

Multi-Objective Design Exploration for Aerodynamic Configurations

Shigeru Obayashi*,
Tohoku University, Sendai, 980-8577, Japan

Shinkyu Jeong†
Tohoku University, Sendai, 980-8577, Japan

and

Kazuhisa Chiba‡
Japan Aerospace Exploration Agency, Tokyo, 181-0015, Japan

A new approach, Multi-Objective Design Exploration (MODE), is presented to address Multidisciplinary Design Optimization problems. MODE reveals the structure of the design space from the trade-off information and visualizes it as a panorama for Decision Maker. The present form of MODE consists of Kriging Model, Adaptive Range Multi Objective Genetic Algorithms, Analysis of Variance and Self-Organizing Map. The main emphasis of this approach is visual data mining. Two data mining examples using high fidelity simulation codes are presented: four-objective aerodynamic optimization for the fly-back booster and Multidisciplinary Design Optimization problem for a regional-jet wing. The first example confirms that two different data mining techniques produce consistent results. The second example illustrates the importance of the present approach because design knowledge can produce a better design even from the brief exploration of the design space.

I. Introduction

This paper discusses a new approach for Multidisciplinary Design Optimization (MDO). MDO has been a rapidly growing area of research.¹⁻³ Thanks to these pioneering works, researchers in Computational Fluid Dynamics (CFD) are getting interested in MDO research as well. MDO research is still expanding because high fidelity CFD codes are becoming available with the aid of increasing computer power.

A typical MDO problem involves competing objectives, for example in the aircraft design, minimization of aerodynamic drag, minimization of structural weight, etc. While single objective problems may have a unique optimal solution, multi-objective problems (MOPs) have a set of compromised solutions, largely known as the trade-off surface, *Pareto-optimal* solutions or *non-dominated* solutions.⁴ These solutions are optimal in the sense that no other solutions in the search space are superior to them when all objectives are considered (Fig. 1).

Traditional optimization methods such as the gradient-based methods^{5,6} are single objective optimization methods that optimize only one objective. These methods usually start with a single baseline design and use local gradient information of the objective function with respect to changes in the design variables to calculate a search direction. When these methods are applied to a MOP, the problem is transformed into a single objective optimization problem by combining multiple objectives into a single objective typically using a weighted sum method. For example, to minimize competing functions f_1 and f_2 , these objective functions are combined into a scalar function F as

$$F = w_1 \cdot f_1 + w_2 \cdot f_2 \quad (1)$$

* Professor, Institute of Fluid Science, 2-1-1 Katahira, Sendai, Japan, 980-8577, Associate Fellow AIAA

† Research Associate, Institute of Fluid Science, 2-1-1 Katahira, Sendai, Japan, 980-8577, Member AIAA

‡ Research Associate, 7-44-1 Jindaiji-Higashi, Chofu, Japan, 182-8522, Member AIAA

This approach, however, can find only one of the Pareto-optimal solutions corresponding to each set of the weights w_1 and w_2 . Therefore, one must run many optimizations by trial and error adjusting the weights to get Pareto-optimal solutions uniformly over the potential Pareto-front. This is considerably time consuming in terms of human time. What is more, there is no guarantee that uniform Pareto-optimal solutions can be obtained. For example, when this approach is applied to a MOP that has concave trade-off surface, it converges to two extreme optimums without showing any trade-off information between the objectives (Fig. 2). To overcome these difficulties, Normal-Boundary Intersection Method⁷ and Aspiration Level Method⁸ were developed.

An alternative approach to solve MOP is to find as many Pareto-optimal solutions as possible to reveal trade-off information among different objectives. Once such solutions are obtained, Decision Maker (DM) will be able to choose a final design with further considerations. Evolutionary Algorithms (EAs, for example, see Refs. 9 and 10) are particularly suited for this purpose.

Evolutionary Algorithm is a generic name for population-based optimization methods, such as Genetic Algorithms (GAs), Evolutionary Strategies (ESs), Genetic Programming (GP), *etc.*¹¹ EAs simulate the mechanism of natural evolution, where a biological population evolves over generations to adapt to an environment. Fitness, the individual, and genes in the evolutionary theory correspond to the objective function, design candidate, and design variables in design optimization problems, respectively.

EAs have been extended successfully to solve MO problems.¹² EAs use a population to seek optimal solutions in parallel. This feature can be extended to seek Pareto solutions in parallel without specifying weights between the objective functions. Because of this characteristic, EAs can find Pareto solutions for various problems having convex, concave and discontinuous Pareto front. The resultant Pareto solutions represent global trade-offs. In addition, EAs have other advantages such as robustness and suitability for parallel computing. Due to these advantages, EAs have been applied to MOPs very actively (EMO proceedings). EAs have been also applied to single objective and multi-objective aerospace design optimization problems.¹³⁻¹⁹

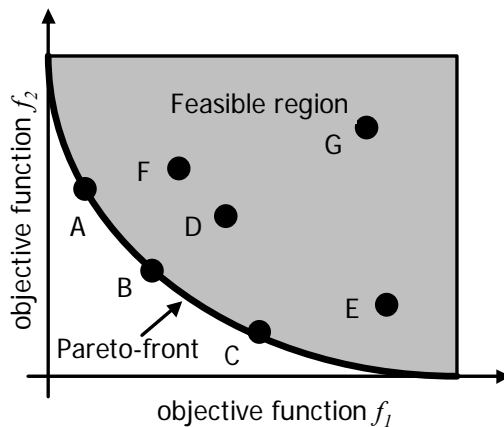


Figure 1 The concept of Pareto-optimality

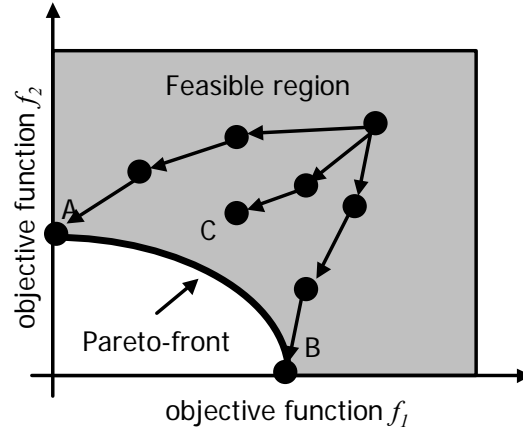


Figure 2 Weighted-sum method applied to a MOP having a convex Pareto-front

This approach of finding many Pareto solutions works fine as it is, however, only when the number of objectives remains small (usually two, three at most, as shown in Fig. 3). To reveal trade-off information from the resultant Pareto front for real-world problems with many objectives, visualization of the Pareto front becomes an issue. Several techniques have been considered, such as parallel coordinates,²⁰ box plot,²¹ and Self-Organizing Map (SOM).²² The importance of visualization of design space is also discussed in Ref. 23. Because such visualization is a tool for data mining, data mining is found very important in this approach.

To support data mining activities, response surfaces are found versatile. Once the surface is constructed, it can be used for statistical analysis, for example, analysis of variance (ANOVA).²⁴ ANOVA shows the effect of each design variables on objective functions quantitatively while SOM shows the information qualitatively. When the response surface method (RSM) is introduced for data mining as post-process of optimization, it can be applied to pre-process of optimization as a surrogate model,²⁵⁻²⁷ too. Pre-process has been an important aspect of introduction of surrogate models because it would reduce the computational expense greatly, while it would produce rich non-dominated

solutions efficiently. In this paper, surrogate models are introduced for both pre- and post-processes. However, it should be noted that RSM is needed for post-process primarily. EAs may be applied from the beginning in parallel to building the surrogate model. If function evaluations are very cheap, EAs may also be applied directly.

As a result, the new approach for MDO named as Multi-Objective Design Exploration (MODE) can be summarized as a flowchart shown in Fig. 4. MODE is not intended to give an *optimal* solution. MODE reveals the structure of the design space from the trade-off information and visualizes it as a panorama for DM. DM will know the reason for trade-offs from non-dominated designs, instead of receiving an *optimal* design without trade-off information.

The rest of the paper will explain the components of MODE used in our group, although the concept of MODE can be coupled with other RSM and optimization algorithms. Examples of data mining^{17,24,28} will be given briefly.

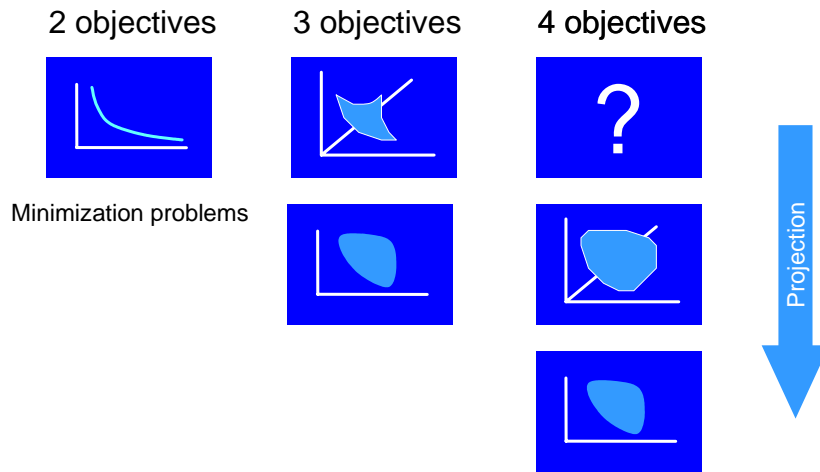


Figure 3 Visualization of Pareto front

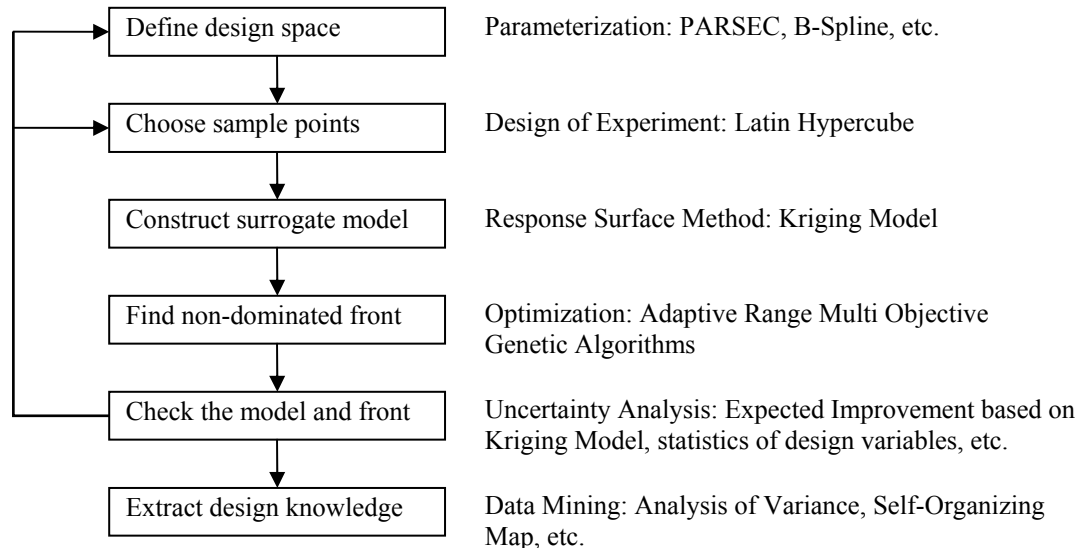


Figure 4 Flowchart of Multi-Objective Design Exploration (MODE) with component algorithms

II. Surrogate Model

A. Kriging Model

The present Kriging model expresses the unknown function $y(\mathbf{x})$ as

$$y(\mathbf{x}) = \mu + Z(\mathbf{x}) \quad (2)$$

where \mathbf{x} is an m -dimensional vector (m design variables), μ is a constant global model, and $Z(\mathbf{x})$ represents a local deviation from the global model. In the model, the local deviation at an unknown point is expressed using stochastic processes. The sample points are interpolated with the Gaussian correlation function to estimate the distribution of the function value at the unknown point. The correlation between $Z(\mathbf{x}^i)$ and $Z(\mathbf{x}^j)$ is strongly related to the distance between the two corresponding points, \mathbf{x}^i and \mathbf{x}^j . In the Kriging model, a special weighted distance is used instead of the Euclidean distance because the Euclidean distance weighs all design variables equally. The distance function between the point at \mathbf{x}^i and \mathbf{x}^j is expressed as

$$d(\mathbf{x}^i, \mathbf{x}^j) = \sum_{k=1}^m \theta_k |x_k^i - x_k^j|^2 \quad (3)$$

where θ_k ($0 \leq \theta_k < \infty$) is the k th element of the parameter $\boldsymbol{\theta}$. The correlation between the point \mathbf{x}^i and \mathbf{x}^j is defined as

$$\text{Corr}[Z(\mathbf{x}^i), Z(\mathbf{x}^j)] = \exp[-d(\mathbf{x}^i, \mathbf{x}^j)] \quad (4)$$

The Kriging predictor^{27,29} is

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \mathbf{r}'\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}) \quad (5)$$

Where $\hat{\mu}$ is the estimated value of μ , \mathbf{R} denotes the $n \times n$ matrix whose (i, j) entry is $\text{Corr}[Z(\mathbf{x}_i), Z(\mathbf{x}_j)]$. \mathbf{r} is vector whose i th element is

$$r_i(\mathbf{x}) \equiv \text{Corr}[Z(\mathbf{x}), Z(\mathbf{x}^i)] \quad (6)$$

and $\mathbf{y} = [y(\mathbf{x}^1), \dots, y(\mathbf{x}^n)]$.

The unknown parameter, $\boldsymbol{\theta}$, for the Kriging model can be estimated by maximizing the following likelihood function:

$$\begin{aligned} \ln(\hat{\mu}, \hat{\sigma}^2, \boldsymbol{\theta}) = & -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\hat{\sigma}^2) - \frac{1}{2} \ln(|\mathbf{R}|) \\ & - \frac{1}{2\hat{\sigma}^2} (\mathbf{y} - \mathbf{1}\hat{\mu})' \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu}) \end{aligned} \quad (7)$$

where $\mathbf{1}$ denotes an m -dimensional unit vector.

Maximization of the likelihood function is an m -dimensional unconstrained non-linear optimization problem. In this paper, the alternative method³⁰ is adopted to solve this problem.

For a given $\boldsymbol{\theta}$, $\hat{\mu}$ and $\hat{\sigma}^2$ can be defined as

$$\hat{\mu} = \frac{\mathbf{1}'\mathbf{R}^{-1}\mathbf{y}}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}} \quad (8)$$

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})' \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu})}{n} \quad (9)$$

The accuracy of the estimated value on the Kriging model largely depends on the distance from the sample points. Intuitively speaking, the closer point \mathbf{x} is to the sample points, the more accurate $\hat{y}(\mathbf{x})$ is. This intuition is expressed in the mean squared error of the predictor.

$$s^2(\mathbf{x}) = \hat{\sigma}^2 \left[1 - \mathbf{r}'\mathbf{R}^{-1}\mathbf{r} + \frac{(\mathbf{1} - \mathbf{1}\mathbf{R}^{-1}\mathbf{r})^2}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}} \right] \quad (10)$$

$s^2(\mathbf{x})$ is the mean squared error at point \mathbf{x} , indicating the uncertainty of the estimated value.

B. Exploration of Global Optimum and Treatment of Constraints on the Kriging model

Once the approximation model is constructed, the optimum point can be explored using an arbitrary optimizer. However, there is a possibility of missing the global optimum because the estimated value includes uncertainty.

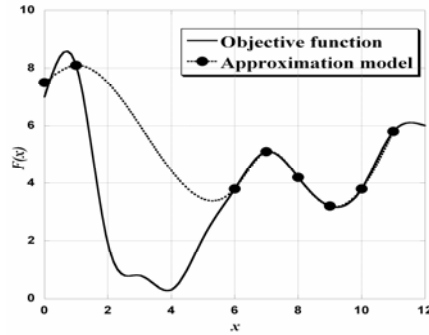


Figure 5 The objective function and the approximation model

In Fig. 5, the solid line is for the real shape of the objective function and the dotted line is for the approximation model. The minimum point on the approximation model is located near $x=9$, whereas, the real global minimum of the objective function is situated near $x=4$. Exploration of global minimum using the approximation model is apt to result in the local minimum. For a robust search of the global optimum on the approximation model, the uncertainty information is very useful.

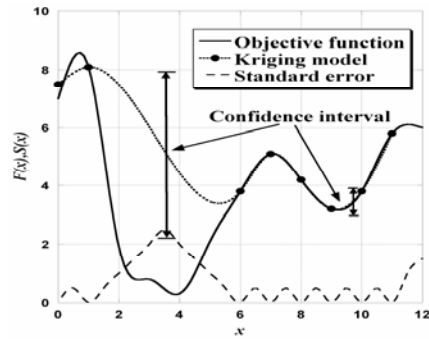


Figure 6 The estimated value and the standard error of the Kriging model

Figure 6 shows the estimated value and the standard error (uncertainty) of the Kriging model. Around $x=9.5$, the standard error of the Kriging model is very small because there are many sample points around this point. Thus, the confidence interval is very short as shown in Fig. 6. On the other hand, the standard error around $x=3.5$ is very large due to the lack of sample points around there. Thus, the confidence interval at this point is very wide. The lower bound of this interval is smaller than current minimum on the Kriging model. Thus, it can be said that this point has some probability of being the global minimum.

The probability of being the global optimum concept can be expressed by the criterion of ‘expected improvement (EI)’³¹. In case of a minimization problem, the EI is expressed as follows:

$$I(x) = \begin{cases} [f_{\min} - y(x)] & \text{if } y(x) < f_{\min} \\ 0 & \text{otherwise} \end{cases} = \max(f_{\min} - y, 0) \quad (11)$$

$$E(I) = \int_{-\infty}^{f_{\min}} (f_{\min} - y) \phi(y) dy \quad (12)$$

where ϕ is the probability density function representing uncertainty about y . By selecting the maximum EI point as additional sample points of the Kriging model iteratively, the robust exploration of the global optimum is possible.

Then, if there are constraints as follows,

$$a_i \leq c_i(x) \leq b_i \quad i = 1, \dots, k \quad (13)$$

EI subject to constraints is expressed as follows:

$$\begin{aligned} E_{y, c_1, c_2, \dots, c_k}(I_c) \\ = E_y(\max(f_{\min} - y, 0) \int_{a_1}^{b_1} \dots \int_{a_k}^{b_k} f_{c_1, \dots, c_k|y} d_{c_1} \dots d_{c_k}) \end{aligned} \quad (14)$$

In order to evaluate Eq. (14), the multivariate normal distribution $\phi(y, c_1, \dots, c_k)$, which is very complicated, should be specified. Thus, in this paper, we assume that y, c_1, c_2, \dots, c_k are statistically independent in order to simplify Eq. (14).

The modified Eq. (14) is as follows

$$\begin{aligned} E_{y, c_1, \dots, c_k}(I_c) \\ = E_y(I) \cdot \prod_{i=1, \dots, k} (\Phi_{c_i}(b_i) - \Phi_{c_i}(a_i)) \\ = E_y(I) \cdot P(a_1 \leq c_1(x) \leq b_1) \dots P(a_k \leq c_k(x) \leq b_k) \end{aligned} \quad (15)$$

In order to calculate this value, the Kriging model is constructed for the objective function and all constraint functions separately. On the Kriging model of objective function, EI is calculated, and on the Kriging models of constraints, the probability to satisfying each constraint is calculated. Based on these values, the next additional point for balanced local and global search is selected, while satisfying the constraints.

III. Adaptive Range Multi-Objective Genetic Algorithms

Pareto solutions and Pareto front are exact solutions by definition. Because it is difficult to show that numerical solutions are exact, numerical solutions and the corresponding front are usually called as non-dominated solutions and non-dominated front, respectively. They are non-dominated among the solutions generated by the computation (Fig. 7).

Except for the introduction of range adaptation operator, the present ARMOGAs' operators¹⁵ are the same as the MOEAs.⁹ Therefore, each genetic operator of the MOEAs adopted here is firstly explained. Then the unique procedure of ARMOGAs is described in this chapter.

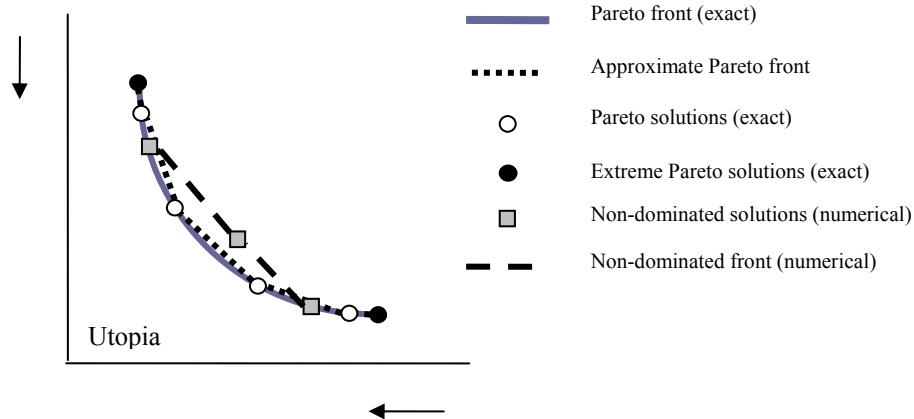


Figure 7 Definition of Pareto solutions and non-dominated solutions

A. Algorithm of Multi-Objective Evolutionary Algorithms

1. Binary and Floating-Point Representation

As GAs originally simulated natural evolution, binary numbers were often used to represent design parameter values. However, for real function optimizations, such as aerodynamic optimization problems, it is more straightforward to use real numbers. Thus, the floating-point representation is adopted here.

2. Coding and Decoding

GAs require both phenotype and genotype design variables. The phenotype design variable represents the actual design variables, such as length, angle, shape, *etc.* On the other hand, the genotype design variable is a binary number (Binary GAs) or a real number in $[0,1]$ (Real-coded GAs). The operators of many GAs require genotype representation of design parameters. Therefore, actual design variables (phenotype representation) must be converted to the genotype representation. The conversion from phenotype to genotype is called “coding,” and conversely, the conversion from genotype to phenotype is “encoding.” For real-parameter design problems, such as aerodynamic optimizations, it is not favorable to use binary representation. One reason for this is that phenotype design space is not continuous by binary representation.

For the present floating-point representation, i -th design parameter p_i is coded to genotype value r_i , which is normalized in $[0,1]$:

$$r_i = \frac{p_i - p_{i,min}}{p_{i,max} - p_{i,min}} \quad (16)$$

3. Initial Population

The results of GAs can be affected by the initial population if the number of individuals per generation is small. It would be better to generate initial individuals in a wide range of design spaces. Here, the initial population is generated randomly.

4. Evaluation

As GAs use only objective-function values for optimization, no modification of evaluation tools is required. In addition, it is easy to apply Master-Slave type parallelization systems to conserve computational resources because GAs do not have to compute design candidates sequentially, unlike gradient-based method.

5. Selection

GAs choose superior individuals as parents to generate new design candidates. Therefore, selection has a large influence on search performance of GAs. For single-objective optimizations, as the aim is to obtain the best solution, selection is based on the fitness value given by the objective-function value. However, Pareto-optimal solutions must be obtained for MO optimization. To obtain Pareto solutions effectively, each individual is assigned a rank based on the Pareto ranking method and fitness sharing. In the present MOEAs, Fleming and Fonseca’s Pareto-ranking method¹² is adopted. Each individual is assigned a rank according to the number of individuals dominating it, as shown in Fig. 8. The fitness value (F_i) of individual i is assigned based on the following equation:

$$F_i = N - \sum_{k=1}^{R_i-1} \mu(k) - 0.5(\mu(R_i) - 1) \quad (17)$$

where N is the number of solutions, and $\mu(R_i)$ is the number of solutions in rank R_i . Thereafter, the standard sharing approach is adopted to prevent choosing similar solutions as parents and to maintain diversity of the population. The assigned fitness values are divided by the niche count:

$$F'_i = F_i / nc_i \quad (18)$$

Here, niche count nc_i is calculated by summing the sharing function values:

$$nc_i = \sum_{j=1}^N sh(d_{ij}) \quad (19)$$

$$sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{share}} \right)^\alpha & d_{ij} < \sigma_{share} \\ 0 & \text{others} \end{cases} \quad (20)$$

$$d_{ij} = \sqrt{\sum_{k=1}^M \left(\frac{f_k^i - f_k^j}{u_k - l_k} \right)^2} \quad (21)$$

where u_k is the maximum objective-function value of k at the present generation, l_k is the minimum objective-function value of k at the present generation, and α is the sharing function parameter. If the distance between individuals i and j is lower than σ_{share} , then niche count increases to reduce the fitness of the solution. The normalized niching parameter σ_{share} is proposed as follows:

$$(1 + \sigma_{share})^M - 1 = N(\sigma_{share})^M \quad (22)$$

where N is the size of the population and M is the number of objective functions.

After shared fitness values are determined for all individuals, the stochastic universal selection (SUS) is applied to select better solutions for producing a new generation. Unlike roulette wheel selection method, only one random number is chosen for the whole selection process for SUS. As many different solutions should be chosen to maintain the diversity, a set of N equi-spaced numbers is created.

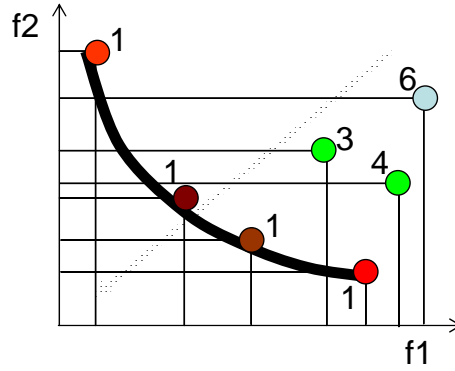


Figure 8 Pareto ranking method (Rank 1 means non-dominated solutions)

6. Crossover

Crossover is an operator that interchanges the genotype parameters of selected parents and produces two different design candidates. Probability of crossovers and crossover method markedly affect the search performance of GAs.

For the binary representation, crossover interchanges the bit strings of selected parents. However, many crossover methods have been proposed for real-parameter GAs. Simulated binary crossover (SBX) operator⁹ creates offspring based on the distance between the parents. If the two parents are closely related to each other, SBX is likely to generate new offspring near the parents. On the other hand, if the two parents are more distantly related, it is possible for solutions to be created away from the parents. This operator is described as follows:

$$Child1 = 0.5 [(1 + \beta_q) \cdot Parent1 + (1 - \beta_q) \cdot Parent2] \quad (23a)$$

$$Child2 = 0.5 [(1 - \beta_q) \cdot Parent1 + (1 + \beta_q) \cdot Parent2] \quad (23b)$$

$$\beta_q = \begin{cases} (2 \cdot ran1)^{1/(\eta_c + 1)} \\ \left(\frac{1}{2 \cdot (1 - ran1)} \right)^{1/(\eta_c + 1)} \end{cases} \quad (23c)$$

7. Mutation

Mutation maintains diversity and expands the search space by changing the design parameters. If the mutation rate is high, a GA search is close to a random search and results in slow convergence. Therefore, an adequate value is required for the mutation rate. For binary representation, mutation is performed to reverse the bit strings. It is not as simple for real-coded GAs as for binary GAs. This is realised by adding disturbances to the design parameters.

Polynomial mutation, which is similar to the SBX operator described in previous section, has been proposed⁹:

$$Child_{mutation} = Child_{crossover} + (x_{max} - x_{min}) \cdot \delta \quad (24)$$

where δ is calculated from the polynomial probability distribution:

$$\delta = \begin{cases} (2 \cdot ran2)^{1/(\eta_m + 1)} - 1 \\ 1 - [2 \cdot (1 - ran2)^{1/(\eta_m + 1)}] \end{cases} \quad (25)$$

where $ran2$ is a uniform random number in $[0,1]$. A value of η_m determines the perturbation size of mutation.

8. Archiving

To obtain Pareto solutions efficiently, it would be better to include past excellent solutions as current solutions. In the present MOEAs, two archiving techniques are combined. The first is the *Best-xN* technique, which keeps the latest better solutions and parent generation of $(x-1)N$ size and uses these solutions for the selection process. The second is the standard archiving technique, which is comprised of all previous solutions to prevent the loss of previous excellent solutions. These two methods are combined in the present MOEAs as shown in Fig. 9. The procedure is as follows:

1. Fitness values based on the fitness assignment operators are assigned to the present population and the *Best-xN* group. Here, x is set to 2.
2. According to the fitness value, the top N individuals are chosen for the next step. In addition, the top $(x-1)N$ individuals are preserved as the *Best-xN* group.
3. Fitness values are assigned to chosen N individuals.
4. SUS is used to select the parents. Then, crossover and mutation are applied to generate new individuals.
5. Several individuals from the *Best-xN* group are replaced by the same number of individuals from the archives.

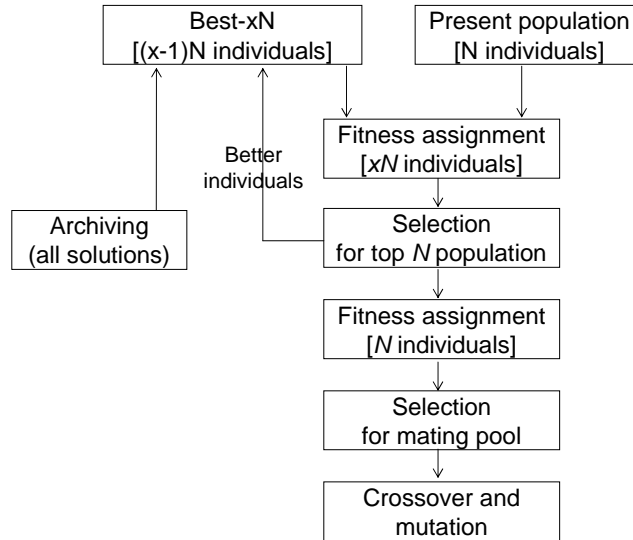


Figure 9 Archiving procedure used in the present MOEAs

9. Constraint-Handling Technique

In many real-world problems, it is common to have several constraints. Many constraint-handling techniques have been proposed, however, it is not easy for GAs to solve constrained-problems compared to gradient-based methods. A popular and easy constraint-handling strategy is the penalty function approach in which a penalty value is added to the objective-function value if the design violates the constraint. Although several penalty functions have been proposed, it is difficult to choose appropriate penalty values *a priori*.

In the present MOEAs, an extended Pareto ranking method based on constraint-dominance is used. Constraint-dominance is defined as follows⁹:

A solution \mathbf{x}_i is said to 'constrain-dominate' a solution \mathbf{x}_j , if any of the following conditions are true:

1. \mathbf{x}_i is feasible and \mathbf{x}_j is not.

2. \mathbf{x}_i and \mathbf{x}_j are both infeasible, but \mathbf{x}_i has a smaller constraint violation.
3. \mathbf{x}_i and \mathbf{x}_j are feasible and \mathbf{x}_i dominates \mathbf{x}_j in the usual sense.

Figure 10 shows the example of a Pareto ranking method based on constraint-dominance for the two-objective minimization problem with one constraint. Based on this approach, it would be easy to generate new offspring that satisfy the constraints because feasible solutions are likely to be chosen as the parents. However, it is possible for good solutions to lie close to the edge of the feasible and infeasible region in many industrial problems. Therefore, an adequate tolerance of the constraint (c_{tol}) should be introduced to the constraint violation:

$$G - c_{tol} \leq 0 \quad (26)$$

where G is an original constraint less than zero. As the tolerance c_{tol} is introduced, solutions having smaller violation than c_{tol} are assumed to be feasible for constraint-dominance. This enables EAs to search for solutions near the boundary between feasible and infeasible solutions.

To consider the aerodynamic optimization using time-consuming CFD, it is unfavorable to generate many violated candidates. If it is possible to determine that the solution violates constraints before CFD computation, such as geometrical constraints (length, angle, etc.), it is possible to prevent generating such solutions, as it would be a waste of computation time in CFD. In the case of aerodynamic optimization, it would be better to take account of the above problem.

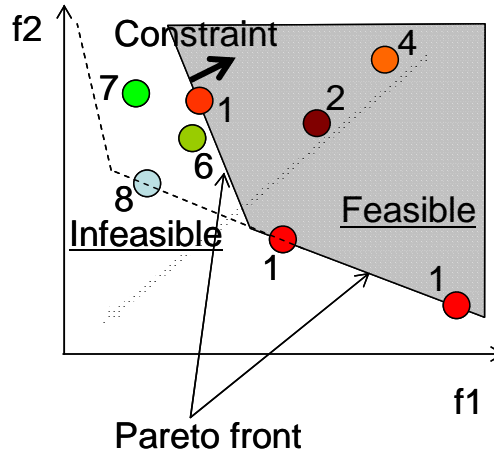


Figure 10 Example of constrain-dominance.

B. Algorithm of Adaptive Range Multi-Objective Genetic Algorithms

To reduce the large computational burden, the reduction of the total number of evaluations is needed. On the other hand, a large string length is necessary for real parameter problems. Oyama developed real-coded ARGAs and applied them to the transonic wing optimization.¹⁴ According to the encoding system based on normal distribution (Fig. 11) built by population statistics consisting of better designs computed before, ARGAs can find a good optimal design efficiently.

The basis of ARMOGAs is the same as ARGAs, but a straightforward extension may cause a problem in the diversity of the population. Therefore, ARMOGAs have been developed based on ARGAs to deal with multiple Pareto solutions for the multi-objective optimization. In addition, archiving and constraint-handling techniques are considered to select better solutions to decide new search range.

This section describes genetic operators of ARMOGAs. ARMOGAs differ from MOEAs described above with regard to the application of range adaptation. Therefore, before starting range adaptation, the MOEAs and ARMOGAs in the present study are identical. A flowchart of ARMOGAs is shown in Fig. 12. The range adaptation starts at M_{sa} generation and is carried out every M_{ra} generations. The new decision space is determined based on the statistics of selected better solutions, and then the new population is generated in the new decision space. Thereafter, all the genetic operators are applied to the new design space.

ARMOGAs are able to find Pareto solutions more efficiently than conventional MOEAs because of the concentrated search of the promising design space out of the large, initial design space. ARMOGAs can adapt their

search region as shown in Fig. 13. In contrast, the search region of conventional EAs remains unchanged. The encoding system is based on the normal distribution with the plateau region as shown in Fig. 14. The selected designs locate in the plateau region, and the normal distribution region is determined based on the population statistics to better preserve the diversity of solution candidates. Re-initialization helps to maintain the population diversity.

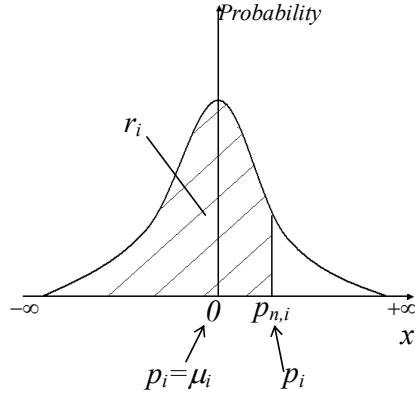


Figure 11 Normal distribution for encoding in real-coded ARGAs

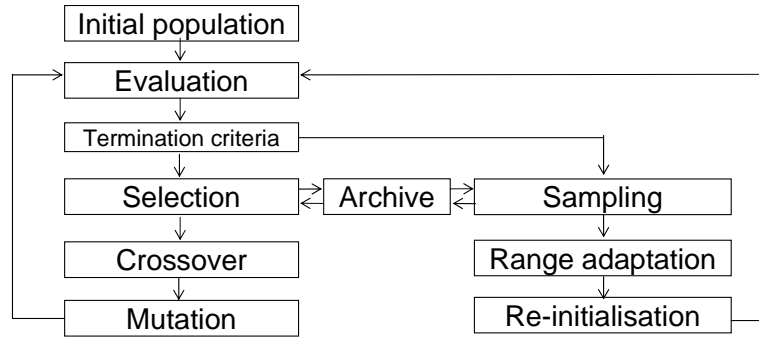


Figure 12 Flowchart of ARMOGAs

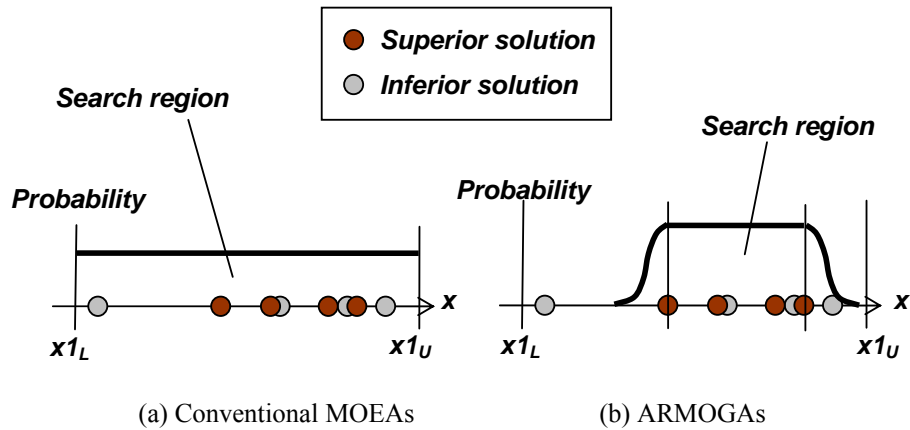


Figure 13 Sketch of search region

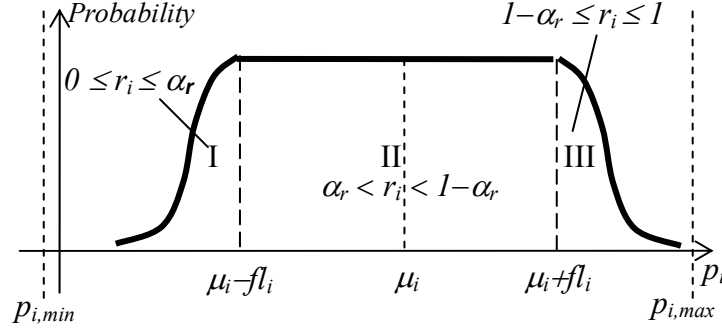


Figure 14 Sketch of probability distribution of phenotype design variable p_i in ARMOGAs

1. Sampling for Range Adaptation

Range adaptation needs to select superior solutions to determine the new design space based on the statistics. The solutions, which have higher fitness values based on Pareto ranking method, are selected to determine the reasonable search range. It would be better to select many solutions to prevent the creation of new search regions that do not include the global optimum. On the other hand, many solutions for range adaptation generally interfere with the decrease in size of the search space. The solutions are selected at random according to their fitness given by the following solution sets:

1. PR_{non} % non-dominated solutions from all solutions. ($PR_{non}=100$)
2. PR_{arc} % solutions from the archive. ($PR_{arc}=0$)
3. PR_{prs} % solutions from the latest generation. ($PR_{prs}=0$)
4. PR_{vio} % solutions that violate the constraint. ($PR_{vio}=1$, at least one design)

Solution set 4 is introduced to search near the boundary between feasible and infeasible solutions, as the global optimum for constraint problems is often located there. According to the amount of violation, violated designs are sampled. The probabilities in bracket are used in this optimization. In this case, only non-dominated solutions with several infeasible designs are selected to determine new design range.

2. Range Adaptation

In the ARMOGAs, the search region is changed according to the population statistics of the average and the standard deviation. The range adaptation adopts the Normal distribution to search global solutions efficiently. Figure 11 shows the Normal distribution used for encoding in the real-coded ARGAs. The real value of the i -th design variable p_i is encoded to a real number r_i defined in $(0,1)$ such that r_i is equal to the integrations of the normal distribution from $-\infty$ to $p_{n,i}$:

$$r_i = \int_{-\infty}^{p_{n,i}} N(0,1)(z)dz \quad (27a)$$

$$p_{n,i} = \frac{p_i - \mu_i}{\sigma_i} \quad (27b)$$

where μ_i is the average of the i -th design variable, and σ_i is the standard deviation of the i -th design variable.

The basic idea of encoding system in ARMOGAs is the same as for real-coded ARGAs, but a straightforward extension is not suitable in diversity of the population. To better preserve the diversity of solution candidates, the Normal distribution for encoding has to be changed.

Figure 14 shows the search range with the distribution of probability. The search region is partitioned into three parts, I, II, and III. Regions I and III make use of the same encoding method as ARGAs. The real value of the i -th design variable p_i is encoded to a real number r_i defined in $(0,1)$. In contrast, region II adopts the conventional real-number encoding method. The plateau region (region II) is defined by the upper and lower design variables of chosen solutions. Then, the normal distribution is considered at both sides of the plateau determined by the average (μ_i) and the standard deviation (σ_i). This encoding system is controlled by the parameters α_r and fl_i , where $\alpha_r (<0.5)$

is the population ratio at region I and fl_i is half the length of the plateau at region II. The encoding is conducted at each region described below:

Region I ($p_i \leq \mu_i - fl_i$, $0 \leq r_i \leq \alpha_r$):

$$r_i = \alpha_r \cdot r'_i \quad (28a)$$

$$r'_i = \int_{-\infty}^{p_{n,i}} N(0,1)(z) dz \quad (28b)$$

$$p_{n,i} = \frac{p_i - (\mu_i - fl_i)}{2\sigma_i} \quad (28c)$$

Region II ($\mu_i - fl_i < p_i < \mu_i + fl_i$, $\alpha_r < r_i < 1 - \alpha_r$):

$$r_i = (1 - 2\alpha_r) \cdot r'_i + \alpha_r \quad (28d)$$

$$r'_i = \frac{p_i - (\mu_i - fl_i)}{2fl_i} \quad (28e)$$

Region III ($\mu_i + fl_i \leq p_i$, $1 - \alpha_r \leq r_i \leq 1$):

$$r_i = \alpha_r \cdot r'_i + (1 - \alpha_r) \quad (28f)$$

$$r'_i = \int_{-\infty}^{p_{n,i}} N(0,1)(z) dz \quad (28g)$$

$$p_{n,i} = \frac{p_i - (\mu_i + fl_i)}{2\sigma_i} \quad (28h)$$

IV. Data Mining

A. ANOVA

ANOVA is one of the data mining techniques showing the effect of each design variable to the objective and the constraint functions in a quantitative way. ANOVA uses the variance of the model due to the design variables on the approximation function. By decomposing the total variance of model into the variance due to each design variable, the influence of each design variable on the objective function can be calculated. The decomposition is accomplished by integrating out the variables of model \hat{y} .

The total mean ($\hat{\mu}_{total}$) and variance ($\hat{\sigma}_{total}^2$) of model \hat{y} are as follows:

$$\hat{\mu}_{total} \equiv \int \cdots \int \hat{y}(x_1, \dots, x_n) dx_1 \cdots dx_n \quad (29)$$

$$\hat{\sigma}_{total}^2 = \int \cdots \int [\hat{y}(x_1, \dots, x_n) - \hat{\mu}]^2 dx_1 \cdots dx_n \quad (30)$$

The main effect of variable x_i is

$$\hat{\mu}_i(x_i) \equiv \int \cdots \int \hat{y}(x_1, \dots, x_n) dx_1 \cdots dx_{i-1} dx_{i+1} \cdots dx_n - \hat{\mu} \quad (31)$$

The variance due to the design variable x_i is

$$\int [\hat{\mu}_i(x_i)]^2 dx_i \quad (32)$$

The proportion of the variance due to design variable x_i to the total variance of the model can be expressed by dividing Eq. (32) by Eq. (30).

$$\frac{\int [\hat{\mu}_i(x_i)]^2 dx_i}{\int \cdots \int [\hat{y}(x_1, \dots, x_n) - \hat{\mu}]^2 dx_1 \cdots dx_n} \quad (33)$$

This value indicates how much effect design variable x_i gives to the objective function \hat{y} .

B. Self-Organizing Map (SOM)

1. General SOM algorithm

SOM is an unsupervised learning, nonlinear projection algorithm^{32,33} from high to low-dimensional space. This projection is based on self-organization of a low-dimensional array of neurons. In the projection algorithm, the weights between the input vector and the array of neurons are adjusted to represent features of the high dimensional data on the low-dimensional map. The closer two patterns are in the original space, the closer is the response of two neighboring neurons in the low-dimensional space. Thus, SOM reduces the dimension of input data while preserving their features.

A neuron used in SOM is associated with weight vector $\mathbf{m}_i = [m_{i1}, m_{i2}, \dots, m_{in}]$ ($i=1, \dots, M$) where n is equal to the dimension of input vector and M is number of neuron. Each neuron is connected to adjacent neurons by a neighborhood relation and usually forms two-dimensional rectangular or hexagonal topology as shown in Fig. 15.

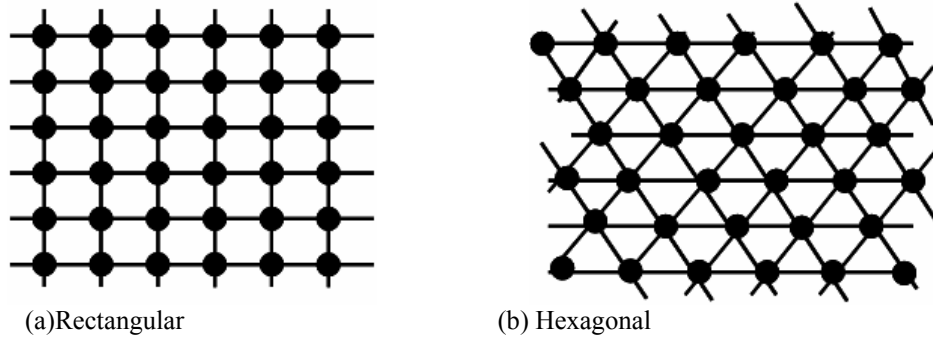


Figure 15 Topology used in SOMs

The learning algorithm of SOM is started with finding the best-matching unit (\mathbf{m}_c) which is closest to the input vector \mathbf{x} as follow:

$$\|\mathbf{x} - \mathbf{m}_c\| = \min_k \|\mathbf{x} - \mathbf{m}_k\| \quad (k = 1, \dots, M) \quad (34)$$

Once the best-matching unit is determined, the weight adjustments are performed not only for the best-matching unit but also for its neighbors. The adjustment depends on the distance (similarity) between the input vector and the neuron. Based on the distance, the best-matching unit and its neighboring become closer to the input vector as shown in Fig. 16. The weight vectors are situated in the cross of the solid lines. The best-matching unit is the weight vector who is closest to the input vector \mathbf{x} . The best-matching unit and its neighbors are adjusted to be closer to the input vector \mathbf{x} . The adjusted topology is represented with dashed lines. Repeating this learning algorithm, the weight vectors become smooth not only locally but also globally. Thus, the sequence of close vectors in the original space results in a sequence of the corresponding neighboring neurons in the two-dimensional map.

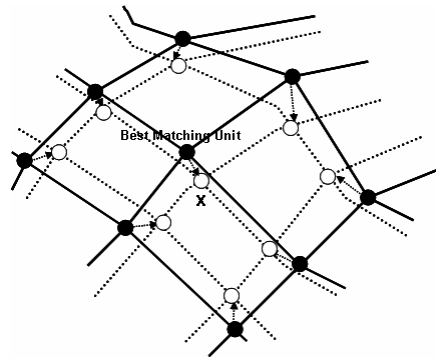


Figure 16 Adjustment of the best-matching unit and its neighbors

2. Kohonen's Batch-SOM

In this investigation, SOMs are generated by using commercial software Viscovery®SOMine plus 4.0³⁴ produced by Eudaptics GmbH. Although SOMine is based on the general SOM concept and algorithm, it employs an advanced variant of unsupervised neural networks, *i.e.* Kohonen's Batch SOM. The algorithm consists of two steps that are iterated until no more significant changes occur: search of the best-matching unit c_i for all input data $\{\mathbf{x}_i\}$ and adjustment of weight vector $\{\mathbf{m}_j\}$ near the best-matching unit. The Batch-SOM algorithm can be formulated as follows:

$$c_i = \arg \min_j \|\mathbf{x}_i - \mathbf{m}_j\| \quad (35)$$

$$\mathbf{m}_j^* = \frac{\sum_i h_{jc_i} \mathbf{x}_i}{\sum_{i'} h_{jc_{i'}}} \quad (36)$$

where \mathbf{m}_j^* is the adjusted weight vector. The neighborhood relationship between two neurons j and k is defined by the following Gaussian-like function:

$$h_{jk} = \exp\left(-\frac{d_{jk}^2}{r_t^2}\right) \quad (37)$$

where d_{jk} denotes the Euclidean distance between the neuron k and the neuron j on the map, and r_t denotes the neighborhood radius which is decreased with the iteration steps t .

The standard Kohonen algorithm adjusts the weight vector after all each record is read and matched. On the contrary, the Batch-SOM takes a 'batch' of data (typically all records), and performs a 'collected' adjustment of the weight vectors after all records have been matched. This is much like 'epoch' learning in supervised neural networks. The Batch-SOM is a more robust approach, since it mediated over a large number of learning steps. In the SOMine, the uniqueness of the map is ensured by the adoption of the Batch-SOM and the linear initialization for input data. Much like some other SOMs, SOMine creates a map in a two-dimensional hexagonal grid. Starting from numerical, multivariate data, the nodes on the grid gradually adapt to the intrinsic shape of the data distribution. Since the order on the grid reflects the neighborhood within the data, features of the data distribution can be read off from the emerging map on the grid. The trained SOM is systematically converted into visual information.

3. Cluster Analysis

Once the high-dimensional data projected on the two-dimensional regular grid, the map can be used for visualization and the data mining. It is efficient to group all neurons by the similarity to facilitate SOM for the qualitative analysis, because number of neurons on the SOM is large as a whole. This process of grouping is called 'clustering'.

Hierarchical agglomerative algorithm is used for the clustering here. First, each node itself forms a single cluster and two clusters, which are adjacent in the map, are merged in each step. The distance between two clusters is calculated by using the SOM-ward distance.³⁰ The number of clusters is determined by the hierarchical sequence of clustering. A relatively small number of clusters are used for visualization, while a large number are used for the generation of weight vectors for respective design variables.

V. Data Mining Results

A. Fly-back Booster of Reusable Launch Vehicle (RLV) Design²⁴

The first example considers the four-objective aerodynamic optimization for the fly-back booster using high fidelity CFD code. The resulting non-dominated front reveals trade-offs in the design space. Two different data mining techniques were applied to the resulting non-dominated front to examine whether they will produce consistent results. Although it is difficult to validate data mining results in general, this example gives a verification of the present data mining approach.

Geometry of the fly-back booster used in the present optimization is shown in Fig. 17(a). The design variables used to define wing shape are related to planform, airfoil, wing twist and relative wing position to fuselage. A wing planform is determined by five design variables as shown in Fig. 17(b). Airfoil shapes are defined at wing root, kink

and tip, respectively, by using thickness and camber distributions. Both distributions are parameterized by using Bezier curves and linearly interpolated in the spanwise direction. Wing twist is refined by using a B-spline curve with six control points. Relative position of the wing root to the fuselage is parameterized by x and z coordinates of the leading edge, angle of attack and dihedral angle. Total 71 design variables are used to wing geometry definition.

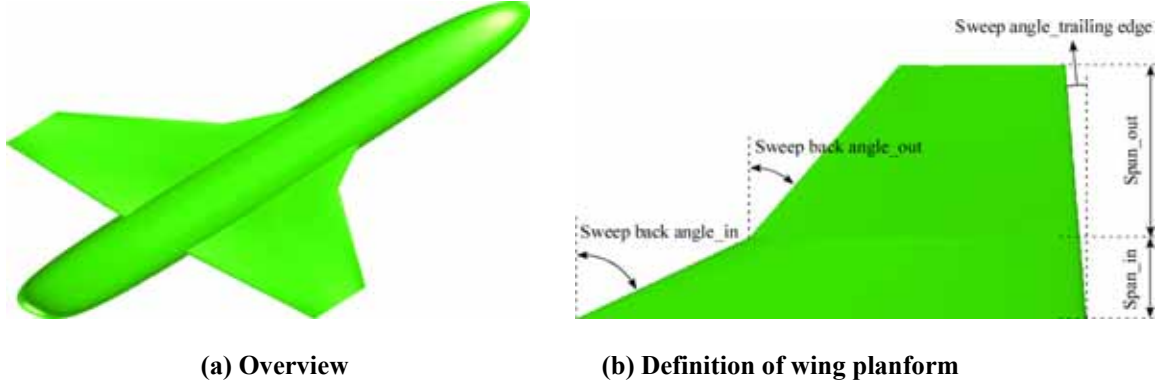


Figure 17 Geometry of fly-back booster

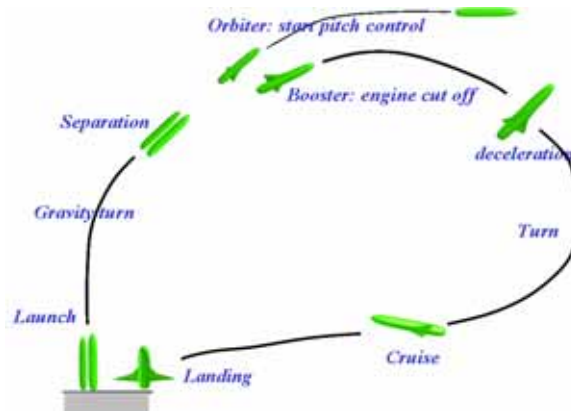


Figure 18 Typical flight sequence for TSTO fly-back booster

According to the trajectory analysis, the separation of the booster and orbiter takes places around Mach 3 and the booster turns over, slows down, cruise at transonic speed and lands at subsonic speed as shown in Fig. 18. In order to maintain good aerodynamic performances in wide flight range, the following 4 objective functions are considered in this design.

1. Minimization of the difference between supersonic pitching moment and transonic pitching moment.

$$F_1 = \left| C_{M_p}^{SUPERSONIC} - C_{M_p}^{TRANSONIC} \right| \quad (38)$$

2. Minimization of the pitching moment at the transonic flight conditions

$$F_2 = \left| C_{M_p}^{TRANSONIC} \right| \quad (39)$$

3. Minimization of the drag at the transonic flight conditions

$$F_3 = C_D^{TRANSONIC} \quad (40)$$

4. Maximization of the lift at the subsonic flight conditions

$$F_4 = C_L^{SUBSONIC} \quad (41)$$

As the optimizer, ARMOGA is used without a surrogate model. The population size of the present ARMOGA is 8 and 40 generations are performed. Figure 19 shows 102 non-dominated solutions obtained by ARMOGA.

However, it is difficult to understand the feature of design space from the Fig. 19. For the better understand of the design space, ANOVA and SOM are performed with 102 non-dominated solutions.

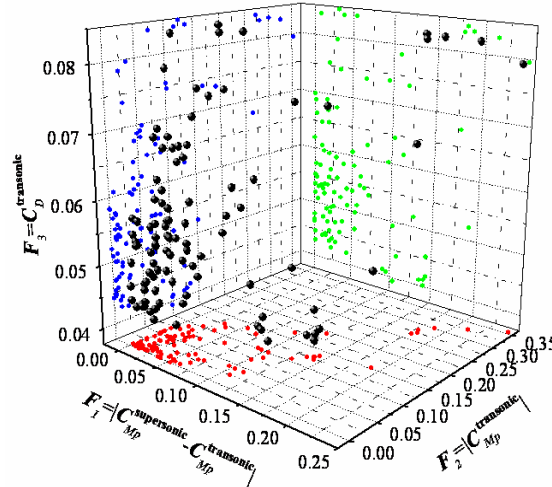


Figure 19 Non-dominated solutions projected onto three-dimensional objective function space

1. ANOVA

ANOVA is performed for four objective functions to analyze the non-dominated front. Variance of design variables and their interactions whose proportion to the total variance is over than 1.0% are shown in Fig. 20.

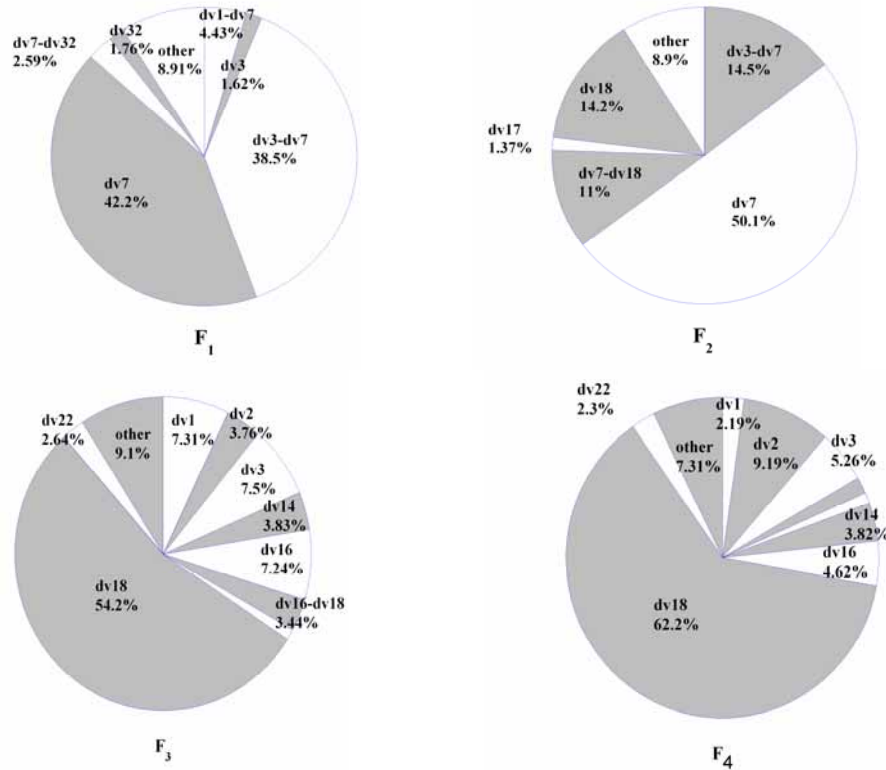
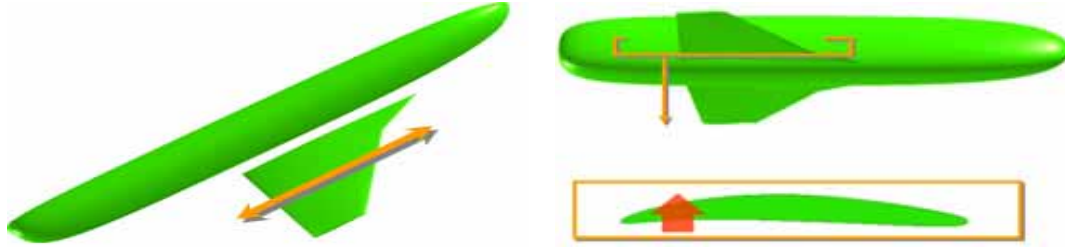


Figure 20 ANOVA results

According to the results, dv7 (x coordinate of relative wing position to fuselage) gives the largest effect on the objective functions F_1 and F_2 . About F_3 and F_4 , dv18 (rearward camber height at kink) gives the largest effect. dv7 and dv18 are illustrated in the Fig. 21. These findings correspond to the aerodynamic knowledge.



(a) x coordinate of relative wing position to fuselage (dv7) (b) rearward camber height at kink (dv18)

Figure 21 Illustrations of dv7 and dv18

2. SOM

SOM is also applied to the non-dominated front. Figure 22 show the resulting SOM colored by respective objective functions. The plots for F_1 and F_2 show similar color patterns. Roughly speaking, F_1 and F_2 can be minimized simultaneously, and thus they are not in the trade-off relation. On the other hand, although the plots for F_3 and F_4 show the similar color distribution, F_3 and F_4 are in a severe trade-off relation because F_3 should be minimized but F_4 should be maximized.

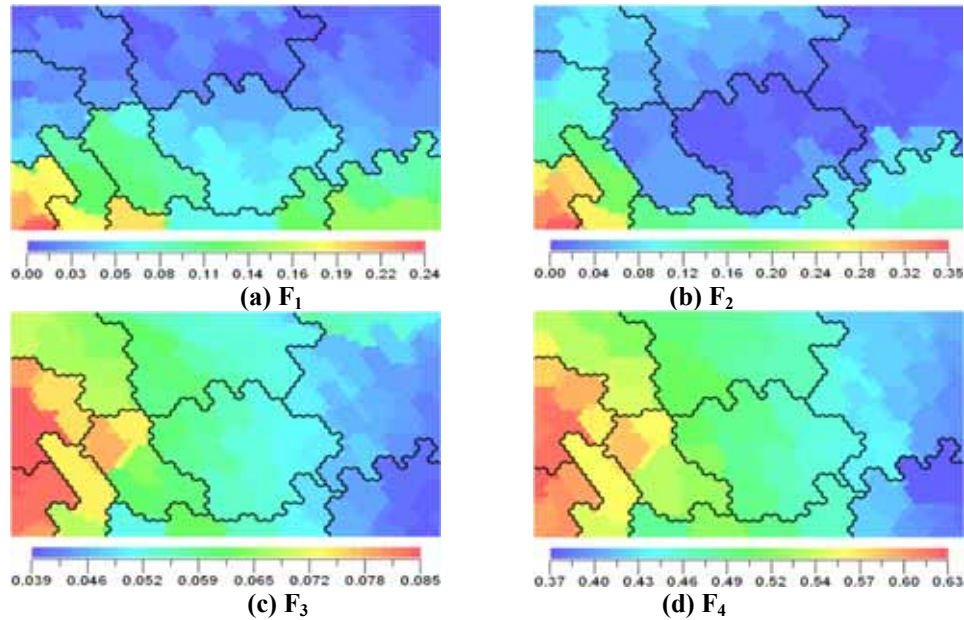


Figure 22 SOM colored by respective objective functions

Figure 23 shows SOM colored by three design variables (dv7, dv18 and dv15). In Fig. 23(a), colored by dv7, large dv7 values can be found at the lower left corner. This area corresponds to large F_1 and F_2 values as shown in Figs. 22(a) and 22(b). This means that large dv7 values lead to poor performances of F_1 and F_2 . In Fig 23(b), colored by dv18, large dv18 values can be found in the left-hand side. This color pattern is very similar to those for F_3 and F_4 as shown in Figs. 22(c) and 22(d). This means that large dv18 values lead to large F_3 and F_4 values. These results suggest that dv7 has a large effect on the objective functions F_1 and F_2 and that dv18 has a large effect on the

objective functions F_3 and F_4 . In Fig. 23(c), colored by dv15 (x coordinate of forward at kink), there is no noticeable trend of color distribution. This means that dv15 has little influence on the objective functions. These results are coincident with the results of ANOVA. The results indicate that ANOVA shows the effect of each design variables on objective functions quantitatively while SOM shows the information qualitatively.

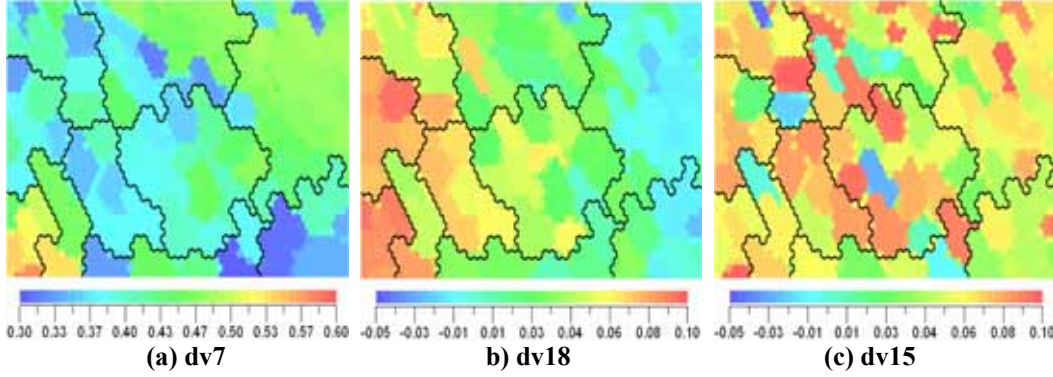


Figure 23 SOM colored by three design variables

B. MDO for Regional-Jet Wing¹⁷

Data mining for a large-scale, real-world MDO problem is shown here. Because high fidelity CFD solvers are desired for transonic wing design, the computational cost for MDO will be enormous. In this example, instead of searching for the *optimal* solution, we have applied ARMOGA to explore the design space briefly. The optimization process was stopped when improvements were observed in all objectives. Then, SOM was applied to visualize the design space by using all the solutions computed so far. Based on the observation, a new wing design has been suggested and the resulting wing has been confirmed to outperform the other computed solutions. This illustrates the importance of the present approach because design knowledge can produce a better design even from the brief exploration of the design space.

1. Multidisciplinary Wing Design

- *Objective Functions*

In this optimization, minimization of the block fuel at a required target range derived from aerodynamics and structures is considered as the primary objective function. In addition, two more objective functions are considered: minimization of the maximum takeoff weight and minimization of the drag divergence between transonic and subsonic conditions.

- *Geometry Definition*

The design variables describe airfoil, twist, and wing dihedral. The airfoil was defined at three spanwise cross-sections using the modified PARSEC with nine design variables (x_{up} , z_{up} , z_{xxup} , x_{lo} , z_{lo} , z_{xxlo} , α_{TE} , β_{TE} , and r_{LElo}/r_{LEup}) for each cross-section as shown in Fig. 24. The twists were defined at six spanwise locations, and then wing dihedrals are defined at kink and tip locations. The entire wing shape was thus defined using 35 design variables.

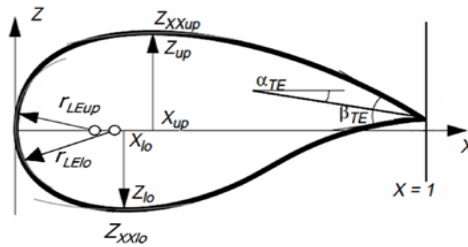


Figure 24 Illustration of the modified PARSEC airfoil definition

- *Evaluation Method*

The present ARMOGA generates eight individuals per generation, and evaluates aerodynamic and structural properties of each design candidate as follows:

1. Structural optimization is performed to jig shape to realize minimum wing weight with constraints of strength and flutter requirements using NASTRAN. And then, weights of wing box and carried fuel are calculated.
2. Static aeroelastic analysis is performed at three flight conditions to determine the aeroelastic deformed shapes (1G shape) using Euler solver and NASTRAN.
3. Aerodynamic evaluations are performed for the 1G shapes using a N-S solver.
4. Flight envelope analysis is performed using the properties obtained as above to evaluate the objective functions. Using the objective functions, the optimizer generates new individuals for the next generation via genetic operations, such as selection, crossover, and mutation.

2. Optimization Results

The population size was set to eight, and then roughly 70 Euler and 90 N-S computations were performed in one generation. It took roughly one and nine hours of CPU time on NEC SX-5 and SX-7 per PE for single Euler and N-S computations, respectively. The population was re-initialized every five generations for the range adaptation. A total evolutionary computation of 19 generations was carried out. The evolution did not converge yet. However, the results are satisfactory because several non-dominated solutions have achieved significant improvements over the initial design. Furthermore, a sufficient number of solutions are searched so that the sensitivity of the design space around the initial design can be analyzed.

Figure 25 shows all solutions projected on a two-dimensional plane between two objectives, the block fuel and the drag divergence. The non-dominated front is formed, indicating the trade-off between the block fuel and the drag divergence. All solutions projected on two-dimensional planes between other combinations are shown in Figs. 26, and 27. As the non-dominated solutions did not comprise a front, these figures indicate that there are no global trade-offs between these combinations of the objective functions.

The comparison between initial and optimized geometries is investigated. Although the wing box weight tends to increase as compared with that of the initial geometry, the block fuel can be reduced. Thus, the aerodynamic performance can redeem the penalty due to the structural weight. An individual on the non-dominated front shown in Fig. 25 is selected, indicated as ‘*optimized*’, and then the *optimized* geometry is compared with the initial geometry.

Although the drag minimization is not considered here, C_D is reduced. By comparison of the polar curves at constant C_L for the cruising condition, C_D of the *optimized* geometry is found to be reduced by 5.5 counts. Due to the improvement of the drag, the block fuel of the *optimized* geometry is decreased by over one percent even with its structural weight penalty.

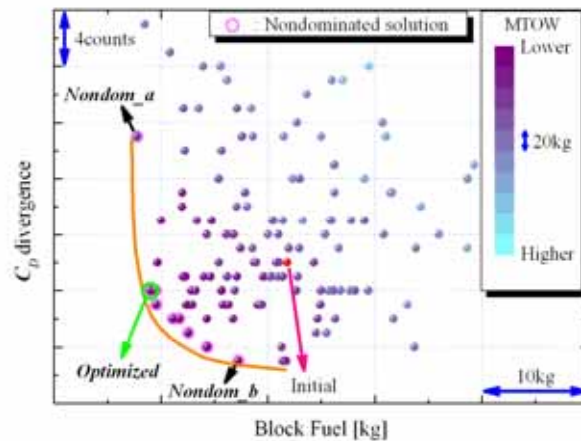


Figure 25 All solutions on two-dimensional plane between block fuel and drag divergence

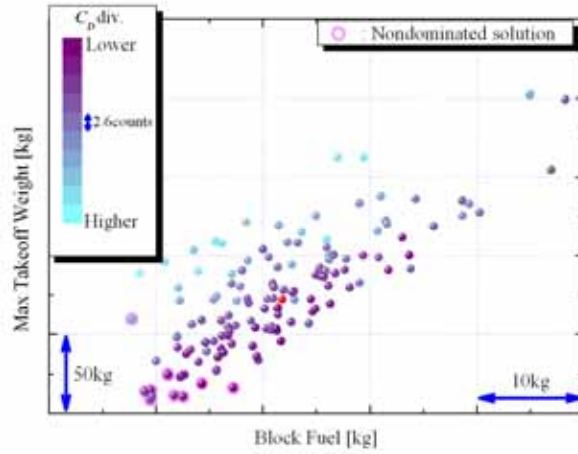


Figure 26 All solutions on two-dimensional plane between block fuel and maximum takeoff weight

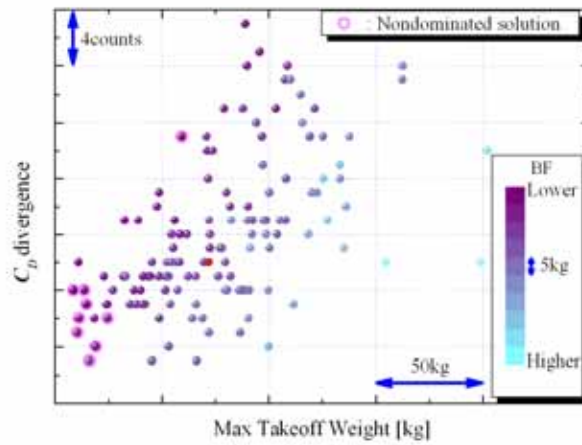


Figure 27 All solutions on two-dimensional plane between maximum takeoff weight and drag divergence

3. Data Mining by SOM

Detailed flow visualization for the *optimized* geometry indicates that the main drag reduction is achieved at the kink location. However, the *optimized* geometry has inverted gull at the kink. Figure 28(a) shows the SOM colored by the angle between inboard and outboard on the upper wing surface for the gull-wing at the kink location. Angles greater and less than 180 deg correspond to gull and inverted gull-wing, respectively. Higher values of this angle as shown in Fig. 28(a) correspond to higher C_D at the transonic cruising flight condition as shown in Fig. 28(b). However, at angles less than 180 deg, there is little correlation between Fig. 28(a) and 28(b). The inverted gull did not affect aerodynamic performance very much. Furthermore, SOM also shows that higher angles shown in Fig. 28(a) correspond to higher maximum takeoff weights as shown in Fig. 28(c). The inverted gull-wing is known to have a structural weight increase, which is also observed in the present results. From the visualization of the design space by SOM, it is suggested that non-gull wings should be designed even though the *optimized* geometry has inverted gull.

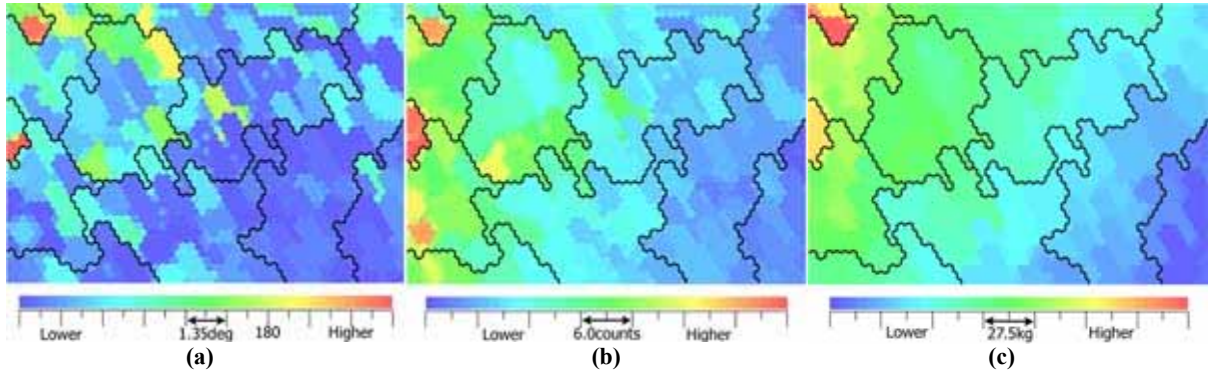


Figure 28 SOM; (a) colored by the angle on upper surface expressing the gull-wing at the kink location, (b) colored by C_D under transonic cruising flight condition, (c) colored by the maximum takeoff weight.

4. Evaluation of the Non-Gull Geometry

The *optimized* wing shape has been modified to examine the non-gull wing shape (called as '*optimized_mod*') can achieve better performance and to verify the design knowledge obtained by the previous data mining.

The result is shown in Figs. 29 to 31. These figures show that *optimized_mod* improves both block fuel and maximum takeoff weight. Moreover, by comparison of the polar curves at constant C_L for cruising condition shown in Fig. 32, C_D of *optimized_mod* is found to be reduced by 10.6 counts over the initial geometry. Due to the improvement of drag, the block fuel of *optimized_mod* is reduced by 3.6 percent.

The present optimization is probably incomplete because only the small number of the generations has been performed. In addition, the automatic mesh generator may clip the design space severely. In the present MDO system, surface spline function of the geometry deviation is used for the modification of the wing surface mesh, and then the volume mesh is modified accordingly by the unstructured dynamic mesh method. However, this process made the surface mesh distorted around the leading edge. This mesh generation might be the primary reason for the difficulty in finding the non-gull geometry. However, the present result demonstrates that data mining can produce a good design even from the results of the incomplete optimization.

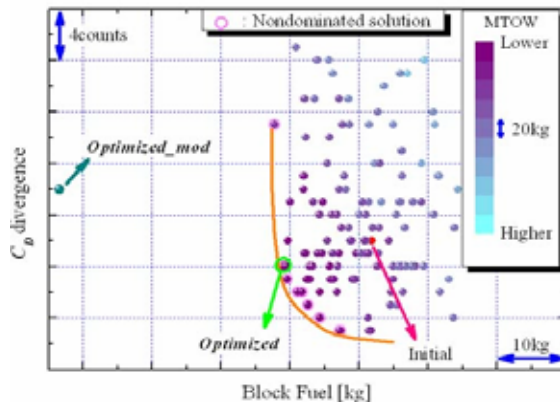


Figure 29

Comparison of *optimized_mod* and all solutions on two-dimensional plane between block fuel and C_D divergence

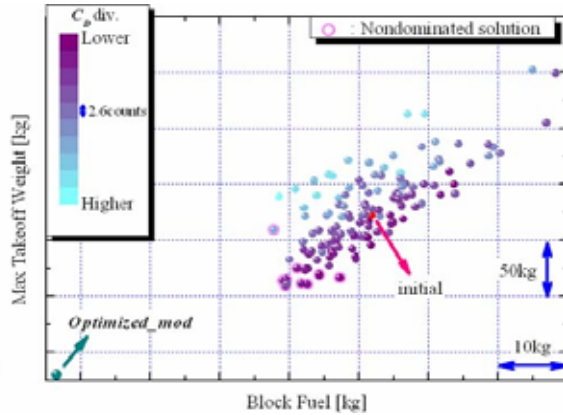


Figure 30

Comparison of *optimized_mod* and all solutions on two-dimensional plane between block fuel and maximum takeoff weight

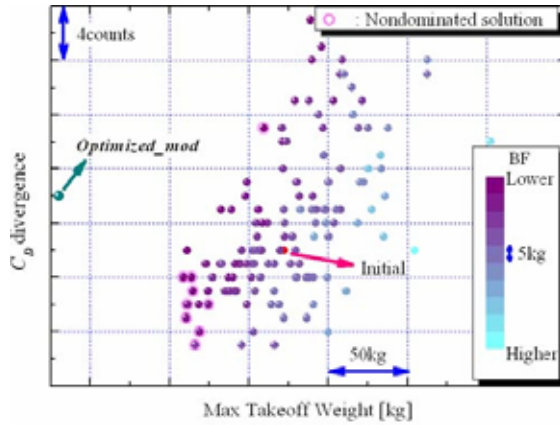


Figure 31

Comparison of *optimized_mod* and all solutions on two-dimensional plane between maximum takeoff weight and C_D divergence

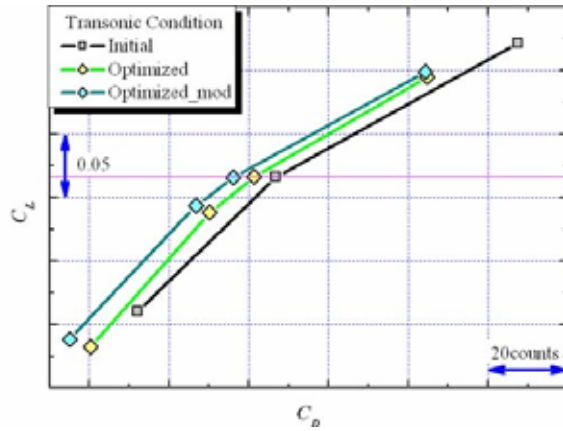


Figure 32

Comparison of the C_L - C_D curves among three geometries as initial, *optimized*, and *optimized_mod* under transonic flight condition

VI. Concluding Remarks

A new approach, MODE, has been presented to address MDO problems. MODE is not intended to give an *optimal* solution. MODE reveals the structure of the design space from the trade-off information and visualizes it as a panorama for DM. DM will know the reason for trade-offs from non-dominated designs, instead of receiving an *optimal* design without trade-off information. The present components of MODE are described, although the concept of MODE can be coupled with other RSM and optimization algorithms.

The main emphasis of this approach is visual data mining. Two data mining examples are presented. The first example considers the four-objective aerodynamic optimization for the fly-back booster using high fidelity CFD code. The resulting non-dominated front reveals trade-offs in the design space. Two different data mining techniques were confirmed to produce consistent results. Although it is difficult to validate data mining results in general, this example gives a verification of the present approach. The results indicate that ANOVA shows the effect of each design variables on objective functions quantitatively while SOM shows the information qualitatively.

The second example considers the high fidelity MDO problem for a regional-jet wing. It optimizes aerodynamic performance and structural weight under aeroelastic constraints. Because the design space was large and high fidelity simulation codes were time-consuming, ARMOGA was used to explore the design space briefly. The optimization was stopped after improvements were obtained. Then, SOM was applied to visualize the design space. Based on the observation, a new, better wing design has been suggested. This illustrates the importance of MODE because design knowledge can produce a better design even from the brief exploration of the design space.

Although it is not discussed in this paper, the flowchart of MODE shown in Fig. 4 has feedback loops. The design space can be redefined by analyzing the surrogate model.³⁵ Moreover, from data mining, competing objectives and active constraints can be identified. This will lead to the re-definition of the MDO problem itself (Fig. 33). MDO often uses conceptual performance equations as design objectives. However, sensitivities of those equations to high fidelity simulation codes are not well understood. As more and more high fidelity simulation codes become available to MDO, selection of objective functions will become more crucial. The outermost feedback loop in Fig. 33 will be essential to address this issue.

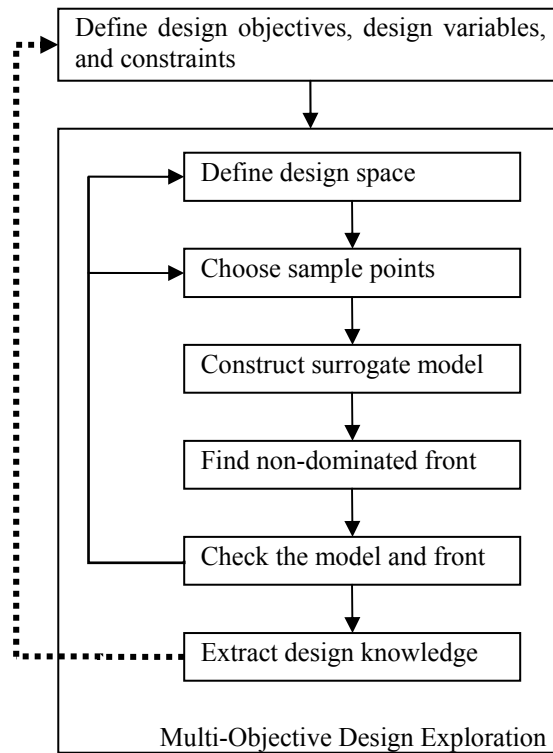


Figure33 Future direction of Multi-Objective Design Exploration (MODE)

Acknowledgments

The authors would like to acknowledge Advanced Fluid Information Research Center, Institute of Fluid Science, Tohoku University, for providing the computer resources. The authors would also like to thank Mitsubishi Heavy Industries, Ltd., for their support and cooperation to the present work. The part of this research is sponsored by the New Energy Development Organization of Japan (NEDO) under the small jet aircraft R&D project. The first author's work is also supported by No. 15206091, Grants-In-Aid for Scientific Research, Japan.

References

- ¹ Sobieszczanski-Sobieski, J. and Haftka, R. T., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," *Structural Optimization*, Vol. 14, No. 1, 1997, pp.123.
- ² Kodiyalam, S. and Sobieszczanski-Sobieski, J., "Multidisciplinary design optimization-some formal methods, framework requirements, and application to vehicle design," *Int. J. Vehicle Design* (Special Issue), 2001, pp. 3-22.
- ³ Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., Tucker, P. K., "Surrogate-based analysis and optimization," *Progress in Aerospace Sciences*, Vol. 41, No. 1, 2005, pp. 1-28.
- ⁴ Stewart, T. J., "A critical survey on the status of multiple criteria decision making and practice," *Int. J. Management Sci.*, 20, 1992, pp. 569-586.
- ⁵ Vanderplaats, G. N., *Numerical optimization techniques for engineering design*, McGraw-Hill Book Company, 1984.
- ⁶ William, H. P., Saul, A. T., William, T. V., and Brian, P. F., *Numerical recipes in Fortran 77: The art of scientific computing*, Cambridge Univ. Press, Cambridge, 1996, pp 436-448.
- ⁷ Das, I. and Dennis, J. E. "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, Vol. 8, No. 3, 1998, pp. 631-657.
- ⁸ Nakayama, H., "Aspiration Level Approach to Interactive Multi-objective Programming and its Applications," Paldalos, P. M., Siskos, Y. and Zopounidis, C. (eds.), *Advances in Multicriteria Analysis*, Kluwer Academic Publishers, 1995, pp. 147-174.
- ⁹ Deb, K., *Multi-objective optimization using evolutionary algorithms*, John Wiley & Sons, Ltd., New York, 2001.

- ¹⁰ Coello Coello, A. C., Lamont, G. B., *Applications of Multi-objective Evolutionary Algorithms*, World Scientific Publishing Co. Pte. Ltd., Singapore, 2004.
- ¹¹ Bäck, T., Fogel, D. B., and Michalewicz, Z., *Handbook of Evolutionary Computation*, IOP Publishing Ltd., Bristol and Oxford University Press, New York, 1997.
- ¹² Fonseca, C. M. and Fleming, P. J., "Genetic Algorithms for Multi-objective Optimization: Formulation, Discussion and Generalization," *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1993, pp.416-423.
- ¹³ Giannakoglou, K. C., "Design of Optimal Aerodynamic Shapes using Stochastic Optimization Methods and Computational Intelligence," *Progress in Aerospace Sciences*, Vol. 38, 2002, pp. 43-76.
- ¹⁴ Oyama, A., Obayashi, S., Nakamura, T. "Real-coded adaptive range genetic algorithm applied to transonic wing optimization," *Applied Soft Computing*, Vol. 1, 2001, pp. 179-187.
- ¹⁵ Sasaki, D. and Obayashi, S., "Efficient Search for Trade-Offs by Adaptive Range Multi-objective Genetic Algorithms," *Journal of Aerospace Computing Information, and Communication*, Vol. 2, No. 1, 2005, pp. 44-64.
- ¹⁶ Nemec, M., Zingg, D. W., Pulliam, T. H., "Multipoint and Multi-objective Aerodynamic Shape Optimization," *AIAA Journal*, Vol.42, No.6, 2004, pp. 1057-1065.
- ¹⁷ Chiba, K., Obayashi, S., Nakahashi, K. and Morino, H., "High-Fidelity Multidisciplinary Design Optimization of Aerostructural Wing Shape for Regional Jet," *AIAA Paper 2005-5080*, June 2005.
- ¹⁸ Parussini, L., Pediroda, V., and Obayashi, S., "Design under Uncertainties of Wings in Transonic Field," *JSME International Journal*, Series B, Vol.48, No.2, 2005, pp.218-228.
- ¹⁹ Epstein, B., and Peigin, S., "Robust Hybrid Approach to Multi-objective Constrained Optimization in Aerodynamics," *AIAA Journal*, Vol. 42, No. 8, 2004, pp. 1572-1581.
- ²⁰ Fleming, P. J., Purshouse, R. C., Lygoe, R. J., "Many-Objective Optimization: An Engineering Design Perspective," *Evolutionary Multi-Criterion Optimization (EMO2005)*, LNCS 3410, Springer, Heidelberg, 2005, pp. 14-32.
- ²¹ Parmee, I. C. and Abraham, J. A. R., "Supporting Implicit Learning via the Visualisation of COGA Multi-objective Data," *Proceedings of IEEE Congress on Evolutionary Computation*, 2004, pp. 395-402.
- ²² Obayashi, S. and Sasaki, D., "Visualization and Data Mining of Pareto Solutions Using Self-Organizing Map," *Evolutionary Multi-Criterion Optimization (EMO2003)*, LNCS 2632, Springer, Heidelberg, 2003, pp. 796-809.
- ²³ Goel, A., Baker, C. A., Shaffer, C. A., Grossman, B., Hafika, R. T., Mason, W. H., Watson, L. T. "VizCraft: A Problem Solving Environment for Configuration Design of a High Speed Civil Transport," *IEEE Computing in Science & Engineering*, Vol. 3, No. 1, 2001, pp.56-66.
- ²⁴ Jeong, S., Chiba, K., and Obayashi, S., "Data Mining for Aerodynamic Design Space," *AIAA Paper 2005-5079*, June 2005.
- ²⁵ Giannakoglou, K. C., Giotis, A. P., and Karakasis, M. K., "Low-Cost Genetic Optimization Based on Inexact Pre-Evaluations and the Sensitivity Analysis of Design Parameters," *Journal of Inverse Problems in Engineering*, Vol. 9, 2001, pp.389-412.
- ²⁶ Willson, B., Cappelleri, D., Simpson, T. W., Frecker, M., "Efficient Pareto Frontier Exploration using Surrogate Approximations," *Optimization and Engineering*, Vol. 2, 2001, pp. 31-50.
- ²⁷ Keane, A. J., "Wing Optimization Using Design of Experiment, Response Surface, and Data Fusion Methods," *Journal of Aircraft*, Vol. 40, No. 4, 2003, pp. 741-750.
- ²⁸ Takenaka, K., Nakahashi, K., Obayashi, S., Matsushima, K., "The Application of MDO Technologies to the Design of a High Performance Small Jet Aircraft - Lessons Learned and Some Practical Concerns," *AIAA Paper 2005-4797*, June 2005.
- ²⁹ Koehler, J and Owen, A. Computer experiments, in S. Ghosh and C. R. Rao (eds.), *Handbook of Statistics*, 13: Design and Analysis of Experiments, Elsevier, Amsterdam, 1996, pp. 261-308.
- ³⁰ Mardia, K. V. and R. J. Marshall, "Maximum likelihood estimation of models for residual covariance in Spatial regression," *Biometrika* Vol. 71, 1984, pp. 135-146.
- ³¹ Donald, R. J., Matthias S and William J. W, "Efficient Global Optimization of Expensive Black-Box Function," *Journal of global optimization*, Vol. 13, 1998, pp. 455-492.
- ³² Kohonen, T., *Self-Organizing Maps*, Springer, Berlin, Heidelberg, 1995.
- ³³ Krzysztof, J. C., Witold, P. and Roman, W. S., *Data Mining Methods for Knowledge Discovery*, Kluwer Academic Publisher, 1998.
- ³⁴ Eudaptics software gmbh, <http://www.eudaptics.com/somine/>, last access on April 14, 2005.
- ³⁵ Jeong, S., Murayama, M. and Yamamoto, K., "Efficient Optimization Design Method Using Kriging Model," *Journal of Aircraft*, Vol. 42, 2005, pp.413-420.