey of recent developments, AIAA Paper 96-0711, (1996).
[20] Wakayama, S. and Kroo, I.: Subsonic wing planform design using multidisciplinary optimization, *Journal of Aircraft*, Vol. 32, No. 4, pp. 746-753, July-August (1995).
[21] Anderson, M. B. and Gebert, G. A.: Using Pareto genetic algorithms for preliminary subsonic wing design, AIAA Paper 96-4023, (1996).
[22] Doorly, D. J., Peiro, J. and Oesterle, J-P.: Optimisation of aerodynamic and coupled aerodynamic-structural design using parallel genetic algorithms, AIAA Paper 96-4027, (1996).
[23] Davis, L.: *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, (1990).
[24] Tsutsui, S. and Fujimoto, Y.: Forking genetic algorithms with blocking and shrinking

modes (fGA), Proceedings of the 5th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, Inc., San Mateo, pp. 206-213, (1993).
[25] Fujii, K. and Obayashi, S.: Navier-Stokes simulations of transonic flows over a practical wing configuration, *AIAA Journal*, Vol. 25, No. 3, pp. 369-370, (1987).
[26] 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).
[27] Torenbeek, E.: *Synthesis of Subsonic Airplane Design*, Kluwer Academic Publishers, Dordrecht, (1982).
[28] Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, Second extended edition, Springer-Verlag, Berlin, (1994).
[29] Carlson, H. W. and Middleton, W. D., "A Numerical Method for the Design of Camber surfaces of supersonic Wings with Arbitrary Planforms," NASA TN D-2341, June 1964.

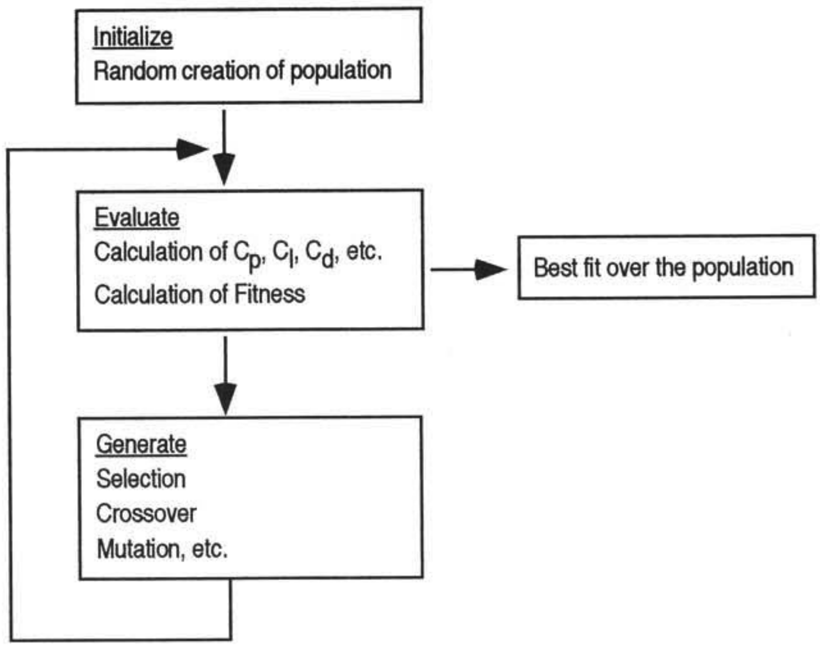

Fig. 1 Flowchart of GA

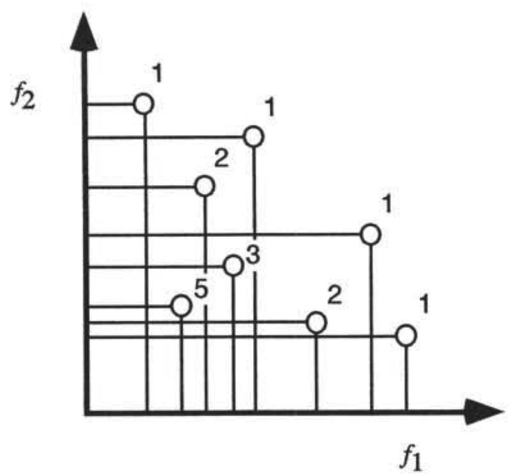

Fig.2 Pareto ranking

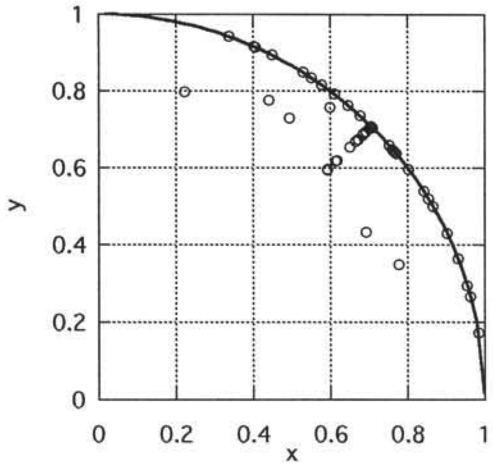

Fig. 3 Result of SOGA

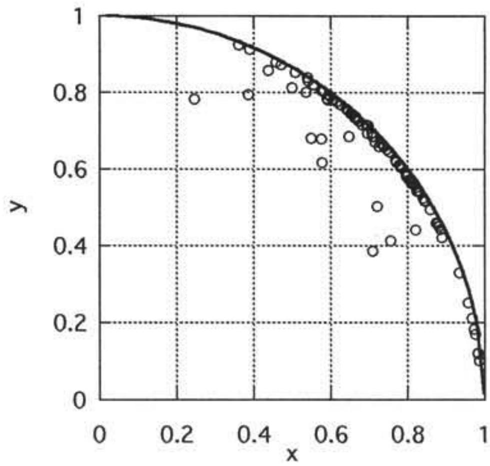

Fig. 4 Result of MOGA (Pareto ranking)

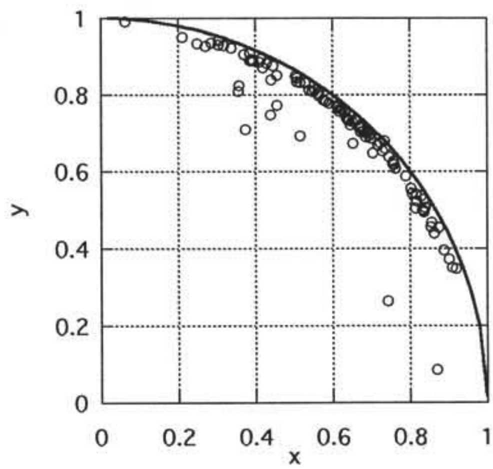

Fig. 5 Result of MOGA (Plus fitness sharing)

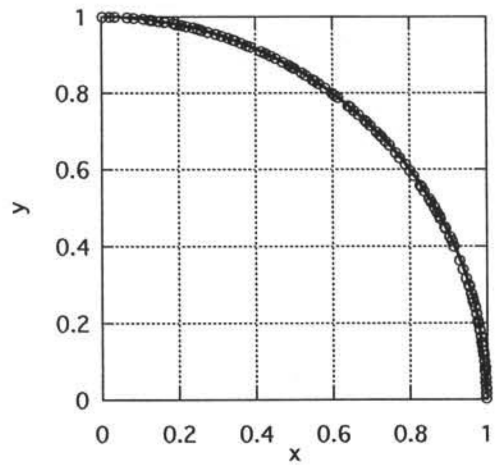

Fig. 6 Result of MOGA (Plus best-N selection)

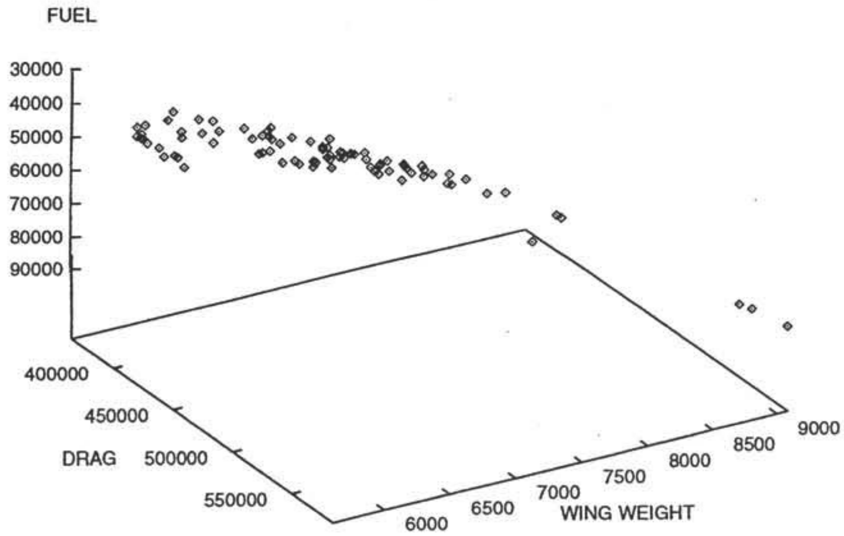

Fig. 7 Locus of Pareto solutions in the objective function space for the transonic case

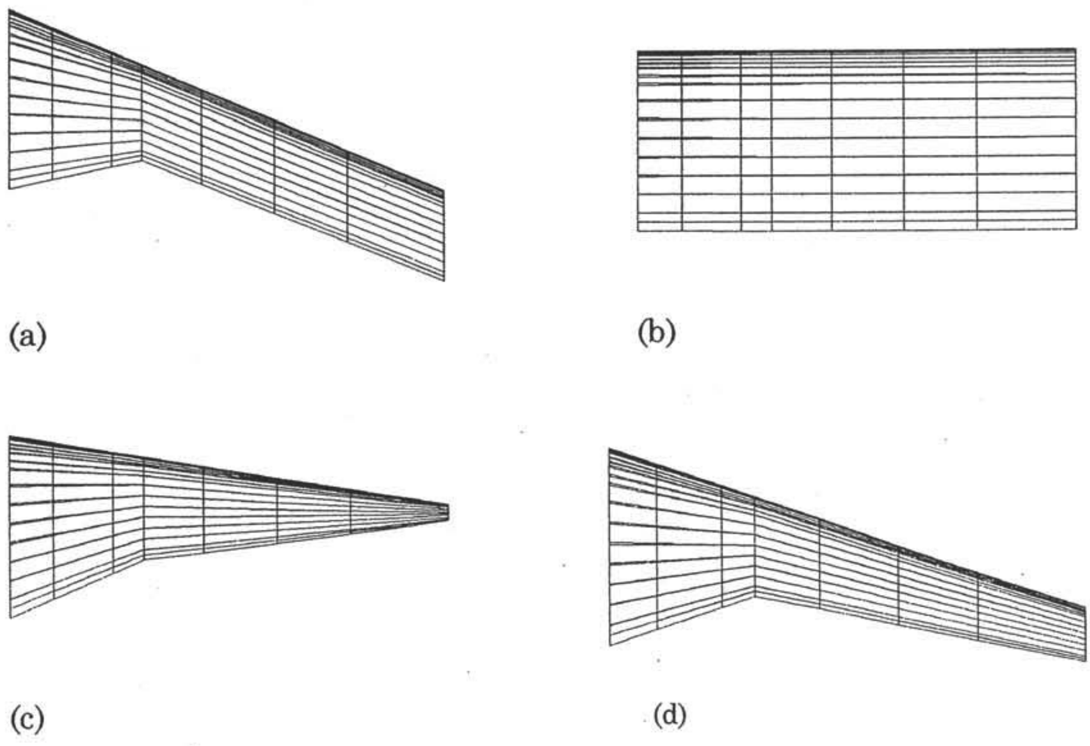

Fig. 8 Various Pareto solutions for the transonic case with a) the lowest drag, b) the maximum fuel weight, c) the minimum wing weight, and d) center of Pareto surface

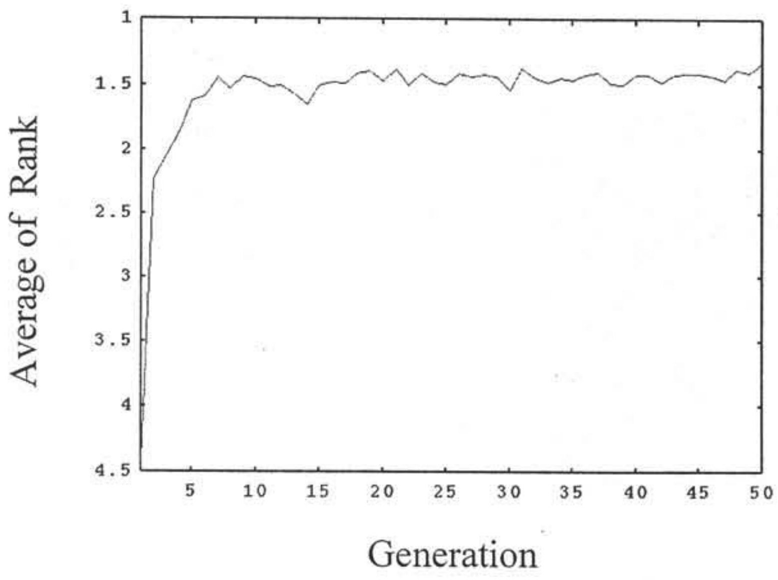

Fig. 9 Convergence history of the supersonic case

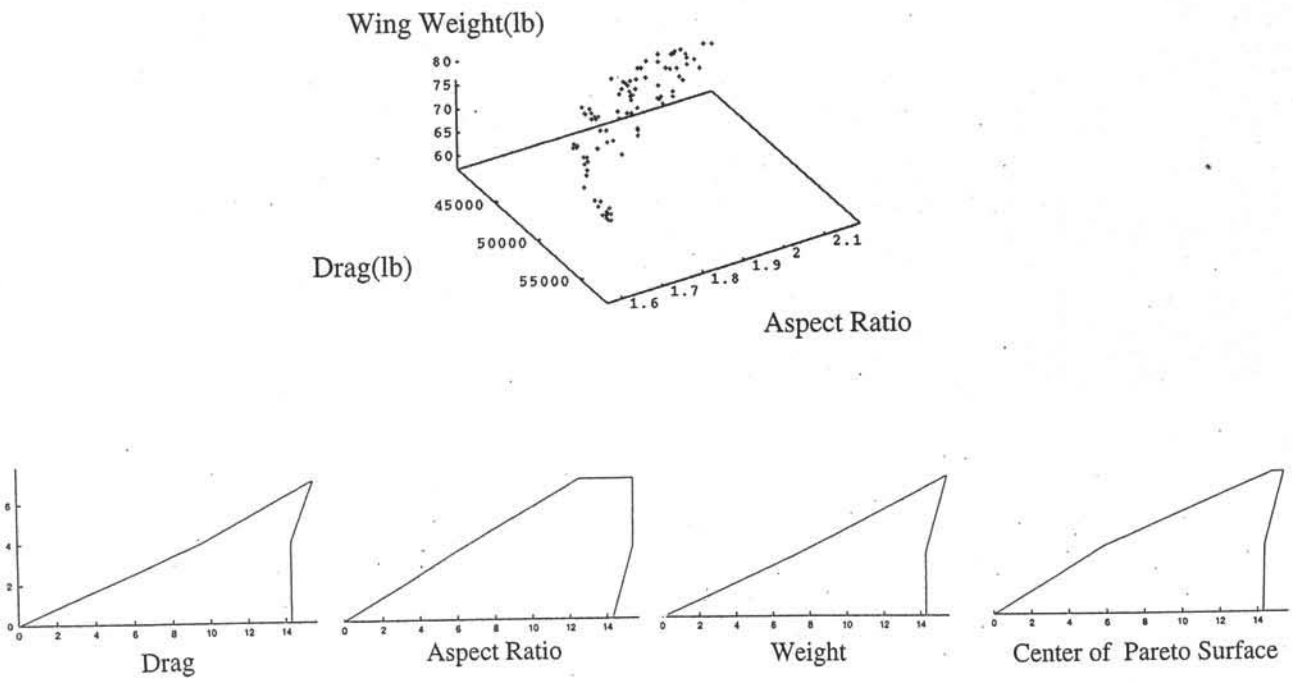

Fig. 10 Pareto front and various Pareto solutions for the supersonic case