



Progress in design optimization using evolutionary algorithms for aerodynamic problems

Yongsheng Lian^a, Akira Oyama^b, Meng-Sing Liou^{c,*}

^a University of Louisville, Louisville, KY 40292, USA

^b Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency, 3-1-1 Yoshinodai Sagamihara, Kanagawa 229-8510, Japan

^c MS 5-11, NASA Glenn Research Center, Cleveland, OH 44135, USA

ARTICLE INFO

Available online 24 September 2009

Keywords:

Multi-objective design optimization

Evolutionary algorithms

Surrogate model

Robust and reliability-based design

Data mining

NASA rotor 67 blade

ABSTRACT

Evolutionary algorithms (EAs) are useful tools in design optimization. Due to their simplicity, ease of use, and suitability for multi-objective design optimization problems, EAs have been applied to design optimization problems from various areas. In this paper we review the recent progress in design optimization using evolutionary algorithms to solve real-world aerodynamic problems. Examples are given in the design of turbo pump, compressor, and micro-air vehicles. The paper covers the following topics that are deemed important to solve a large optimization problem from a practical viewpoint: (1) hybridized approaches to speed up the convergence rate of EAs; (2) the use of surrogate model to reduce the computational cost stemmed from EAs; (3) reliability based design optimization using EAs; and (4) data mining of Pareto-optimal solutions.

Published by Elsevier Ltd.

Contents

1. Brief introduction of evolutionary algorithms	200
1.1. Evolutionary algorithms	200
2. Optimization using hybridized EAs	200
2.1. Surrogate model	201
2.2. A trust region management	201
2.3. Numerical tests with hybrid method	202
2.4. Remarks on hybrid approaches	203
3. Optimization using surrogate model	203
3.1. Response surface methodology	204
3.2. Latin hypercube designs	204
3.3. The coupled model	204
3.4. Example: multi-objective design of NASA rotor 67 blade	205
4. Robust and reliability-based design	209
4.1. Problem formulation	211
4.2. Monte Carlo simulation	212
4.3. Probabilistic sufficiency factor	212
4.4. Reliability-based optimization using response surface approximation	213
4.5. RBDO procedure	213
4.6. Numerical results for RBDO	214
4.7. Error analysis	215
5. Data mining of Pareto-optimal solutions	215
5.1. Overview	215
5.2. Data mining of the Pareto-optimal solutions of an aerodynamic flapping airfoil design	217
5.2.1. Design optimization problem	217
5.2.2. Aerodynamic force evaluation	217

* Corresponding author. Tel.: +1 216 433 5855; fax: +1 216 433 5802.

E-mail address: meng-sing.liou@nasa.gov (M.-S. Liou).

5.2.3.	Data mining	218
5.2.4.	Results and discussion	218
5.3.	Data mining of Pareto-optimal solutions using proper orthogonal decomposition	219
5.3.1.	Analyzed solutions	220
5.3.2.	POD-based data mining of Pareto-optimal solutions	220
5.3.3.	Result	221
	Acknowledgments	222
	References	222

1. Brief introduction of evolutionary algorithms

1.1. Evolutionary algorithms

Evolutionary algorithms (EAs) mimic mechanics of natural selection and natural genetics, in which a biological population evolves over generations to adapt to an environment by selection, crossover, and random mutation. Likewise, EAs start with a random population of candidates (chromosomes), for each of them both the objective and constraint functions are evaluated. Based on the objective function value and constraint violations a metric (fitness) is defined and assigned to each candidate. In general, penalty is put on infeasible candidates so that all infeasible solutions have a worse fitness than feasible solutions. Typically EAs involve three operators, selection, crossover, and mutation. The primary purpose of selection operator is to make duplicates of good candidates and eliminate bad candidates in a population while often keeping population size constant [16]. Tournament selection, proportionate selection, and ranking selection are common methods to achieve the task. For single-objective optimization problems, the ranking is based on the fitness of a candidate. For multi-objective optimization problems, the ranking can be based on Fonseca's non-dominated ranking method in which an individual's rank is equal to the number of individuals in the present generation who are better than the corresponding individual in all the objective functions [23]. After ranking, the N best candidates, same size as the initial population, are chosen from both the current and previous generations and then placed in the mating pool. The elitist strategy [17] is often chosen to ensure a monotonic improvement for the EA, in which some of the best individuals are copied directly into the next generation without applying any evolutionary operators. The ranking selection method assigns selection probability p_s based on an individual's rank instead of its fitness value. The selection probability function is defined as

$$p_s = c(1 - c)^{(\text{rank}-1)}, \quad (1)$$

where $c < 1$ is a user-defined parameter. That is, the selection probability is reduced by a factor of $(1 - c)$ each time when the rank is increase by 1. Then, a pair of parents are selected by using either stochastic universal sampling or roulette-wheel section, with which same candidate can be selected more than once to become a parent. As a consequence, the selection operator ensures that good candidates are preserved at the cost of bad candidates.

A crossover operator is applied next to create offsprings. For example, using the blend crossover (BLX- α) operator, two parent candidates, Parent 1 and Parent 2, will have two offspring as follows:

$$\text{Child 1} = \gamma \text{Parent 1} + (1 - \gamma) \text{Parent 2}, \quad (2)$$

$$\text{Child 2} = (1 - \gamma) \text{Parent 1} + \gamma \text{Parent 2}, \quad (3)$$

where

$$\gamma = (1 + 2\alpha)u - \alpha \quad (4)$$

u is a uniform random number in the range of [0 1]. The value of $\alpha = 0.5$ is usually chosen because it can maintain a good balance

between two conflicting goals of a crossover operator: exploiting good solutions and exploring the search space [9].

The mutation operator is used to maintain the population diversity. For real-coded EA a uniform random number is added to each design variable at a probability of p_m . If x_i represents such a design variable, then the corresponding variable of the new offspring after mutation has the following form:

$$x_i^{\text{new}} = x_i[1 + (r_i - 0.5)\Delta_i], \quad (5)$$

where x_i^{new} is the design variable after mutation, r_i is a random number ranging [0 1], Δ_i is the user-defined maximum perturbation allowed in the i -th decision variable. We set the mutation probability $p_m = 0.1$ and mutation amplitude $\Delta_i = 0.1$ in our computations.

EAs have been successfully applied to aerodynamic design optimization problems because of their ease of use, broad applicability, and global perspective. For example, Oyama et al. applied an EA in their redesign of the NASA Rotor67 transonic compressor blade [64]. Benini implemented an EA to improve the performance of the NASA rotor37 blade [7]. Oyama and Liou [63] and Lian et al. [44,47] utilized EAs to the redesign of rocket turbo pumps. Another thrust behind these broad applications is that EAs are particularly suitable for multi-objective optimization problems, which are often encountered in aerospace designs. For example, in the turbo pump design problems, the objectives are to maximize the total head rise and to minimize the input power [63,44,47]. These two objectives are competing: improving one objective will inevitably deteriorate the other. Unlike a single objective optimization problem, a multi-objective optimization problem does not have such an optimal solution that is better than others in terms of all objectives. Instead, one expects a set of compromised solutions, each of which is better than the others in one objective but is worse in the other objectives. These solutions are largely known as the non-dominated solutions or Pareto-optimal solutions. When dealing with multi-objective optimization problems, classical methods, such as the gradient-based methods, usually convert a multi-objective problem into multiple single-objective problems by introducing parameters such as weight vectors [16]. In this approach, each optimal solution is associated with a particular vector. To find another Pareto-optimal solutions, one has to choose a different weight vector and solve the resulting single-objective optimization problem repeatedly. On the other hand, EA's population approach can be exploited to emphasize all non-dominated solutions in a population equally and to preserve a diverse set of multiple non-dominated solutions using a niche-preserving operator [16]. As a consequence, EAs eliminate the need of choosing different parameters and can find as many Pareto-optimal solutions as possible in one run.

2. Optimization using hybridized EAs

EAs are powerful tools in optimization. However, EAs suffer a slow convergence because they do not use gradient information. The required population size and generation size usually demand tremendous amount of computing resources. For instance, Benini redesigned the NASA rotor37 with an EA, with a population size of

20 and generation size of 100. The overall turn around time was 2000 h using a four-processor workstation [7]. Oyama et al. used an EA to redesign the NASA Rotor67, with a population size of 64 and generation size of 100. The turn around time was about two months using a 64-processor cluster [64]. Therefore, when expensive function evaluations are involved, the required CPU time impedes practical applications of EAs even with today's computing power.

Researchers have proposed numerous approaches to improve the efficiency of EAs. One approach is to hybridize stochastic EAs with deterministic gradient-based methods. The idea is to resort to gradient-based methods whenever the EA convergence rate is slow. This strategy takes the advantage of the fast convergence of the gradient-based methods. Muyl et al. [55] coupled a GA with a BFGS (Broyden-Fletcher-Goldfarb-Shanno [75]) method to optimize an automobile shape. Ong et al. [60] coupled an EA with a feasible sequential quadratic programming (SQP) solver. These proposed approaches typically decompose the original problem into a sequence of subproblems confined to a subregion of original design space. In each subregion, a surrogate model is constructed and optimized with gradient-based methods. These works focus on single-objective optimization.

Lian et al. [43] proposed a strategy to enhance the EA performance by coupling an EA with a gradient-based method for multi-objective design problems. The gradient-based method is an SQP solver [19]. In their approach, the GA was first used to generate a population of data by evaluating real functions, and then, surrogate models were built. Thereafter, they conducted local search on surrogate models. The GA and local search were alternately used under a trust-region framework until an optimum was found. Alexandrov et al. [2] showed that under certain assumptions, solutions produced from surrogate models under a trust region framework converge to the optimum of the original problem. By hybridizing a GA and a gradient-based method under a trust-region framework, favorable characteristics of both local search and global search are maintained. The local search determines a faster convergence. The global search assures that solution produced by an optimization algorithm working with the surrogate models, which started at an arbitrary initial value, will converge to a stationary point or local optimum for the original problem.

The building blocks of their hybrid approach include a real-coded GA, an SQP solver, and a surrogate model. Next we will briefly review the surrogate model and the SQP solver.

2.1. Surrogate model

A nonlinear inequality constraint problem in general has the following form:

$$\begin{aligned} &\text{Minimize } f(\vec{x}) \\ &\text{Subject to } g_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, p, \quad \vec{x}_l \leq \vec{x} \leq \vec{x}_u. \end{aligned} \quad (6)$$

This problem has a set of constraints $\{g_i\}_1^p$, $\vec{x} \in R^n$ is the design variable vector, and \vec{x}_l and \vec{x}_u are the lower and upper bounds of the design variables, respectively. In practice, when $f(\vec{x})$ and $g_i(\vec{x})$ are computationally expensive, instead of directly solving Eq. (6), the following approximate problem will be solved:

$$\begin{aligned} &\text{Minimize } \hat{f}(\vec{x}) \\ &\text{Subject to } \hat{g}_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, p, \quad \vec{x}_l \leq \vec{x} \leq \vec{x}_u, \end{aligned} \quad (7)$$

where $\hat{f}(\vec{x})$ and $\hat{g}_i(\vec{x})$ are the surrogate models of $f(\vec{x})$ and $g_i(\vec{x})$, respectively. Popular surrogate techniques include response surface methods and Kriging methods.

In constructing surrogate models, because neighboring individuals have more impact than remote ones, this motivates the use of radial basis functions in our work. The thin-plate splines (TPS) interpolation [20] was chosen by Lian et al. [43]. Suppose (\vec{x}_0, y_0) is an arbitrary individual, around which surrogate models will be constructed, and $\{\vec{x}_i, y_i\}_1^m$ are its closest neighboring points, then the surrogate model of $f(\vec{x})$ can be constructed as follows:

$$\hat{f}(\vec{x}) = \sum_{j=1}^m \alpha_j \|\vec{x} - \vec{x}_j\|^2 \log \|\vec{x} - \vec{x}_j\|. \quad (8)$$

$\{\alpha_j\}_1^m$ denote the vector of weights, which can be determined by solving a system of linear equations. Substituting $\{\vec{x}_i, y_i\}_1^m$ into Eq. (8) gives a linear system of equations:

$$\sum_{j=1}^m \alpha_j \|\vec{x}_i - \vec{x}_j\|^2 \log \|\vec{x}_i - \vec{x}_j\| = y_i, \quad i = 1, 2, \dots, m. \quad (9)$$

$\{\alpha_j\}_1^m$ can be determined by solving Eq. (9). Micchelli proved that Eq. (9) has a unique solution when the set of neighboring points $\{\vec{x}_i, y_i\}_1^m$ are distinct. Parameter m determines the number of points used to construct the surrogate model, it plays an important role in the overall performance of the hybrid method. It is observed that the resulting linear system of equations becomes ill-conditioned when some neighboring points are close to each other, which becomes severer when a local optimum is approached. To avoid unnecessary numerical error, double-precision computations are generally recommended to solve Eq. (9). Also, a suitable preconditioning for handling ill-conditioned matrices may be necessary. Surrogate models for constraint functions can be formulated in the same way.

2.2. A trust region management

A trust-region framework is used to manage surrogate models to assure that the obtained solution from the surrogate model converges to the original problem. Classical trust region algorithm uses a quadratic model based on the Taylor series expansion, which produces a good approximation confined in a small neighborhood. In engineering practice, it is more practical to use other surrogate models which have better approximations in a larger neighborhood. Previous work in this field exclusively concentrated on whether or not the optimization technique would converge to a solution of the surrogate model, rather than the original problem. Alexandrov et al. [2] presented an approach which inherits the convergence properties of the classical trust region algorithms. Under simple conditions their approach assures that the solution produced by using the surrogate models converge to a local minimum of the original problem regardless of the initial condition. Rodriguez et al. [72] extended this analysis to nonlinear programming problems with general constraints by using augmented Lagrangian methods. Giunta and Eldred [27], Ong et al. [60], and Lian et al. [43] retained separate objective and constraint functions in their surrogate model application, which is outlined as follows:

1. Generate a database with a GA by evaluating computationally expensive exact models.
2. Proceed by building surrogate models and perform a sequence of K local searches based on the surrogate models under the trust region framework. We noticed that three local searches give the best overall performance.
3. Construct surrogate models for the original problem in each subregion.
4. Optimize in the subregion using surrogate models with an SQP solver and obtain a local optimum.
5. Adjust the subregion (or surrogate models) according to the accuracy of the surrogate models. Perform a local search if

further improvement is achievable. Update the individuals in the population with the improved solutions.

6. Return to Step 1 if further iteration is needed.

In Step 2 a series of problems with the following form is solved:

$$\begin{aligned} & \text{Minimize } \hat{f}^k(\bar{x}_c^k + \vec{s}) \\ & \text{Subject to } \hat{g}_i^k(\bar{x}_c^k + \vec{s}) \leq 0, \quad i = 1, 2, \dots, p, \quad \|\vec{s}\| \leq \delta^k. \end{aligned} \quad (10)$$

In the above problem, \bar{x}_c^k is the starting point for the k -th local search. Surrogate models \hat{f}^k and \hat{g}_i^k are constructed based on the m neighboring points around \bar{x}_c^k ; \vec{s} denotes the prospective descending step size and direction; δ^k is the trust radius; the optimal solution at the k -th search is denoted as \bar{x}_{lo}^k . After performing optimization on the surrogate models, the real functions are resorted to recalibrate trust radius by comparing the actual and predicted improvements, and then local search with surrogate models will be continued. In principle, if the surrogate models are accurate enough to predict the improvement of the exact models, the trust radius will be enlarged; or if the surrogate models are not accurate, either no improvement is made or the improvement is not as much as the predicted, then the trust radius is reduced; otherwise, if reasonable but not great improvement is made, the trust radius [2] will remain the same. The trust radius is updated in a dynamic way based on a measure indicating the accuracy of the surrogate models. For single-objective optimization problem, this measure of merit is designated ρ^k and is calculated as follows:

$$\rho^k = \min(\rho_f^k, \rho_{gi}^k) \quad \text{for } i = 1, 2, \dots, p, \quad (11)$$

where

$$\rho_f^k = \frac{f(\bar{x}_c^k) - f(\bar{x}_{lo}^k)}{\hat{f}^k(\bar{x}_c^k) - \hat{f}^k(\bar{x}_{lo}^k)}, \quad \rho_{gi}^k = \frac{g_i(\bar{x}_c^k) - g_i(\bar{x}_{lo}^k)}{\hat{g}_i^k(\bar{x}_c^k) - \hat{g}_i^k(\bar{x}_{lo}^k)}. \quad (12)$$

Intuitively we can see that the larger the ρ^k , the more accurate the surrogate models, and we can increase our trust radius; otherwise, we need to decrease it to increase accuracy. Mathematically, the trust region size is updated based on the value of ρ^k :

$$\begin{aligned} \delta^{k+1} &= 0.25\delta^k \quad \text{if } \rho^k \leq 0.25, \\ &= \delta^k \quad \text{if } 0.25 < \rho^k < 0.75, \\ &= \xi\delta^k \quad \text{if } \rho^k \geq 0.75, \end{aligned} \quad (13)$$

where $\xi = 2$ if $\|\bar{x}_{lo}^k - \bar{x}_c^k\|_\infty \geq \delta^k$, or $\xi = 1$ if $\|\bar{x}_{lo}^k - \bar{x}_c^k\|_\infty < \delta^k$. With this measure we can update the starting point of the next iteration as follows:

$$\begin{aligned} \bar{x}_c^{k+1} &= \bar{x}_{lo}^k \quad \text{if } \rho^k > 0, \\ &= \bar{x}_c^k \quad \text{if } \rho^k \leq 0. \end{aligned} \quad (14)$$

In practice, Eq. (10) is only solved approximately. The step size is acceptable if $\|\vec{s}\| \leq \alpha\delta_k$ for $\alpha > 1$ independent of k [2]. In the work of Lian et al. [43], δ is set as the maximal distance between the starting point \bar{x}_c^k and the neighboring points, with which the surrogate model is constructed.

For multi-objective optimization problems (MOOP), to construct the surrogate model, the MOOP is reformulated as a single objective problem by maintaining only one objective function while putting the others as constraints with user-specified values. This approach in the literature is called the ε -constraint method.

Lian et al. [43] noticed that for MOOPs Eq. (11) is no longer a very good measure of merit for surrogate models, so they proposed another approach. What they did was to compare the local search solution $f(\bar{x}_{lo}^k)$ with the existing solutions, if it is located on the Pareto-optimal front, then they claim improvement is made.

For unconstrained problems, to ensure convergence, they enforced consistency conditions between the original functions and the surrogates. Specifically, the following conditions must hold for both the objective and constraint functions:

$$\begin{aligned} y(\bar{x}_c^k) &= \hat{y}(\bar{x}_c^k), \\ \nabla y(\bar{x}_c^k) &= \nabla \hat{y}(\bar{x}_c^k). \end{aligned} \quad (15)$$

The first condition assure surrogate model has a good approximation near \bar{x}_c^k , which is guaranteed when the surrogate model is constructed with TPS interpolation [20]. However, the second condition is not easy to satisfy because it is computationally prohibitive to evaluate the gradient with finite difference methods if the real function is not known.

2.3. Numerical tests with hybrid method

In the work of Lian et al. [43] a real-coded GA was used, the blend crossover (BLX- α) operator was used with a value of $\alpha = 0.5$, and all design variables were scaled to [0 1]. The uniform mutation operator was chosen to add a uniform random number to the parent solution at a probability of p_c :

$$y_i = x_i + (r_i - 0.5)\Delta_i, \quad (16)$$

where r_i is a random number, Δ_i is the user-defined maximum perturbation allowed in the i -th decision variable. Both p_c and Δ_i were set to 0.1. For single objective optimization, sorting was based on the value of the objective function; for multi-objective optimization, solutions were ranked based on Goldberg's non-dominated sorting procedure [79]. To maintain a uniform distribution on the Pareto-optimal front, they used fitness sharing [26] in the multi-objective optimization. Local search was performed with an SQP solver available in commercial optimizer DOT [19].

The Rastrigin function, which is commonly used in the literature on global optimization, was chosen as an example. The Rastrigin function is defined as follows:

$$\begin{aligned} & \text{Minimize } 10n + \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i)] \\ & \text{Subject to } -5.12 \leq x_i \leq 5.12, \quad i = 1, 2, \dots, n. \end{aligned} \quad (17)$$

This function has a highly bumpy surface with many local optima. Fig. 1 shows the plot of a two-dimensional Rastrigin function. The function has a global minimal value of zero at $x_i = 0$. The population size was set to 100. To compare the pure GA and the hybrid method, the total function evaluations was set to 10,000. The pure GA approaches 16 after 10,000 function evaluations, while the hybrid method converges to 8 after about 4000 function evaluations. Other trials with different initial conditions show similar results. This test demonstrates advantages of the hybrid method for high-dimensional problems (see Fig. 2).

The single-stage centrifugal pump problem computed by Oyama and Liou with a real-coded GA [63] is chosen as another example. The population size was set as 120 at each generation. The computation terminates after 10,000 function evaluations. The result is shown in Fig. 3 together with the original design. The comparison between a GA and a hybrid GA is not as straightforward as the single objective optimization because the final solution is a Pareto-optimal front. For multi-objective optimization, the performance is based on the

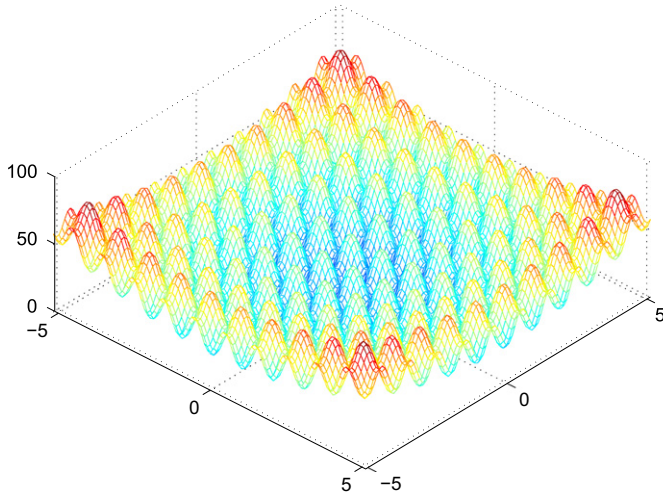


Fig. 1. Two-dimensional Rastrigin function.

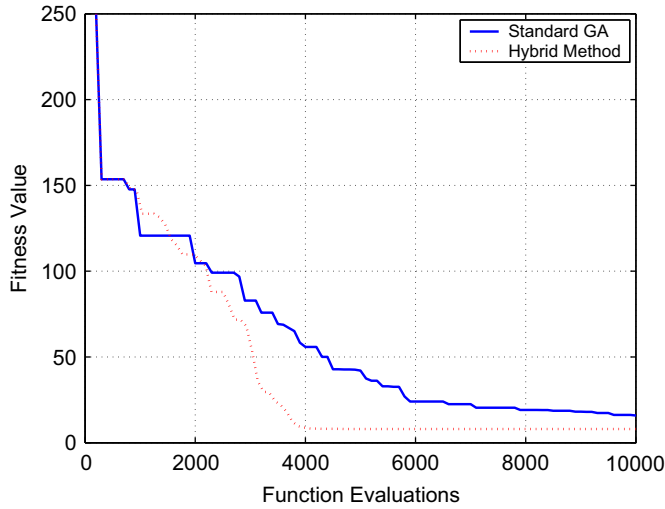


Fig. 2. Convergence history of Rastrigin function.

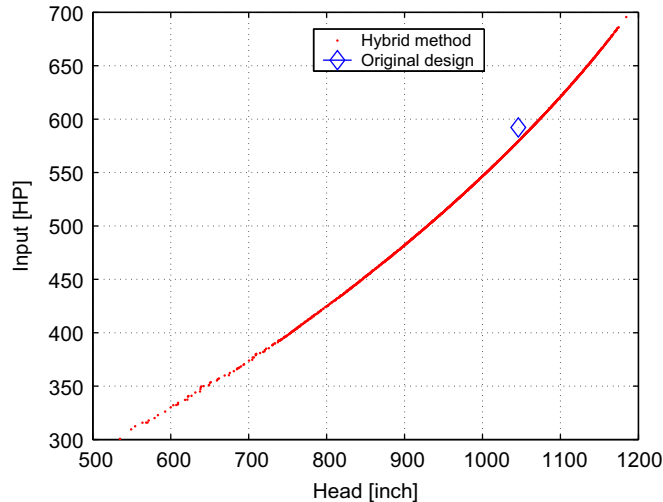


Fig. 3. Objective function values of the centrifugal pump designs.

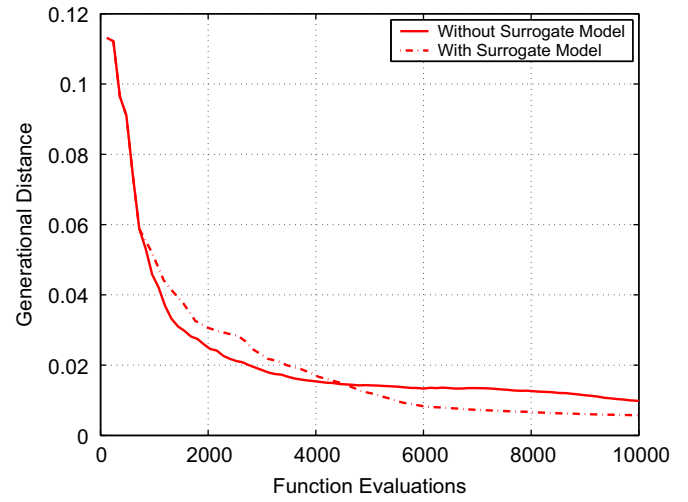


Fig. 4. Generational distance histories.

generational distance G [90]:

$$G = \frac{\sqrt{\sum_{i=1}^q d_i^2}}{q}, \quad (18)$$

where d_i is the distance between a Pareto-optimal solution and its nearest point at the true Pareto-optimal front. Because a true Pareto-optimal front is not available, it is approximated by solution by using a GA with a population size of 400 and generation size of 300.

Fig. 4 depicts the generational distance of the standard GA and the hybrid GA. The surrogate model starts from the sixth generations. The standard GA has a quicker convergence at the early stage and then it slows down after 4000 function evaluations. The hybrid scheme catches up with the pure GA after 4500 function evaluations. The superiority continues until the end. The convergence rate also slows down after about 6000 function evaluations. Other computations with different initial condition also confirm the advantages of our hybrid approach.

2.4. Remarks on hybrid approaches

To certain degree hybrid approaches expedite convergence of the EAs and are more efficient than the pure EAs for certain problems. However, there are some limitations for practical applications. One of the major concern is how to construct a high-fidelity surrogate model with the existing points from the genetic computation. Those points, more often than not, are not well organized and randomly distributed. It is problematic to obtain a good surrogate model sufficiently approximating the true function. In addition, the number of local searches, the number of points used to construct the surrogate model, and the number of surrogate models constructed are empirically chosen. There is no existing rule to follow. The generalization and application of such approaches will be challenged unless the above mentioned questions can be quantified properly. In the next section we will present a more general approach.

3. Optimization using surrogate model

Surrogate models are built to approximate computationally expensive functions; they are computationally orders of magnitude cheaper, while still providing a reasonably accurate approximation to the real functions. Response surface methods (RSMs) are a popular surrogate model in the literature. RSMs have many

favorable characteristics. For example, RSMs produce satisfactory distributions of information throughout the region of interest, ensure the response closely approximates the true function, and provide a simple calculation of parameter coefficients. With response surface approximations, Knill et al. improved a high speed civil transport wing design [40], Madsen et al. optimized a diffuser shape [50], and Keane improved a wing design [38]. Lian and Liou [45] hybridized the desirable characteristics of EAs and response surface approximations to perform MOOPs.

3.1. Response surface methodology

The basic RSM process involves selecting a number of design points at which the computationally expensive function is evaluated. Then, after these points are analyzed, response surfaces are constructed for the functional relationships between the design variables and objective functions. A regular optimization procedure is then applied on the response surfaces to find optimal solutions.

RSMs typically employ low-order polynomial approximations in place of computationally intensive simulations. The second-order model is widely used due to its flexibility and ease of use [56]. A second-order response surface model with d variables can be written as the following:

$$y = \beta_0 + \sum_{i=1}^d \beta_i x_i + \sum_{i=1}^d \beta_{ii} x_i^2 + \sum_{j=2}^d \sum_{i=1}^{j-1} \beta_{ij} x_i x_j + \varepsilon, \quad (19)$$

where x_i are the predictor variables, β are the p regression coefficients, ε denotes the total error, which is the difference between the observed and the estimated response value. The unknown regression coefficients are typically estimated using the method of least squares regression. The fitting model written in matrix form is

$$\hat{\mathbf{y}} = \bar{\mathbf{x}} \bar{\mathbf{b}}, \quad (20)$$

where $\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n]^T$ is the estimated response vector of the n observed response vector $\hat{\mathbf{y}} = [y_1, y_2, \dots, y_n]^T$, $\bar{\mathbf{x}}$ is the matrix of model terms evaluated at the design points, $\bar{\mathbf{b}}$ is the least squares estimator of regression vector $\bar{\mathbf{\beta}}$, which is estimated to minimize the sum of the squares of the error (SS_E) defined as follows:

$$SS_E = \sum_{i=1}^n (y_i - \hat{y}_i)^2. \quad (21)$$

There are several important statistic measures that we will use frequently in this work: the coefficient of determination R^2 statistic, the adjusted R^2 statistic R^2_{adj} and the root mean square error (RMSE). Their definitions involve partitioning the total sum squares $SS_T = \sum_{i=1}^n (y_i - \bar{y})^2$ into a sum of squares due to model and a sum of squares due to error,

$$SS_T = SS_R + SS_E, \quad (22)$$

where $\bar{y} = \sum_{i=1}^n y_i / n$ is the mean of response. The R^2 statistic is defined as the fraction due to model

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_E}{SS_T}, \quad (23)$$

which estimates the proportion of the variation in the response around the mean that can be accounted for by the model. An R^2 of 1 means a perfect model fit ($\hat{y}_i = y_i$); an R^2 of 0 means the fit has a prediction capability no better than the overall response mean. The R^2_{adj} statistic is defined as

$$R^2_{adj} = 1 - \frac{n-1}{n-p} (1 - R^2), \quad (24)$$

which is an improved measure of R^2 . Unlike R^2 , R^2_{adj} does not increase when redundant parameters are added to a response

surface approximation. The RMSE is commonly denoted as σ , which estimates the standard deviation of the random error and can be written as the following form:

$$\sigma = \sqrt{\frac{SS_E}{n-p}}. \quad (25)$$

3.2. Latin hypercube designs

The central composite design (CCD) is a popular approach for practitioners in DOE when the number of design variables are small. When a large number of design variables is involved, alphabetical optimal designs, such as the D -optimal design can be employed. Generally, there are two types of simulation models, stochastic and deterministic. In a stochastic simulation model, the output responses are random variables. The variance of the response measurements becomes a dominant issue for experimental designs constructed for building response surface models. In a deterministic simulation model the output responses are entirely deterministic quantities whose values are determined by the mathematical models and there is no variability in analysis outputs. Sacks et al. [74] found that under such a situation a good experimental design tended to fill the design space rather than to concentrate on the boundary. They stated that standard designs, such as CCD and D -optimality design, which produce good results when measurement variabilities are involved, might be inefficient for deterministic computer codes.

Latin hypercube design (LHD) [52] is a favorite space-filling design and it provides more information within a design space and is suitable for the approximation of computer experiments which mainly have system error rather than random error. LHD is found to be more accurate than random sampling and stratified sampling to estimate the means, variances, and distribution functions of an output. In addition, LHD can handle a large number of design variables and is computationally cheap to generate. The LHD process is relatively straightforward. Suppose we have d design variables $\{x_i\}_1^d$, the range of each design variable (input) x_i is first equally divided into n intervals and one random number (observation) is generated in each interval. So for each input there are n observations. To choose a design point, LHD randomly chooses one observation from the first input x_1 . This observation matches with another randomly chosen observation from the second design variable x_2 . This process continues through the last design variable x_d . Similar processes can be used to choose other design points. This sample process ensures a uniform distribution of design variables at each interval.

In real design, some combinations are not feasible or can even cause crash of the CFD code. When this happens, LHD provides the flexibility to adjust the variable to a certain degree without undermining the property of an LHD sample [92]. There is no comprehensive theory about the number of design points required to construct the response surface models with LHD. Relevant work was conducted by Carpenter [11] and Unal [88] who compared the performance of experimental design methods for building response surface models with deterministic models. Their results showed that the use of 20–50% over-determined D -optimal designs was a good choice.

3.3. The coupled model

The proposed approach by Lian and Liou [45] has the following steps:

1. Sampling design points using Latin hypercube sampling based on the number of design variables.

2. Evaluating the design points using the CFD code.
3. Constructing the surrogate model using the response surface method.
4. Applying the genetic algorithm on the surrogate model and finding the Pareto-optimal solutions.
5. Choosing representative solutions along the Pareto-optimal front obtained from Step 4 using the K-means clustering algorithm (described below) and verifying against the CFD code.

It is noted that the Pareto-optimal solutions based on the computationally cheap surrogate model do not necessarily satisfy the CFD code because they are from the surrogate model. It is necessary to verify these solutions against the CFD code. To verify the whole set of Pareto-optimal solutions can be time demanding. Instead, a subset of the solutions is chosen. For that purpose K-means clustering algorithm can be used to select a subset of representative solutions.

K-means clustering algorithm is a popular clustering technique to find a subset of points which accurately reflect the distribution of the whole data points. It is an improvement on randomly choosing a subset. This algorithm partitions the total N data points $\{\vec{x}\}$ into K disjoint subsets S_j containing N_j data points. We wish to find a set of K representative vectors $\{\vec{\mu}_j\}_1^K$ in such a way as to minimize the sum-of-squares clustering function given by

$$J = \sum_{j=1}^K \sum_{n \in S_j} \|\vec{x}^n - \vec{\mu}_j\|^2, \quad (26)$$

where $\vec{\mu}_j$ is usually chosen as the mean of the data points in set S_j . Detailed description and implementation about K-means clustering are documented by Bishop [8].

3.4. Example: multi-objective design of NASA rotor 67 blade

Lian and Liou [45] applied their approach to the design of the NASA Rotor67 compressor blade. The NASA Rotor67 is a transonic axial-flow fan rotor and is the first stage of a two-stage fan [81]. The test rotor with 22 blades is shown in Fig. 5. The inlet and exit tip diameters are 51.4 and 48.5 cm, respectively. And the inlet and exit hub/tip radius ratios are 0.375 and 0.478, respectively. Based on average span/root axial chord, the rotor has a low-aspect-ratio of 1.56. The solidity, defined as the ratio of chord length to spacing between two adjacent blades, varies from 3.11 at the hub to 1.29 at the tip. At the design point, it has a stage pressure ratio of 1.63 at a mass flow rate of 33.25 kg/s. The design rotational speed is 16,043 rpm, which generates a tip speed of 429 m/s and a relative inlet tip Mach number of 1.38. The Reynolds number based on the chord length at the hub and the inlet velocity is 1.797×10^6 . The blade is made of generic titanium (Ti-6Al-4V), which has a density of 4510 kg/m³.

The design problem is formulated as the following:

$$\begin{aligned} &\text{Maximize } p_{02}(\vec{x})/p_{01}(\vec{x}) \\ &\text{Minimize } W(\vec{x}) \\ &\text{Subject to } |\dot{m} - \dot{m}_{bl}|/\dot{m}_{bl} < 0.5\%, \quad \vec{x}_L \leq \vec{x} \leq \vec{x}_U, \end{aligned}$$

where p_{02} is the outlet total pressure, p_{01} is the inlet total pressure, p_{02}/p_{01} is the stage pressure ratio. Here \dot{m} represents the mass flow rate, and subscript “bl” denotes the baseline design. The mass flow rate and the mass-averaged total pressure are computed with the CFD tool; the blade weight W is computed by integrating the blade volume. Vector \vec{x} represents the design variable, and \vec{x}_L and \vec{x}_U are the lower and upper bounds of the design variables, respectively. The lower and upper bounds are defined as 0.95 and 1.05 of the baseline value, respectively.

The aerodynamic objective is to maximize the stage pressure ratio while the structural objective is to minimize the blade weight. The aerodynamic constraint is imposed to ensure the new design has comparable mass flow rate as the baseline design. It should be noted that the two objectives are competing: improving one objective will deteriorate the other. Instead of having one single optimal solution, the design has a set of compromised solutions among which a solution is better than the others in terms of one objective but is worse in terms of the other.

The designed blade geometry is defined by superimposing a perturbation blade upon the baseline Rotor67 blade. The perturbation blade is obtained by linear interpolation of four airfoil profiles along the span (hub, 31% span, 62% span, and tip). Each airfoil profile can be defined by a mean camber line and thickness distributions, and is parameterized by a third-order B-spline curve. The camber is determined by three design variables (with the leading and trailing edge points fixed) while the thickness distribution by five design variables. As a result, eight design variables are required to represent such a perturbation airfoil profile, resulting in 32 design variables in total. By doing this, it readily recovers the baseline geometry by setting all the design variables to zero. To make the optimization results comparable to those of the baseline configuration, the chord distribution along the span and the meridional contours of the hub, casing, sweep, and lean are fixed.

With 32 design variables in Eq. (19), there are 561 unknown coefficients and 1024 design points are sampled, which represent an 80% over-determined design. These design points are evaluated using the TRAF3D code [3,4] to give the mass flow rate and pressure ratio. The blade weight is computed by integrating the blade volume. Three response surface models are thereafter

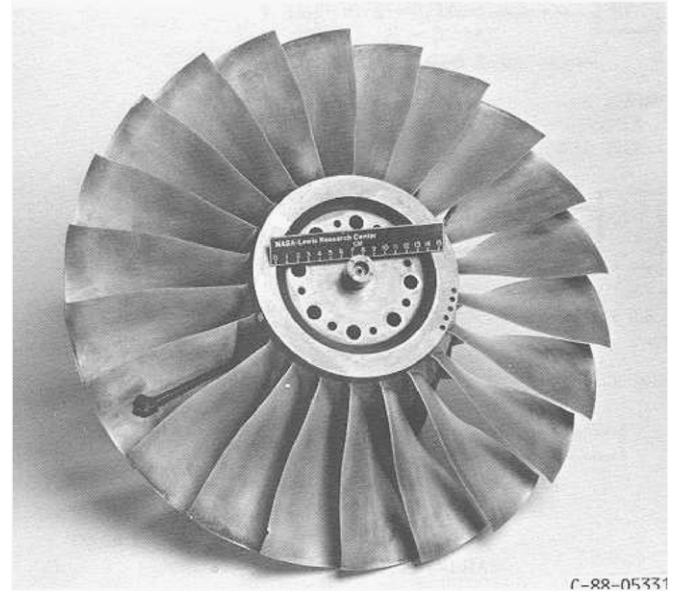


Fig. 5. Test rotor.

Table 1
Statistical measures of the quadratic response surface approximations.

Error statistics	p_{02}/p_{01}	W	\dot{m}
R^2	0.9949	0.9999	0.9979
R^2_{adj}	0.9888	0.9999	0.9954
%RMSE	0.3000e-3	0.1175e-3	0.1270e-3

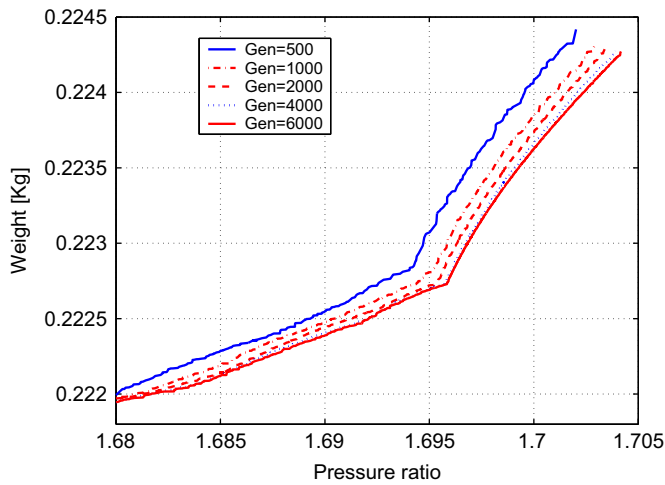


Fig. 6. Convergence history of the genetic algorithm.

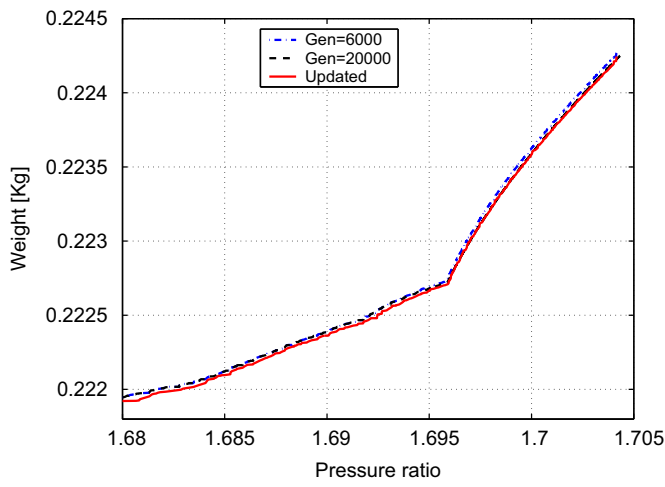


Fig. 7. Updated Pareto-optimal front using local search.

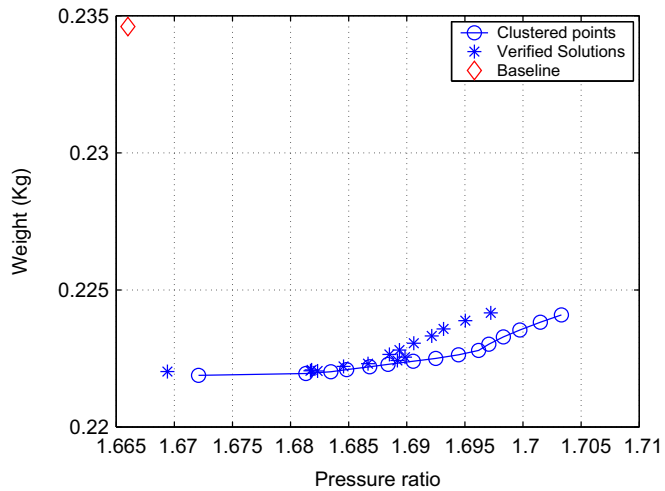


Fig. 8. Pareto-front optimal solutions for the Rotor67.

constructed: two for the objective functions and one for the constraint. The accuracy of these models is evaluated by statistical measures, including the coefficient of determination (R^2), the adjusted coefficient of determination (R^2_{adj}), and the percentage of root mean square error (%RMSE), which is the ratio of root mean square error to the mean of response. The coefficient of determination R^2 measures the proportion of the variation around the mean accounted by the model. It has a range between 0 and 1. A value of 1 means a perfect fit with zero errors. The adjusted coefficient of determination is more comparable over models with different numbers of parameters by using the degrees of freedom in its computation. Unlike R^2 , which always decreases if extra

Table 2
Rotor67 design performance.

Design	Mass flow (kg/s)	Isentropic efficiency	Pressure ratio	Weight ratio
Minimal volume	33.682	0.9166	1.669	0.9464
Maximal pressure ratio	33.355	0.9153	1.697	0.9552
Rotor67	33.446	0.9123	1.666	1.000

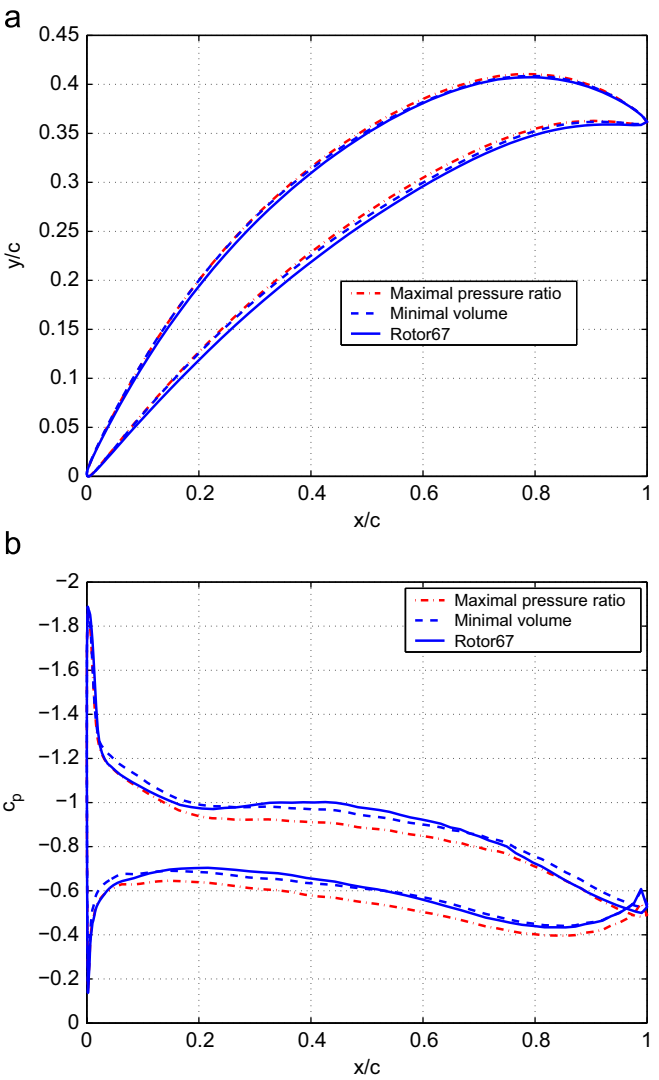


Fig. 9. Comparison between the maximal pressure ratio design and the baseline at the 10% span from the hub. (a) Airfoil shape and (b) Pressure distribution.

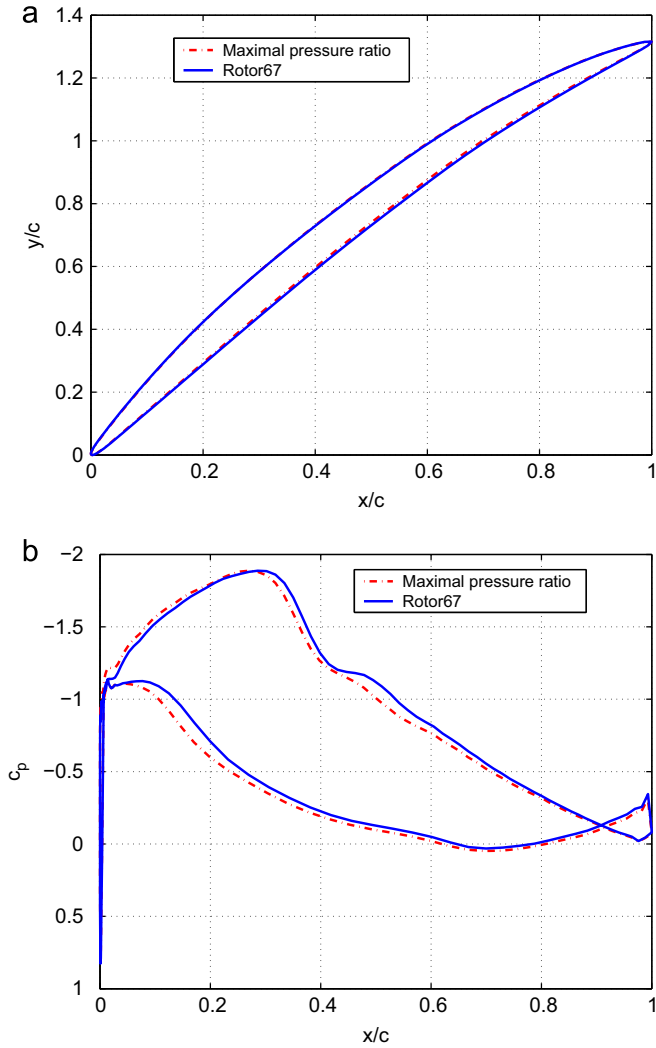


Fig. 10. Comparison between the maximal pressure ratio design and the baseline at the 50% span from the hub. (a) Airfoil shape and (b) Pressure distribution.

terms are added to the model, R_{adj}^2 decreases when redundant terms are added. Table 4 shows the test result. The value of R_{adj}^2 for the stage pressure rise has a value of R_{adj}^2 larger than 0.98, indicating that the response surface model has a good representation of the variability observed from the design points; likewise, the values of R_{adj}^2 for the weight function and constraint function also suggest that the quadratic response approximations have good predictive capabilities (see Table 1).

The population size of the genetic algorithm is set as 320. Fig. 6 shows the convergence history. The optimization result clearly improves as more generations are continued. The convergence rate is fast at the early stage and becomes slow later on. Numerical test shows that an increase in the generation size can improve the convergence further. However, an increase in the generation size is not always a panacea because it will not only increase the computational cost but also raise another question on when to terminate the GA optimization.

A frequently used remedy is to hybridize a GA with a gradient-based method; here a gradient-based method after the GA optimization is switched on. The gradient-based method is the sequential quadratic programming method from commercial software DOT. The starting points of the gradient-based optimization are the Pareto-optimal solutions from the GA optimization at the 6000th generation. The gradient-based method does improve

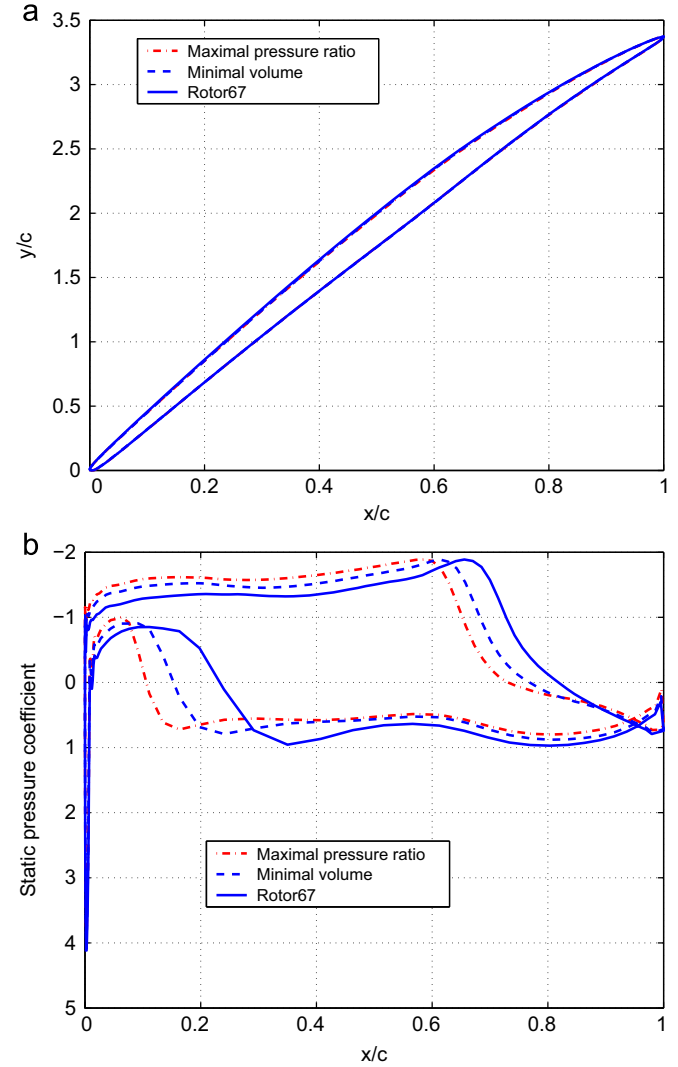


Fig. 11. Comparison between the maximal pressure ratio design and the baseline at the 90% span from the hub. (a) Airfoil shape and (b) Pressure distribution.

the solution, and the improved Pareto-optimal front is shown in Fig. 7. In terms of computational cost, the gradient-based optimization evaluates each response surface for approximately 16,000 times, which is equivalent to the number of functions evaluated by GA in 50 generations. Fig. 7 shows that the improved Pareto-optimal front is better than that from GA optimization using 20,000 generations, even though the improvement may be minor.

The improved Pareto-optimal front has 473 Pareto-optimal solutions. It is time consuming to evaluate all of them. Using the K-means cluster algorithm [8], Lian and Liou [46] chose 16 representative solutions to verify with the CFD code. The validated results together with the baseline design are shown in Fig. 8. Compared with the baseline design, all the solutions increase the stage pressure ratio and reduce the weight. Some Pareto-optimal solutions increase the pressure ratio by as much as 1.8% and reduce the weight by 5.4%.

To understand the physics involved in the shape change, two designs, representing the design with minimal weight and maximal pressure ratio, respectively, were chosen for further analysis. We compare their aerodynamic performance with the baseline design. Table 2 lists their performance characteristics, including the mass flow rate and the isentropic efficiency. The minimal weight design has a larger mass flow rate and higher

isentropic efficiency than the baseline while the high pressure ratio design has a slightly smaller mass flow rate but higher isentropic efficiency than the baseline.

All the new designs in Fig. 8 have a larger pressure ratio and lighter weight than the Rotor67. The design with the maximal pressure ratio will be more meaningful in explaining the efforts of shape change on pressure distribution and flow pattern. Therefore, the pressure distribution and flow pattern between the baseline and the maximal pressure ratio design are compared. Three spanwise blade profiles are shown from Figs. 9 to 11, representing the span of 10%, 50%, and 90%, respectively. The static pressure coefficient profiles are also shown. At the 10% span, the

high pressure ratio design has a larger camber but less thickness than the Rotor67 design. The thinner airfoil contributes to the lighter weight of the new design. The difference in the pressure distribution is rather small. The same conclusion can be made at the 50% span. At the 90% span, the high pressure ratio design has a slightly smaller camber and thinner airfoil than the Rotor67. Nevertheless, the pressure difference is rather large, indicating that transonic flow is sensitive to the shape change. One noticeable change is the shock position. For the new design the shock wave is positioned more forward than the baseline.

Figs. 12–14 present the relative Mach number contours at three spanwise positions, again, at the 10%, 50%, and 90% span,

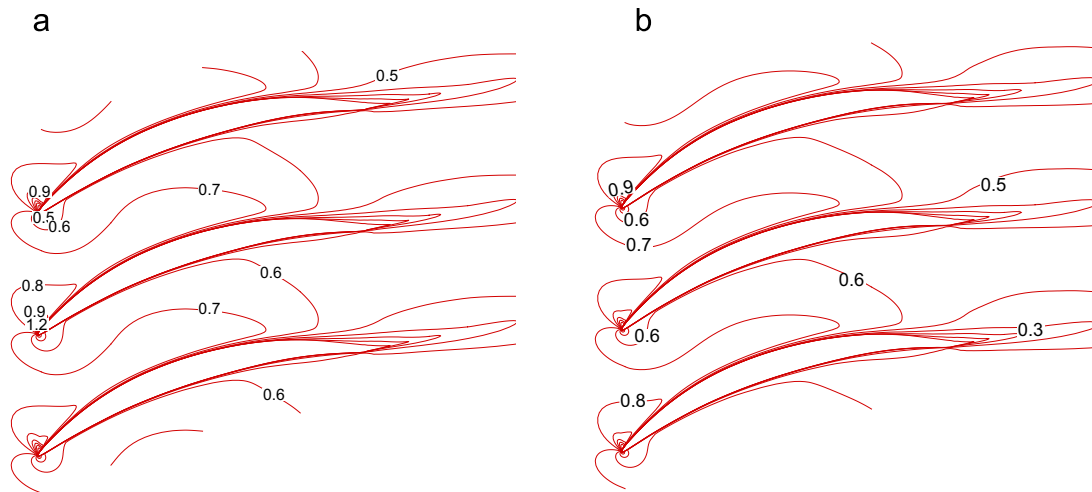


Fig. 12. Relative Mach number contours at the 10% span. (a) Rotor67 and (b) High pressure ratio design.

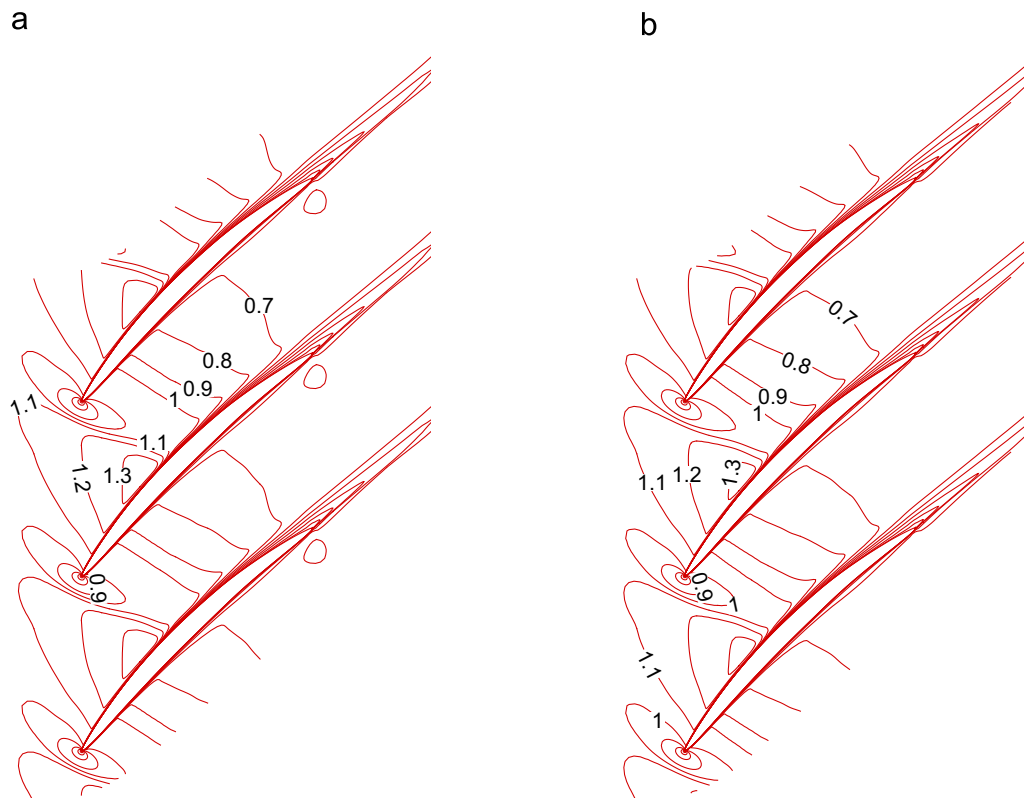


Fig. 13. Relative Mach number contours at the 50% span. (a) Rotor67 and (b) High pressure ratio design.

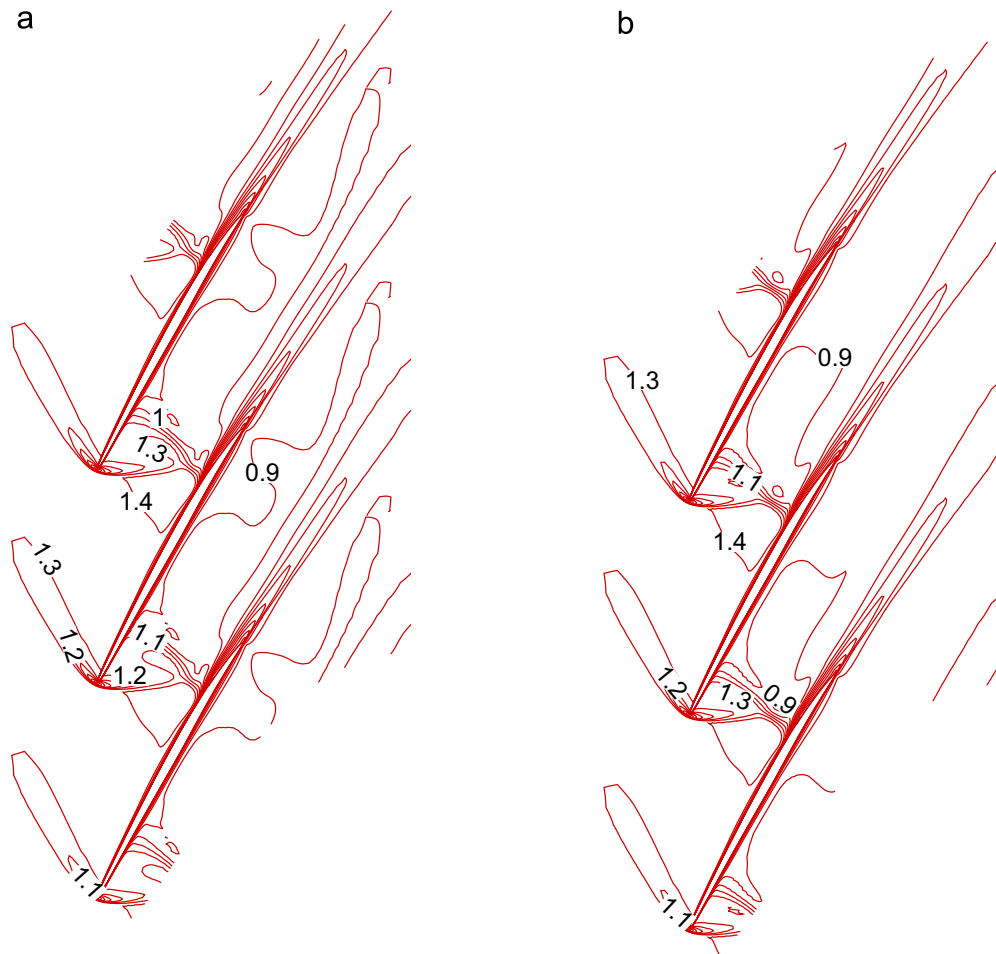


Fig. 14. Relative Mach number contours at the 90% span. (a) Rotor67 and (b) High pressure ratio design.

respectively. At the 10% span, the two contours are similar, except the supersonic bubble is weaker in the new design than in the baseline, the Mach number is 1.21 and 1.3, respectively. At the 50% span an oblique shock stands ahead of the blade. The two shocks have comparable strength. At the 90% span, the shock system has a lambda structure with a bow shock. The relative Mach number of the maximal pressure design is 1.48, which is weaker than 1.52 in the baseline design. The weaker shock is helpful to decrease the global losses.

The pressure distribution on the suction side is used to interpret the flow pattern. Fig. 15 compares the difference between the maximal pressure design and the Rotor67 design. Both designs have a strong passage shock in the upper part of the rotor. In the central part of the blade span, the passage shock loses its intensity and becomes weak. In accordance to Fig. 14 the maximal pressure design has a more forward shock than the baseline. This observation is consistent with our finding from the pressure distribution in Fig. 11. After the shock, flow separates as a result of shock-boundary layer interaction. This separation is evident in Fig. 16. Here, separation lines are characterized by flows going toward the line while reattachment lines look like flows are going away from the line. Compared with the Rotor67, after the shock the maximal pressure design has a smaller separation zone, which is partially responsible for its higher stage pressure ratio. The trailing edge enlargement shows the flow never reattaches after separation for the baseline. These separation zones are also evident in the streamwise velocity contours in Fig. 17, where the

separation zone is indicated with a negative velocity. It is also noticed that in the leading edge close to the hub the low momentum fluid separates and moves radially outwards before it turns in the streamwise direction. Also, close to the hub because of the adverse pressure gradient flow lift-off occurs and eventually flow separation happens near the trailing edge.

Fig. 18(a) shows the performance maps of both the baseline and the maximal pressure ratio design. At the operating condition the new design has a higher total pressure ratio than the baseline. This observation is also true at off-design conditions. Fig. 18(b) shows that the new design has a higher adiabatic efficiency than the baseline at all the operating conditions.

The turn-around time for this optimization is one week using eight Intel Itanium II 1.3GHz processors. With the increase of computational power, it is expected that the turn-around time can be reduced to 24h. Furthermore, to eventually prove that the current procedure works and that CFD is accurate enough to distinguish performance changes with these changes in geometry, it will be of considerable value to build and test the optimized geometry for validating the optimization procedure.

4. Robust and reliability-based design

Reliability-based design optimization (RBDO) is a technique to investigate uncertainties in the design. It provides not only the performance value but also the confidence range. However, when

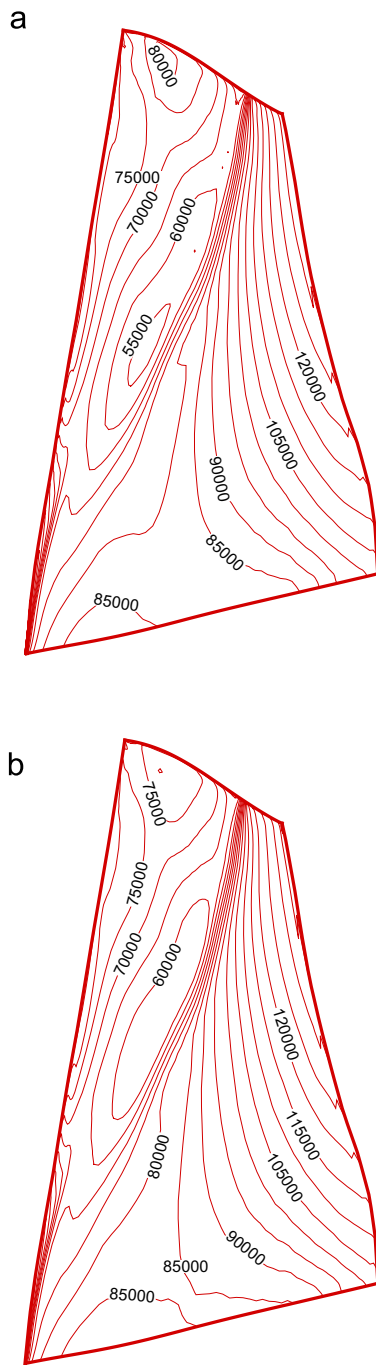


Fig. 15. Pressure contours on the blade suction side. (a) Rotor67 and (b) High pressure ratio design.

computationally demanding models are involved, as commonly encountered in engineering practice, the application of RBDO is limited by the high number of analyses required for uncertainty propagation during the design process. To overcome this limitation, several alternatives with various degrees of complexity, such as moment-based methods [21,87] and the Monte Carlo simulation technique (MCS), have been proposed. Moment based-methods are relatively efficient because they approximate the performance measure at the most probable point using linear or quadratic functions. In general, accuracy is a concern when the performance function exhibits highly nonlinear behavior. It is also pointed out that these methods are not well suited for problems with many competing critical failure modes [70]. Monte Carlo simulation

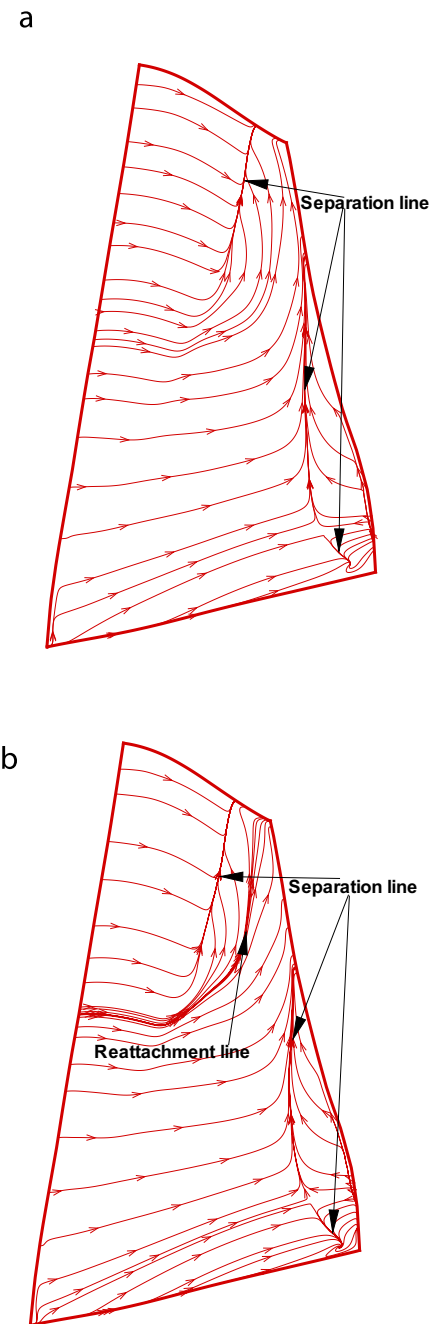


Fig. 16. Streamlines close to the blade suction side. (a) Rotor67 and (b) High pressure ratio design.

technique is a simple form of the basic simulation; it provides a useful tool for evaluating the risk of complex engineering systems. It has been widely used in reliability analysis because of its simplicity and robustness, whereas it requires a large number of analyses for an accurate estimation of the probability of failure especially when the failure probability is small. In addition, Monte Carlo simulation can produce a noisy response [70].

Lian and Kim [41] proposed an approach to tackle the aforementioned challenges stemming from RBDO applications. In their approach, the MCS is used as the probability algorithm; design response surface is adopted to filter the noise from Monte Carlo simulation; response surface approximation also serves to reduce the computational cost involved; probabilistic sufficient

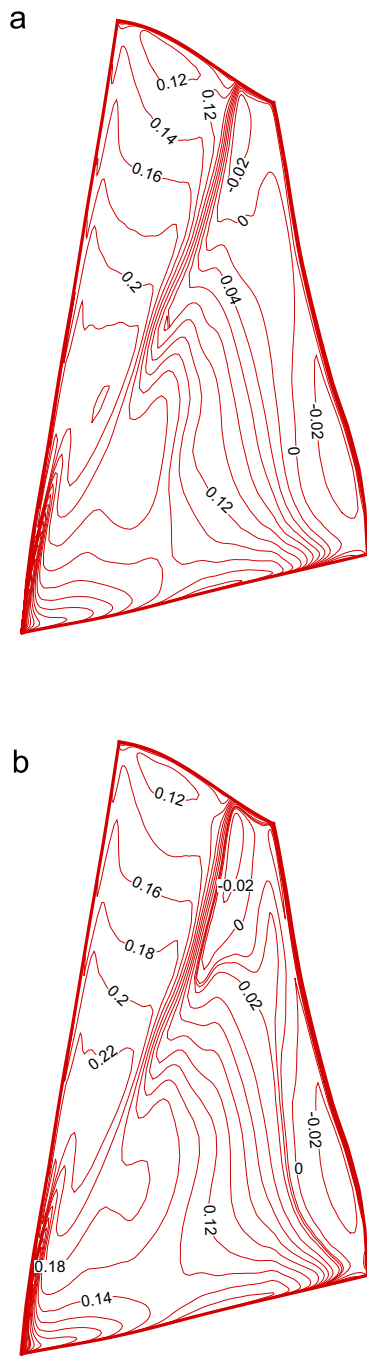


Fig. 17. Streamwise velocity contours close to the blade suction side. (a) Rotor67 and (b) High pressure ratio design.

factor is used in lieu of probability of failure to improve the accuracy in regions of low failure probability and to provide information for the optimization process; a genetic algorithm augmented with a local search is used as the optimization algorithm. This method is employed to perform RBDO of the NASA Rotor67 compressor blade.

4.1. Problem formulation

Lian and Liou [45] performed deterministic design optimization for this blade. Lian and Kim [41] further incorporated uncertainties into the actual design problem and evaluate its

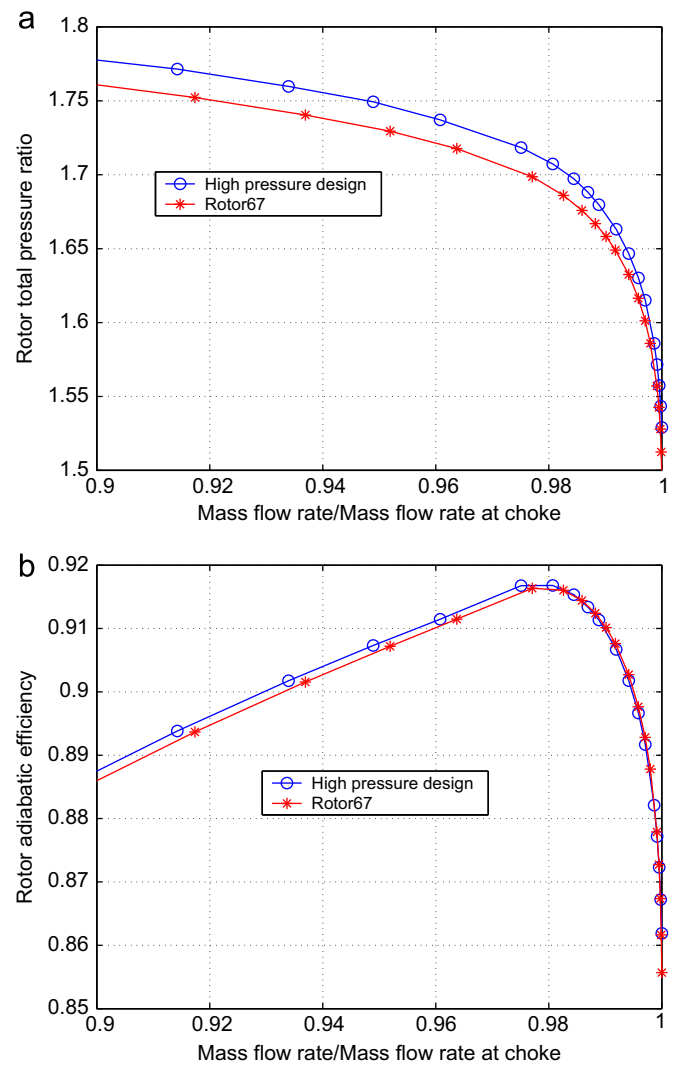


Fig. 18. Compressor blade design speed operating characteristics. (a) Pressure ratio and (b) Adiabatic efficiency.

Table 3
Properties of Ti-6Al-4V, Annealed (genetic).

	Young's modulus (GPa)	Density (kg/m ³)	Poisson's ratio	Endurance limit (MPa)
Mean value	115	4470.5	0.34	547.5
σ	1.33	13.83	0.01	6.17

probability of failure. In engineering design, the probability of failure is usually based on the maximum stress failure criterion, which states that the yielding (failure) occurs when the von Mises stress exceeds the yield strength. The blade is made of generic titanium (Ti-6V-4Al) [81], whose properties are listed in Table 3. For demonstrative purpose, the resistance was chosen as the endurance limit, which is the maximum stress or range of stress that can be repeated indefinitely without failure of the material. Now the failure criterion states that failure occurs when the von Mises stress exceeds the endurance limit. The von Mises stress is computed at the operating condition.

Because the blade has a very good surface finish, the impact of the randomness occurred in shape manufacture can be tightly controlled. Therefore, The design variables which parameterize

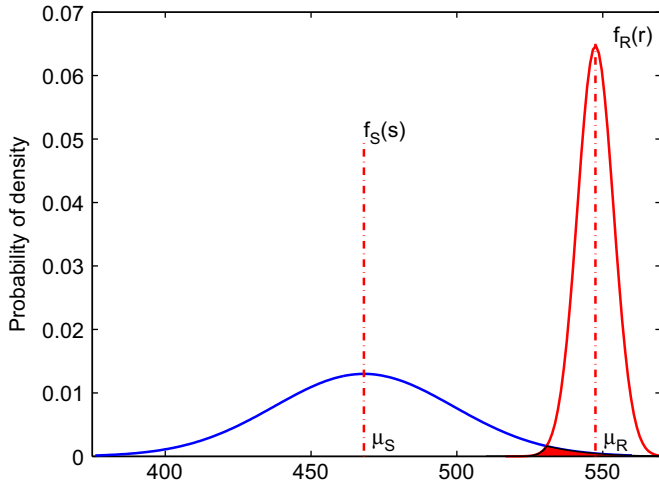


Fig. 19. Distribution functions of the von Mises stress and endurance limit.

the blade geometry are treated as deterministic variables (deterministic design variables). The material density can be measured with good accuracy and it is treated as deterministic too. Young's modulus has a normal distribution, however, it does not change the value of the von Mises stress, and its nominal value was taken in the computation. Therefore, there are two random variables that will be factored into the RBDO: Poisson's ratio and the endurance limit. They are assumed to have a normal distributions around their means. Because a normal distribution, which is varies from $-\infty$ to $+\infty$, lacks a physical interpolation, the random variable is considered as belonging to a range bounded by its mean $\pm 3\sigma$. Here σ is the standard variation. In the presence of uncertainties, the maximal von Mises stress, S , and the endurance limit, R , are random variables in nature. The randomness can be characterized by the probability density functions $f_S(s)$ and $f_R(r)$. The schematic distribution functions of the endurance limit f_R and maximum stress f_S are plotted in Fig. 19.

The multi-objective reliability-based design optimization problem can then be defined as

$$\begin{aligned} &\text{Minimize } W \\ &\text{Maximize } p_{02}/p_{01} \\ &\text{Subject to } P_f(R < S) \leq P_t, \quad |\dot{m} - \dot{m}_b|/\dot{m}_b < 0.05\%, \quad \bar{d}_L \leq \bar{d} \leq \bar{d}_U, \end{aligned} \quad (27)$$

where P_f is the probability of failure, which is based on the maximum stress criterion. The criterion states that failure occurs when the maximum stress S is larger than the resistance R . This optimization is performed under two constraints: the aerodynamic one is to maintain a comparable mass flow rate as the baseline; the structural one is set so that the failure probability, P_f , is less than a target threshold, P_t .

4.2. Monte Carlo simulation

In reliability analysis, the first step is to decide on performance criteria, random parameters, and functional relationships corresponding to each performance criterion. Such a relationship can be written as

$$Z = g(X_1, X_2, \dots, X_n), \quad (28)$$

where Z represents the performance criterion, and X_i is the random variable. The limit state is usually defined as $Z = 0$, which sets the boundary between safe and unsafe regions in the random

variable space. If the failure event is defined as $g < 0$, then the probability of failure P_f can be calculated as

$$P_f = \int \cdots \int_{g < 0} f_X(x_1, x_2, \dots, x_n) dx_1 dx_2, \dots, dx_n, \quad (29)$$

where x_i is the instantiation of X_i , $f_X(x_1, x_2, \dots, x_n)$ is the joint probability density function (PDF) for the input random variables X_1, X_2, \dots, X_n , and the integration is performed over the failure region $g < 0$.

Estimating the probability of failure based on Eq. (29) is involved. First, the joint probability density function does not have an explicit expression and is usually unknown. Second, it requires a multi-dimensional integral over the failure region. When the failure region is an implicit function of the performance criterion, the analytical integration would be very difficult. Moreover, the numerical integration is also impractical for high-dimensional problems. A commonly used simple method in reliability integral is the Monte Carlo simulation technique. The MCS has evolved as a powerful tool for evaluating the reliability of complicated engineering problems. Typically, Monte Carlo simulation technique consists of the following six steps [29]: (1) formulating the problem in terms of all the random variables; (2) qualifying the probabilistic characteristics of each random variable in terms of its PDF; (3) sampling the values of each random variable according to its probabilistic characteristics; (4) evaluating the problem deterministically for each set of realizations of all the random variables; (5) extracting probabilistic information from these simulation cycles by counting the number of failed samples; (6) estimating the accuracy of the simulation.

With all the random variables assumed to be independent, the MCS draws samples of the random variables according to their PDFs and then feeds them into a criterion model to check whether the criterion is satisfied. An estimation of the probability of failure can be expressed as

$$P_f = \frac{N_f}{N}, \quad (30)$$

where N is the total number of simulation cycles and N_f is the number simulation cycles where failure happens. A simulation cycle is defined as solving the problem deterministically for each realization. The accuracy of MCS largely depends on the number of simulation cycles. Its acceptance as an alternative to compute the failure probability mainly depends on its efficiency and accuracy. In general, its estimation accuracy depends on the true probability of failure and the number of simulation cycles. In a 95% confidence interval, the percentage error between the true and estimated probability of failure can be estimated as follows:

$$\varepsilon\% = \sqrt{\frac{(1 - P_f^T)}{N \times P_f^T}} \times 200\%, \quad (31)$$

where P_f^T is the true probability of failure. In our problem we require the probability of failure of the design be less than 10^{-4} . Based on Eq. (31) we can estimate that the percentage error is 20% with 1 million simulation cycles. Therefore, there is 95% probability that the probability of failure estimated with the MCS will fall into the range of $10^{-4} \pm 2 \times 10^{-5}$ with 1 million simulations.

4.3. Probabilistic sufficiency factor

Deterministic designs require a conservative safety margin to ensure the design safety. Usually a nominal safety factor defined in the following is used

$$\text{Nominal } S_F = \frac{R_N}{S_N}, \quad (32)$$

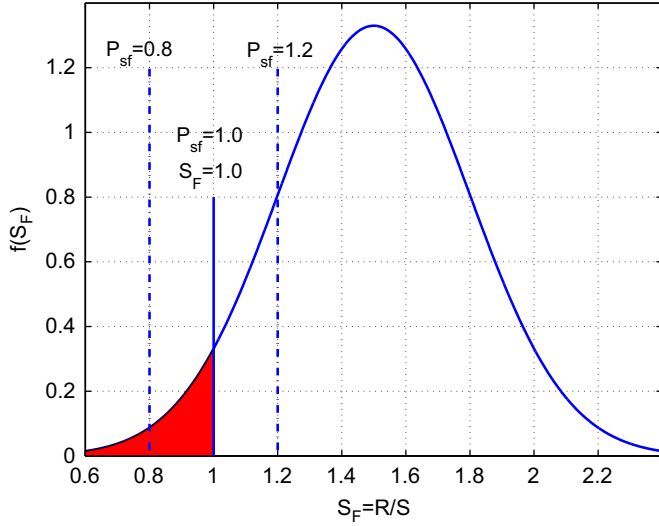


Fig. 20. Probability distribution function of the safety factor.

where S_N and R_N are the deterministic (nominal) values of the von Mises stress and endurance limit, respectively. The S_N is usually below while the R_N is above their respective mean value. The central safety factor, which is the ratio of the mean values of R and S , is also commonly used in deterministic design. However, these measures may be not sufficient to provide information on design reliability. For that reason, in reliability-based design, the reliability can be expressed in terms of the probability of failure

$$P_f(S_F \leq 1) \leq P_t, \quad (33)$$

where S_F is the safety factor defined as the ratio of the resistance to the maximum stress. The concept of probability of failure can be further elaborated graphically. Fig. 20 shows the schematic PDF of the safety factor S_F . The failure criterion states that failure occurs whenever the resistance is less than the maximum stress, i.e., safety factor is less than 1. Thereupon, the probability of failure is represented by the area under the curve left to $S_F = 1$.

Direct use of probability of failure in RBDO problems sometimes may cause problems. For instance, the predicted failure probability can change by several orders of magnitudes. For problems with low failure probability, the MCS predicts zero failure probability, which does not provide useful information in the optimization procedure. In addition, for a fixed number of simulation cycles the error associated with estimated probability of failure from the MCS increases as the probability of failure decreases. Qu and Haftka [69] compared the probability of failure, safety index, and probabilistic sufficiency factor (PSF) in the application to RBDO problems. They found that the PSF did not suffer from accuracy problems in regions of low probability of failure using the MCS and it provided a measure of safety that could be used more readily than the probability of failure or the safety index.

The concept of probabilistic sufficiency factor was introduced by Birger [5]. For an RBDO problem with a known target failure probability P_t , the PSF is the solution to the following equation:

$$P_f(S_F \leq P_{sf}) = P_t. \quad (34)$$

The process of finding the solution of PSF can be regarded as the inverse process of computing the probability of failure, in which we compute the failure of probability by integrating the area under the curve on the left of $S_F = 1$. Here we need to find the value of S_F so that the integrated area under the curve on the left of which is equal to P_t . The concept can be better illustrated using

Fig. 20. If $P_{sf} < 1$, the probability of failure of the design, which is the integrated area on the left of $S_F = 1$, is larger than the target probability of failure, which is the integrated area on the left of P_{sf} , then the design does not meet the safety requirement; if $P_{sf} > 1$, the probability of failure is less than the target one, then the design exceeds the safety requirement; if $P_{sf} = 1$, the design has a probability of failure equal to the target one. Therefore, the following two expressions are mathematically equivalent,

$$1 - P_{sf} \leq 0, \quad (35)$$

$$P_f(S_F \leq 1) \leq P_t. \quad (36)$$

However, Eq. (35) is advantageous in terms of accuracy. Based on these discussion, we can use PSF in our RBDO problem and the problem can be equivalently re-formulated as follows:

$$\begin{aligned} &\text{Minimize } W \\ &\text{Maximize } p_{02}/p_{01} \\ &\text{Subject to } 1 - P_{sf} \leq 0, \end{aligned} \quad (37)$$

The constraints are kept the same as those stated in Eq. (27).

It is simple and straightforward to compute the PSF using the MCS. Suppose we perform N simulation cycles using the MCS around one design point, we compute the safety factor for each cycle, and we sort the safety factor in ascending order and have the following sequence:

$$\{S_{F,1}, S_{F,2}, \dots, S_{F,i}, \dots, S_{F,N}\}, \quad S_{F,i-1} \leq S_{F,i}. \quad (38)$$

If the target failure probability is P_t , then the PSF has the value of $S_{F,i}$, where

$$NP_t - 1 < i \leq NP_t N. \quad (39)$$

4.4. Reliability-based optimization using response surface approximation

The MCS requires a larger amount of function evaluations to obtain a reasonably good accuracy. With the target probability of failure of 10^{-4} , it takes 1 million simulations to ensure the percentage error less than 20%. The use of approximation models is commonly practiced to reduce the computational cost.

Two popular response surface models in reliability-based design are the analysis response surface (ARS) and the design response surface (DRS) [69]. The ARS is fitted to the function in terms of both deterministic variables and random variables while the DRS is fitted to the function exclusively in terms of deterministic variables. In the work of Lian and Kim [41] the ARS is fitted to von Mises stress in terms of the 32 design variables and the random variable Poisson's ratio. At each design point, the probabilistic sufficiency factor is computed by the MCS based on the ARS. The DRS is fitted to the probabilistic sufficiency factor in terms of the 32 deterministic design variables only. The primary purpose of DRS is to filter the noise from Monte Carlo simulation. The objective functions and the structural constraint are also approximated with second-order response surfaces.

4.5. RBDO procedure

The procedure of the RBDO is summarized as follows:

- Sample design points based on both deterministic design variables and random variables with Latin hypercube sampling.
- Evaluate the design points with the high-fidelity analysis tools.
- Construct the ARS model for the maximum stress based on both the deterministic design variables and random variables.

- Perform Monte Carlo simulations based on the ARS to extract the probability sufficient factor.
- Construct the DRS of the objective functions and constraints exclusively based on the deterministic design variables.
- Perform multi-objective optimization using a real-coded genetic algorithm.
- Improve the convergence to the Pareto-optimal front with a gradient-based method.
- Choose representative Pareto-optimal solutions to validate against the high-fidelity tools.

4.6. Numerical results for RBDO

In the problem described in Eq. (27) there are 32 deterministic design variables and two random variables. The objective functions and the aerodynamic constraint therein are only affected by the deterministic design variables while the maximum stress is affected by both the deterministic design variables and one random variable, i.e., Poisson's ratio. The random variable, endurance limit, which is factored into the computation of probability sufficient factor, does not influence the maximum stress. Therefore, the sampling of design points is based on the 32 deterministic design variables and random variable Poisson's ratio. The ARS built for the maximum von Mises stress is therefore has 33 variables and 595 unknown coefficients. With the Latin hypercube sampling we sample 1024 design points, representing a 61% over-determined design. These design points are evaluated using the aforementioned fluid and structure solvers. The accuracy of the response surface approximation is evaluated by statistical measures, including the adjusted coefficient of determination (R_{adj}^2) and the root mean square error (RMSE) predictor. The adjusted coefficient of determination is more comparable over models with different numbers of parameters by using the degrees of freedom in its computation. It measures the proportion of the variation accounted for by fitting means to each factor level. Table 4 shows the test results. The value of R_{adj}^2 for the maximal stress is 0.8369; the stage pressure rise has a value of R_{adj}^2 larger than 0.98 and a %RMSE close to zero, indicating the quadratic response surface model gives accurate representations.

Monte Carlo simulation is performed based on the built ARS. One million simulations are performed at each design point. At each design point the probability of failure and probabilistic sufficiency factor is extracted for the 1024 designs. Fig. 21 shows the distribution of the failure probability. In the plot the design points were sorted according to their probability of failure in ascending sequence. The probability of failure changes several orders of magnitude over a narrow range. A quadratic response surface may not be efficient to capture the change. A high-order response surface model may be required to capture the steep variation. However, it demands more design points to fit the coefficients. In addition, more than 90% of the design has a zero failure probability. Not enough gradient information will be provided in the optimization procedure if a response surface is built based on the failure probability. Even if reliability index is

used, it is still challenging to avoid the large portion of flat region. On the other hand, the distribution of P_{sf} in Fig. 22 shows a smooth variation. For comparison purpose the DRS was constructed for both the probability of failure and probability sufficient factor. The statistical measures are shown in Table 4. It is noticed that the fitting of the failure probability is poor in terms of the statistical measures. However, the DRS of the probability sufficient factor has good statistical measures. The values of R_{adj}^2 and %RMSE are 0.9994 and 0.002337, respectively.

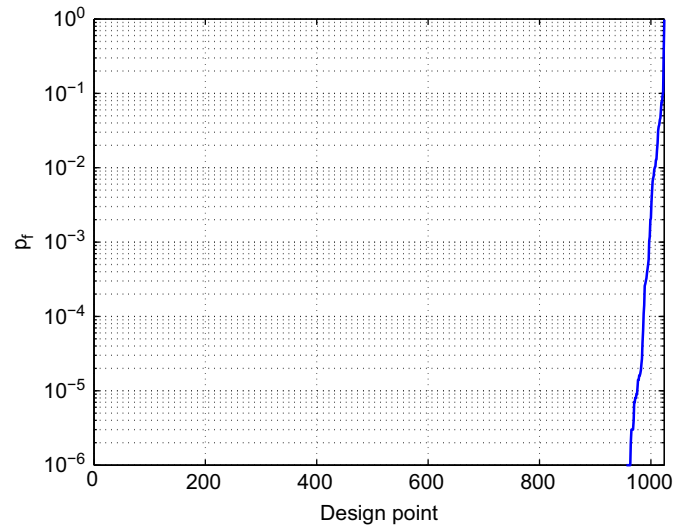


Fig. 21. Distribution of probability of failure of the 1024 design points.

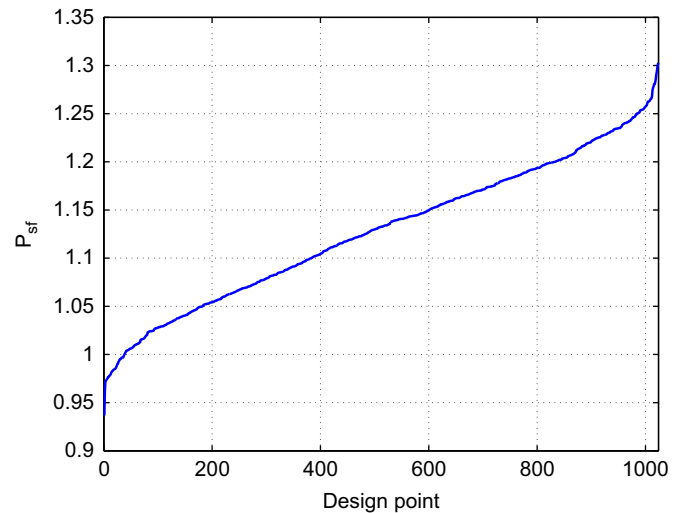


Fig. 22. Distribution of probability sufficient failure of the 1024 design points.

Table 4

Statistical measures of the quadratic response surface approximations.

Error statistics	p_{02}/p_{01}	W	m	S_N	P_{sf}	P_f
R^2	0.9949	0.9999	0.9979	0.9262	0.9994	0.6638
R_{adj}^2	0.9888	0.9999	0.9954	0.8369	0.9987	0.2572
RMSE	0.564e-3	0.800e-5	0.4246e-2	0.1282e-8	0.2637e-2	0.2851e-1
%RMSE	0.3000e-3	0.1175e-3	0.1270e-3	0.2761e-1	0.2337e-2	0.1425e-3

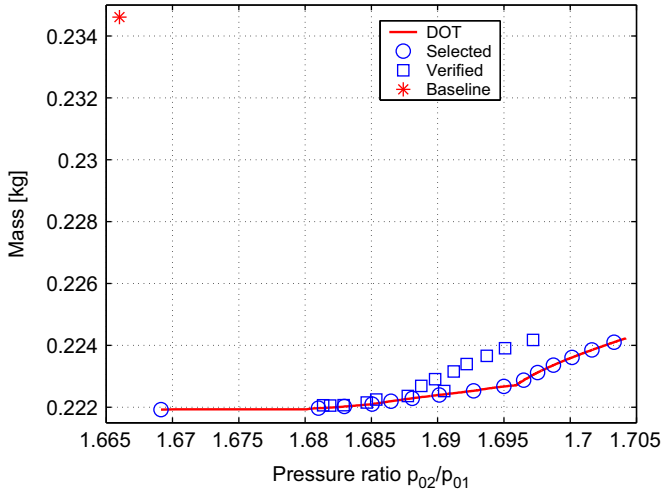


Fig. 23. Comparison of baseline with optimal solutions.

Fifteen representative optimal design points were chosen from the 693 Pareto-optimal solution on the Pareto-optimal front using the K-means clustering algorithm to verify against the high-fidelity analysis tools. The distribution of the selected data points is shown in Fig. 23. Clearly the optimization process decreases the blade weight and increases the stage pressure ratio as well.

4.7. Error analysis

The error of the computed PSF in Fig. 21 stems from two primary sources: the approximation of maximum stress with ARS and the limit size of simulations in the MCS. For the studied problem with a target probability of failure of 10^{-4} and 1 million simulations in the MCS, the percentage error in P_f associated with the MCS is 20%. In modeling the maximum stress with ARS, the %RMSE is $\alpha = 0.02761$. If the maximum stress based on the ARS is S , then the actual maximum stress is estimated in the range of $S(1 \pm \alpha)$. Here it should be noted that the use of RMSE% is not perfectly fitting at this circumstance because the %RMSE is calculated based on the points used to constructed the response surface model and the prediction error for new point could be larger than %RMSE. If the endurance limit takes another random value, R , and the safety factor is computed based on Eq. (32), then the actual safety factor is in the range of

$$\frac{R}{S} \frac{1}{1 \pm \alpha} \quad (40)$$

Plugging $\alpha = 0.02761$ into the above formulation, it is estimated that the uncertainty in the safety factor (also in PSF) is larger than 0.02. Hence the relative error of the probability failure associated with the ARS is 20,000%, which is much larger than the relative error associated with the MCS.

Because it will be time consuming to compute the PSF directly from the structural analysis, the surrogate model and the MCS are used to evaluate it instead. Therefore, the inaccuracy arise from three sources: the ARS approximation, the DRS approximation, and the limit size of the MCS.

Lian and Kim [41] further validated the PSF for each representative optimal design. They computed the PSF using the MCS. By doing this the safety factor R/S was evaluated using the MCS at the optimal design point. The value of the endurance R can be directly obtained based on the distribution function. For a fix blade shape, the maximal stress S for a specific design is only a function of Poisson's ratio. Their discussion indicates that with a

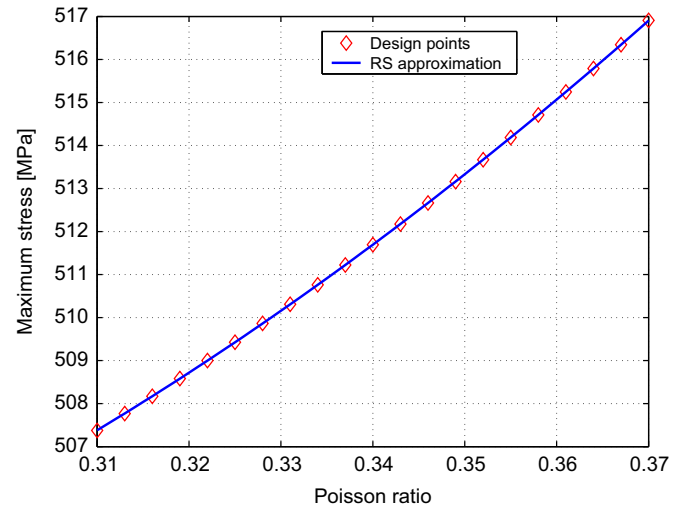


Fig. 24. Approximation the maximum stress at one specific optimal design with quadratic response surface.

target probability of failure of 10^{-4} one needs one million simulations to ensure that the relative error of the failure probability is less than 20%. Even though the structural analysis is relatively cheaper than the CFD analysis, it is computationally formidable to perform that many simulations exclusively based on structural analysis. On the other hand, it is not a sound idea either to evaluate the PSF by using the previously constructed ARS model for the maximum stress. What they did was to construct a new response surface model to approximate the response of the maximal stress to Poisson's ratio at one selected optimal design point.

At each design points they uniformly sampled 21 points for Poisson's ratio. These design points are then evaluated with the structural solver. Their statistical analysis shows that the second-order response surface gives an accurate representation of the relationship between Poisson's ratio and the maximum stress. The value of R^2_{adj} is 1.00 and the %RMSE is 4×10^{-6} . Based on Eq. (40) they computed the actual range of the PSF. A representative quadratic fitting and the sampled points are shown in Fig. 24. The optimal design is then substituted into the newly constructed response surface to evaluate its PSF using the MCS. This calculated PSF is compared with that predicted values in the optimization process. The comparison is illustrated in Fig. 25. They further adjusted the PSF value based on Eq. (40). All the PSF are larger than 1.0, indicating the safety is beyond our target.

5. Data mining of Pareto-optimal solutions

In this section data mining approaches used for analysis of the Pareto-optimal solutions are first discussed. Then an aerodynamic flapping airfoil design is presented as an example of design knowledge extraction from Pareto-optimal solutions using data mining methods. Finally, a new data mining approach using proper orthogonal decomposition is presented.

5.1. Overview

A multi-objective optimization problem is traditionally converted into a single-objective optimization problem and an optimum solution of the converted optimization problem is obtained. When an optimization problem of a real-world design is formulated, many models are introduced and many assumptions are made. Therefore it is not useful for industrial designs to

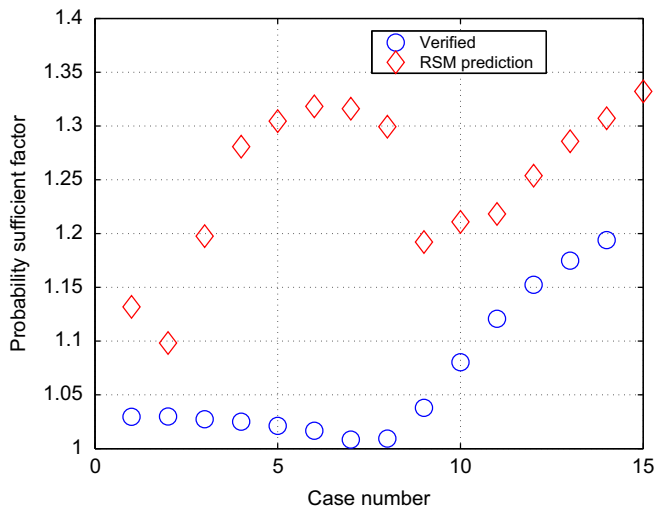


Fig. 25. Verified probability sufficient factor at the optimal design and that from DRS prediction.

get an exactly optimal solution. For example, one can obtain a wing shape that minimizes its aerodynamic drag at a given flight condition by using an optimization method. However, one should modify the optimized wing shape to account for changes in the flight condition such as Reynolds number change, Mach number change, angle of attack change, and so on. One should also modify the optimized wing shape to attach nacelle and to install landing gear, flap, slat, aileron, and so on.

By obtaining Pareto-optimal solutions of a multi-objective problem instead of obtaining one solution of the converted single-objective optimization problem, one can obtain useful design information such as which objective functions are contradicting/independent, which design parameter is sensitive/insensitive to the objective functions, which design parameters are dependent/independent, and so on. Such information is useful for designers because it helps the designers to design and develop real-world products which can not be obtained by solving a design optimization problem.

A multiobjective aerodynamic shape optimization usually results in hundreds (sometimes thousands) of Pareto-optimal solutions each of which has multiple objective function values, constraint function values, and design parameter values, shape, and flow data. Thanks to rapid improvement in computational speed, the number of Pareto-optimal solutions we handle is increasing. One of the hot research topics in multiobjective shape optimization is, thus, how to understand the hundreds of Pareto-optimal solutions and obtain useful knowledge from them.

Data visualization using graphical methods such as scatter-plot matrix [14,85], value path method [25], and star coordinate method [51] are traditionally used to understand the obtained Pareto-optimal solutions. Fig. 26 is an example of the scatter plot matrix. The scatter plot matrix contains all the pairwise scatter plots of the objectives and design parameters in a matrix format. In this example, the objective B depends on the design variable 1 while the objective A does not. This figure also shows that there is a tradeoff between maximization of the objective A and minimization of the objective B. However, each scatter plot becomes too small to analyze when the number of objectives and/or design parameters is large. However, usage of such methods is limited to analysis of small number of Pareto-optimal solutions or small-size design optimization problems that have a few objective functions, constraints, and design parameters.

Multi-objective design exploration (MODE) [35] proposed by Jeong et al. is a framework to extract design knowledge from the

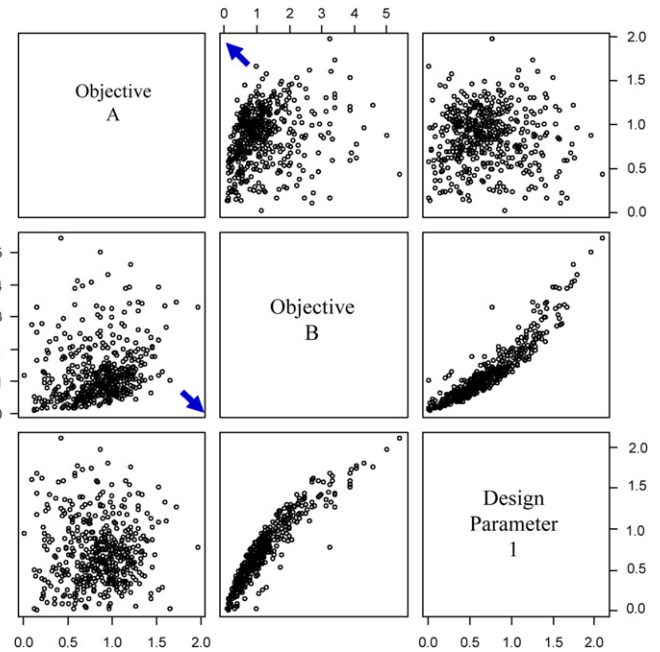


Fig. 26. An example of the scatter plot matrix.

obtained Pareto-optimal solutions such as tradeoff information between contradicting objectives and sensitivity of each design parameter to the objectives. In the framework of MODE, Pareto-optimal solutions are obtained by multi-objective evolutionary algorithm and knowledge is extracted by analyzing the design parameter values and the objective function values of the obtained Pareto-optimal solutions using so-called data mining approaches such as self-organizing map (SOM) [39] and analysis of variance [18] (see flow chart of MODE in Fig. 27). They also propose to use rough set theory [82,83] to obtain rules from the Pareto-optimal solutions. MODE has been applied to a wide variety of design optimization problems including multidisciplinary design of a regional-jet wing [57], aerodynamic design of fly-back booster of reusable launch vehicle design [58], aerodynamic flapping airfoil design [66], aerodynamic turbine blade design for a rocket engine [84], aerodynamic design of a motor blower for a vacuum cleaner [82], and multidisciplinary design of a centrifugal fan for a washer-dryer [83] to get some practically import design knowledge.

Yamashiro et al. [94] proposed an approach to understand Pareto-optimal solutions of many-objective design optimization problems. Their method bases on multi-objective optimization using multi-objective evolutionary algorithm, and data mining using fuzzy c-means (FCM), and fuzzy multiple discriminant analysis (FMDA). The flow of their approach is as follows:

Step 1: Generating Pareto-optimal solutions by a multiobjective evolutionary algorithm.

Step 2: Setting the level of importance (weight) for each objective function.

Step 3: Multiplication of each fitness value by the weight.

Step 4: Clustering Pareto-optimal solutions in multi-dimensional (fitness) space by FCM.

Step 5: Definition of the meanings of a projection axis using labels with variable selection.

Step 6: Application of the FMDA to the selected objective functions for each axis.

Step 7: Identification of projection axes and reduction of dimensionality using each 1st discriminant axis by FMDA.

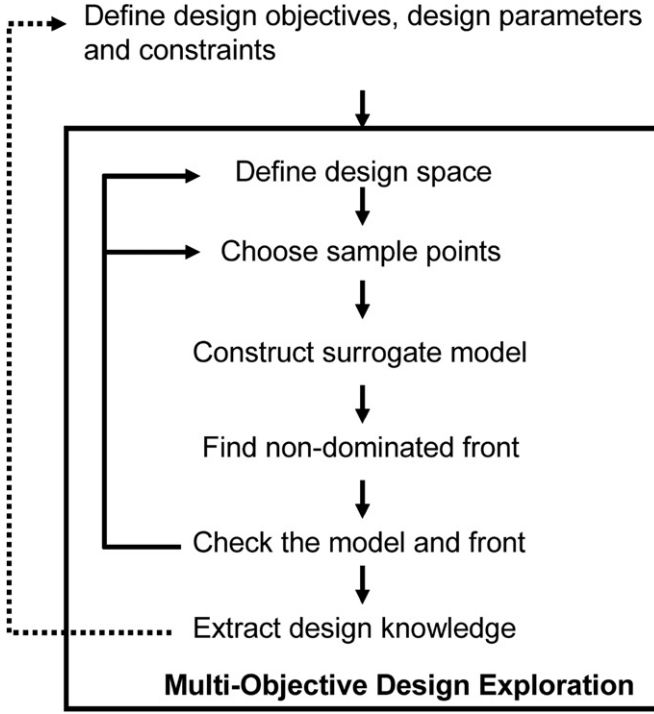


Fig. 27. Flow chart of MODE proposed by Jeong et al. [35].

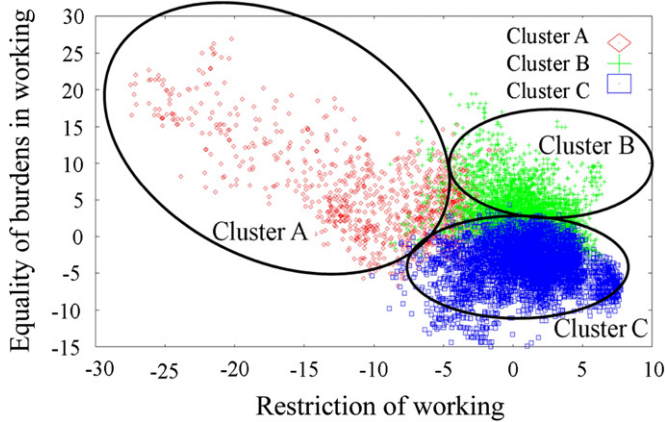


Fig. 28. Visualization of the optimized schedules [94].

Step 8: Projection and visualization of Pareto-optimal solutions onto visible space consisting of two/three identified projection axes.

They applied the proposed method to analyze 11,529 Pareto-optimal solutions of a nurse scheduling problem with 12 objective functions. The 12 objective functions were composed of tree indices reflecting user's intention and a good family of schedules was identified in terms of restriction of working and equality of burdens in working (Fig. 28).

5.2. Data mining of the Pareto-optimal solutions of an aerodynamic flapping airfoil design

The objective of this optimization is to extract aerodynamic knowledge on a flapping airfoil such as (1) tradeoff information between lift, thrust, and required power, (2) effect of flapping motion parameters such as plunge amplitude and frequency, pitching angle amplitude and offset, and phase difference on the

objective functions, and to create guidelines for the design of flapping motion for lift maximization, thrust maximization, and required power minimization.

5.2.1. Design optimization problem

Entomopter, which is an MAV discussed in the United States for future Mars exploration [15,54], is considered. Entomopter has flapping wing system intending higher lift in extremely low atmospheric density at Mars surface (1/70 that at Earth surface) and take off, landing, and hovering capabilities. This MAV has a span length of 1 m and chord length of 0.1 m. The wing airfoil is thin with moderate camber and a sharp leading edge to enhance vortex generation. Its cruising speed is more than 10 km/h and flight time of typical mission is 12 min. For a typical air density of 0.0118 kg/m^3 and air viscosity of $1.36 \times 10^{-5} \text{ Pa s}$, the cruising Reynolds number based on the flight speed and reference length of the chord is assumed to be 1000. It is estimated that the wing of Entomopter will produce lift coefficients between 7.95 and 10.6 with the help of boundary layer blowing. It should be noted that for the purpose of analysis the lift coefficients indicated in Ref. [15] were accepted without questioning their realizability. A recent paper by Kaya et al. [37] showed a flapping wing generates much lower lift coefficient. The results are applicable to low speed MAV on the earth because the Reynolds number is only the non-dimensional parameter that represents Mars atmosphere in this study.

As a first step of understanding flapping wing mechanism, flapping airfoil is considered in this study. The objectives of the present design optimization problem are maximization of the time-averaged lift and thrust coefficients and minimization of the time-averaged required power coefficient at its cruising condition, where the lift, thrust, and required power are averaged over one flapping cycle. Constraints are applied on averaged lift and thrust coefficients so that they are positive. The airfoil is assumed to be NACA 0002 airfoil. The flapping motion of the airfoil (see Fig. 29) is expressed by plunging and pitching motions as

$$x(t) = t, \quad (41)$$

$$y(t) = h \sin(kt), \quad (42)$$

$$\alpha(t) = \alpha_1 \sin(kt + \phi) + \alpha_0, \quad (43)$$

where design parameters are plunge amplitude h , reduced frequency k , pitch offset α_0 , pitch amplitude α_1 , and phase shift ϕ .

5.2.2. Aerodynamic force evaluation

The two-dimensional incompressible Navier–Stokes equations and the continuity equation on the generalized curvilinear coordinates are solved using pseudo-compressible flow simulation approach [13]. The dual-time stepping procedure [34], which allows an implicit method to be used in real time with the updated solution obtained through sub-iterations in pseudo-time, is employed. The numerical fluxes are evaluated by the Roe

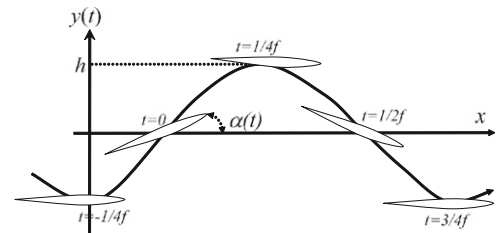


Fig. 29. Parameterization of flapping motion.

scheme [73] where physical properties at the grid interface are evaluated by the MUSCL interpolation [89] based on primitive variables. The viscous terms are evaluated by second-order central differencing scheme. Lower-upper symmetric Gauss–Seidel (LU-SGS) factorization implicit algorithm [95] is used for the time integration. Note that the original version of the program for compressible flow analysis has been used for a wide variety of CFD studies [33,36,24].

The freestream flow conditions are applied to the inflow boundary. The outer boundary of the computational domain is placed at 20 chord length away from the leading edge of the airfoil. On the outflow boundary, pressure is fixed to the freestream value while the other physical properties are extrapolated from the corresponding interior grid points. The airfoil surface is treated as non-slip wall boundary. Physical properties on the wake boundary are interpolated from the adjacent grid points. The initial condition is the uniform flow.

To compute aerodynamic performance of flapping motions, three cycles are simulated and averaged lift and thrust coefficients and required power are obtained for the third flapping cycle. The corresponding computational grid is a C-type grid that has 201 (chordwise direction) \times 101 (normal direction) grid size. Number of grid points on the airfoil is 151. Minimum spacing near the wall

is 0.0001. Number of time steps for each flapping cycle is 1700. To confirm grid and time step size convergence, four computational results with different grid size and number of time step in each flapping cycle is compared.

5.2.3. Data mining

Here, Kohonen's self-organizing map (SOM) is used to analyze the Pareto-optimal solutions. SOM is an artificial neural network where all the solutions are aligned on a grid according to Kohonen algorithm so that neighboring nodes are similar to each other. Mostly, SOM is used for nonlinear projection of input data in three or higher dimensional space onto two-dimensional space to extract knowledge implicit in data such as attributes and features.

A software package called Viscovery SOMine plus 4.0 produced by Eudaptics GmbH is used. 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, which is a more robust approach due to its mediation over a large number of learning steps.

The Pareto-optimal solutions distributed in the present three-dimensional objective function space (CL maximization, CT maximization, and CPR minimization) are mapped into nodes on a two-dimensional grid according to the similarity in terms of the objective function values. Here, map size is 51×41 (2070 nodes) and number of training is 45 with tension of 0.5. These values are automatically determined by the SOMine. It should be noted that direction and Euclidean distance in the objective function space are lost on the SOM. Then the two-dimensional map colored according to each objective function, each design parameter, propulsion efficiency, and Strouhal number are compared for the knowledge acquisition from the present problem.

5.2.4. Results and discussion

The Pareto-optimal flapping motions (shown by spheres) and all the other solutions (shown by circles) obtained by the present optimization are plotted in the three-dimensional objective function space (Fig. 30). Distribution of the Pareto-optimal solutions visualizes tradeoff among the maximization of the averaged lift and thrust and minimization of the averaged required power while it is very difficult to understand effect of each design parameter on the tradeoff in the three-dimensional scatter plot.

Fig. 31 is the obtained map using SOM where each node is colored according to each objective function value. It should be noted that as the number of Pareto-optimal solutions is 560 and the number of nodes of SOM is 2070, each Pareto-optimal solution

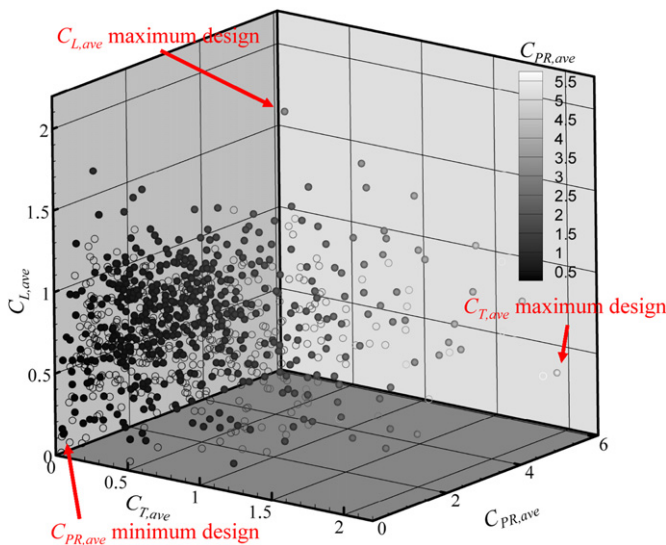


Fig. 30. Evaluated flapping motions (circles) and obtained Pareto-optimal solutions (spheres) plotted in the objective function space.

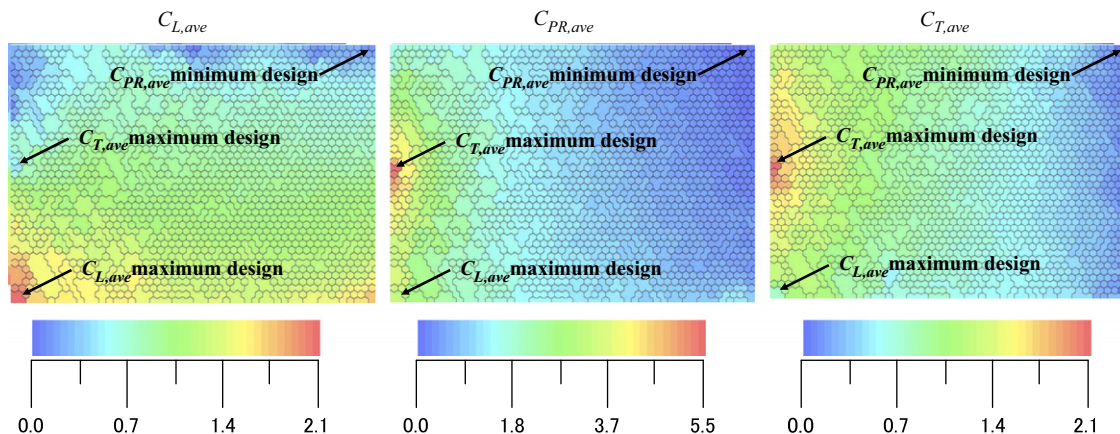


Fig. 31. SOM colored according to each objective function.

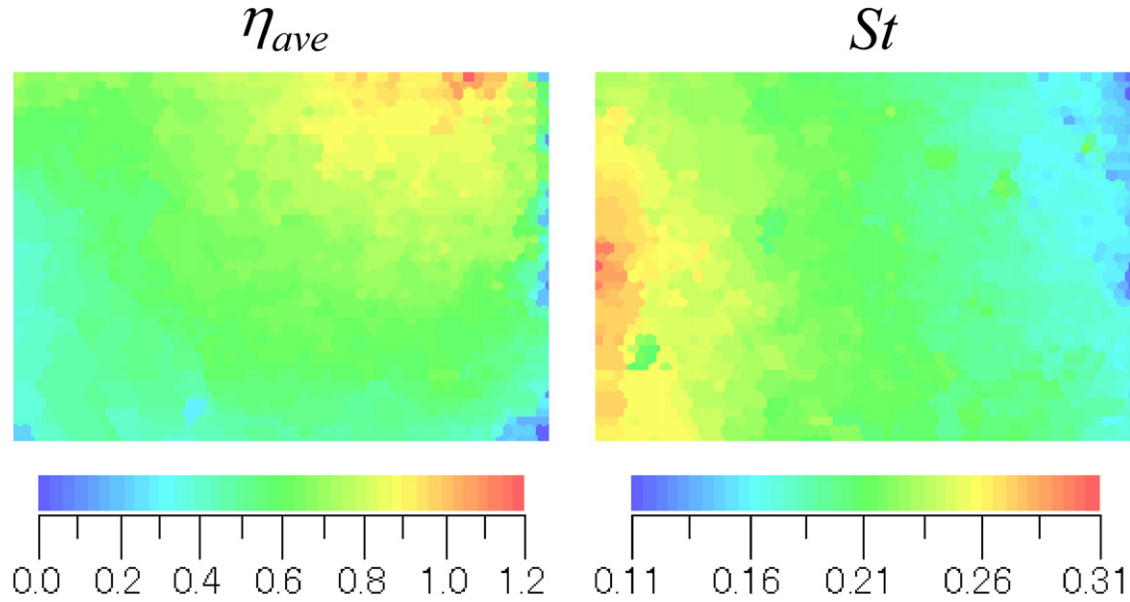


Fig. 32. SOM colored according to propulsive efficiency and Strouhal number.

is represented by one or more nodes on the map as shown in the figure. For example, four nodes located at the lower left corner represent $C_{L,ave}$ maximum design while one node located at the upper right corner represents $C_{PR,ave}$ minimum design. This figure shows flapping motions for smaller required power are mapped on the right side of the map. Flapping motions for larger lift are mapped on the lower left and lower right corners where flapping motions mapped on the lower left corner require large power while those mapped on the lower right corner require smaller power. The flapping motions for larger thrust are mapped on the left hand side. These results indicate the tradeoff between the three objectives exists and thus there is no solution that optimizes all three objectives simultaneously. This figure also indicates that maximizing thrust requires more power than maximizing lift.

The same map colored according to propulsive efficiency and Strouhal number is presented in Fig. 32. According to the research by Taylor et al. [86], flying animals such as birds, bats, and insects in cruise flight operate within a narrow range of Strouhal number between 0.2 and 0.4. Also, Young [96] demonstrated some Navier–Stokes computations to show propulsive efficiency has a peak around a Strouhal number of 0.2. Strouhal number of the obtained flapping motions is consistent with these results.

Distribution of propulsive efficiency shows that flapping motions for maximum propulsive efficiency have small lift. This result is understandable because generation of lift does not contribute to the propulsive efficiency, which is determined by thrust divided by the required power. The propulsive efficiency was maximized at a certain point between maximum of thrust and minimization of required power.

The same SOM colored according to each design parameter value is presented in Fig. 33. Color range of the map corresponds to the present design range. Comparison between Figs. 31 and 33 gives additional knowledge on the present design optimization problem:

1. Phase shift between plunging and pitch angle cycles of the obtained Pareto-optimal solutions are almost 90° . This result is consistent with previous researches on flapping motion such as Ref. [32] where efficiency became high when pitch leads plunging by about 90° .

2. Pitch angle offset of most Pareto-optimal flapping motions is almost zero except for the flapping motions for high lift. This is understandable as the thrust maximum and required power minimum flapping motion is symmetric while lift maximum flapping motion generates lift only in downstroke.
3. Reduced frequency seems to be a tradeoff parameter between minimization of required power and maximization of lift or thrust where smaller frequency leads to smaller required power.
4. Plunge amplitude of most Pareto-optimal flapping motions reached the upper limit of the present design space. This fact indicates that larger plunge amplitude is preferable when two-dimensional flow is assumed. However, in real flapping wing design, the plunge amplitude is restricted by span length and angle of the flapping wing along the flap arc.
5. Pitch angle amplitude of the most Pareto-optimal solutions distributes between 35° and 45° , which indicates that certain level of pitch angle amplitude is optimum for high performance flapping motion. This figure also indicates that better solutions may have been found if the search space was wider since 45° is upper limit of the present search space of 1.

5.3. Data mining of Pareto-optimal solutions using proper orthogonal decomposition

Analysis of objective function values and design parameter values of Pareto-optimal solutions using data mining approach is useful as it can extract essential knowledge of a multi-objective optimization problem such as tradeoff information between contradicting objectives and effect of each design parameter on the objectives. However, to understand physics behind the design problem, data mining of objective function values and design parameters values is not sufficient. For example, if one analyzes only design parameters of a transonic airfoil, he/she may not be able to understand relation between generation of shock wave and aerodynamic characteristics. Recently, Oyama et al. proposed a new approach to extract useful design information from line, face, or volume data of Pareto-optimal solutions of optimization problems [68]. In this approach, the data of Pareto-optimal solutions are analyzed by proper orthogonal decomposition (POD), which is a statistical approach that can

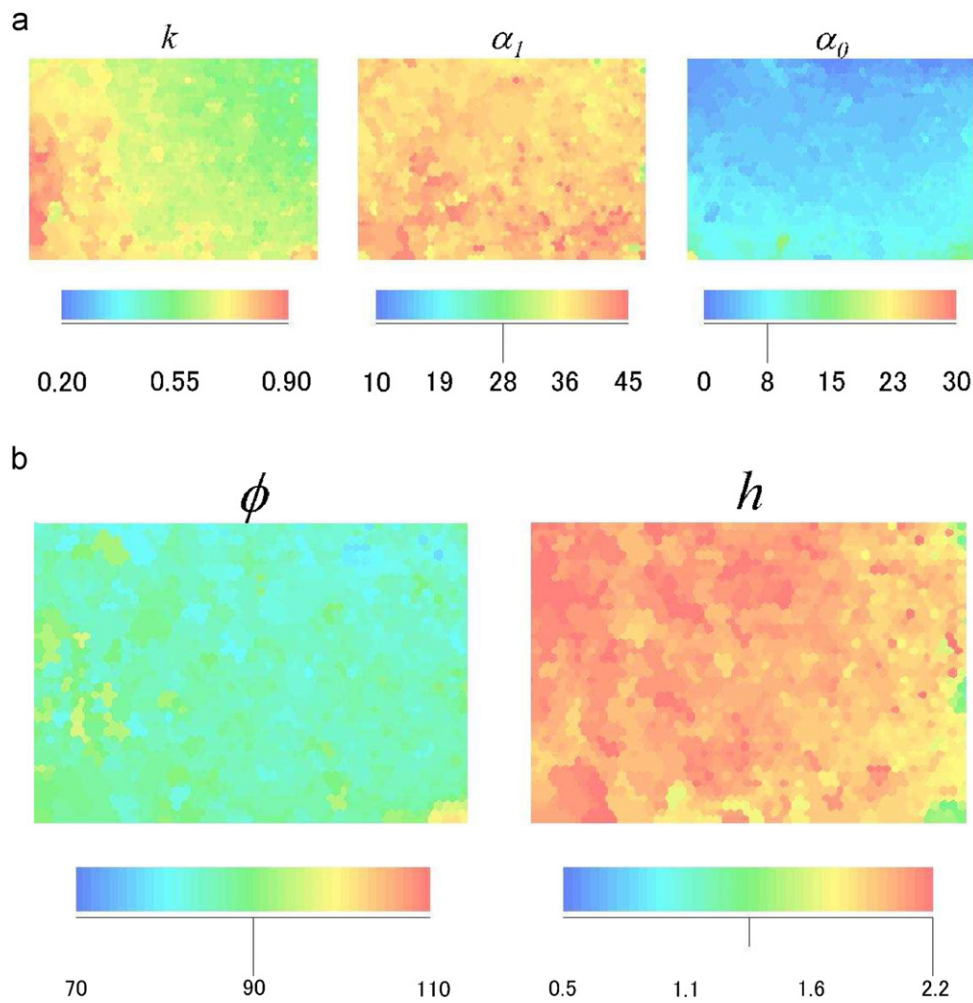


Fig. 33. SOM colored according to each design parameter.

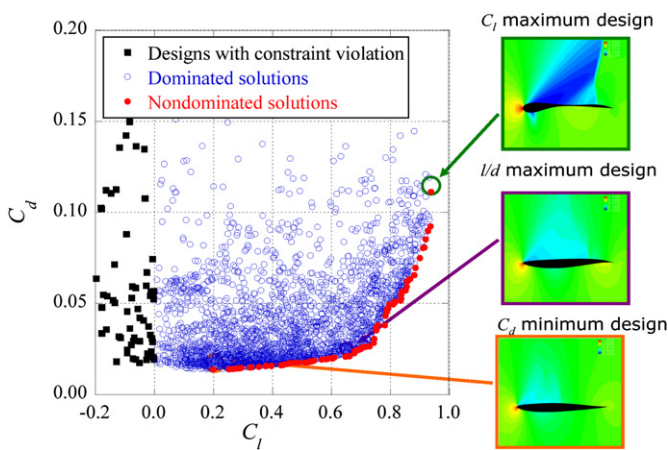


Fig. 34. Distribution of solutions.

extract dominant features in data by decomposing the data into a set of optimal orthogonal base vectors of decreasing importance. In this section, application of this new approach to an aerodynamic transonic airfoil shape optimization problem is briefly presented.

5.3.1. Analyzed solutions

The objective functions of the optimization problem are lift coefficient to be maximized and drag coefficient to be minimized. Constraints are applied so that the lift coefficient becomes positive and the maximum thickness must be greater than 0.10 airfoil chord length. Design parameters are coordinates of six control points of the B-spline curves representing an airfoil shape. The Pareto-optimal solutions are obtained by a multiobjective evolutionary algorithm (MOEA). The lift and drag coefficients of each design candidate are evaluated by using a two-dimensional Reynolds-averaged Navier–Stokes solver. All the design candidates and the Pareto-optimal solutions are plotted in Fig. 34. The number of obtained non-dominated solutions is 85. Static pressure distribution around the lift maximum, lift-to-drag-ratio maximum, and drag minimum airfoils are also shown in the figure.

5.3.2. POD-based data mining of Pareto-optimal solutions

Here, shape data of the Pareto-optimal airfoils are analyzed by using the snapshot POD proposed by Sirovich [76]. The Pareto-optimal solutions are numbered from the drag minimum design to the lift maximum design as shown in Fig. 35 to be analyzed by POD. The shape and surface pressure data analyzed here are y coordinates and surface pressure on all grid points around the

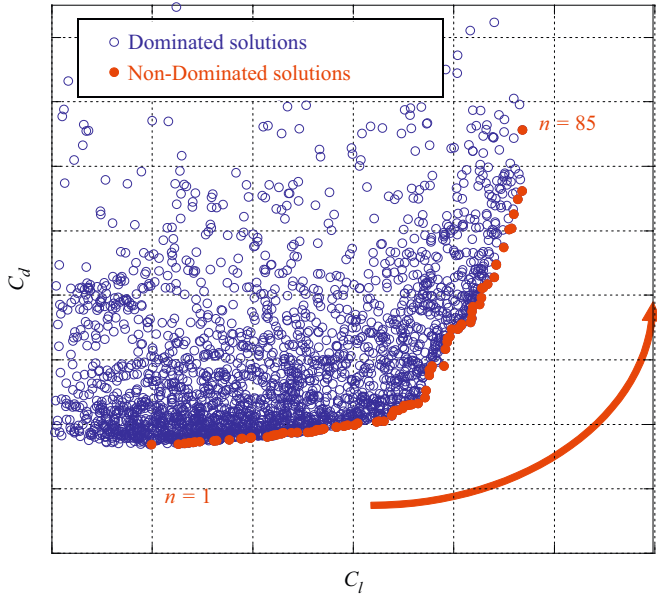


Fig. 35. Numbering of the non-dominated solutions.

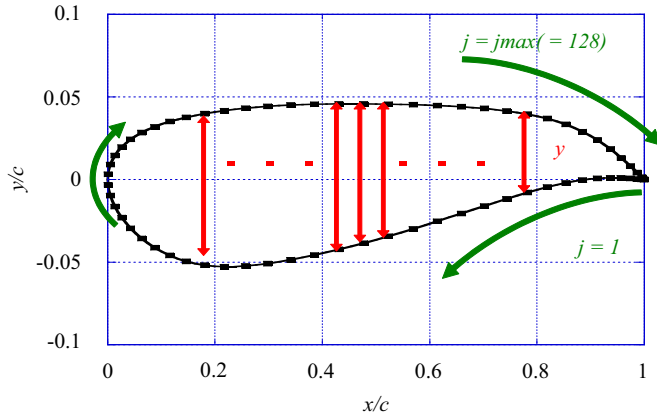


Fig. 36. Definition of the shape data.

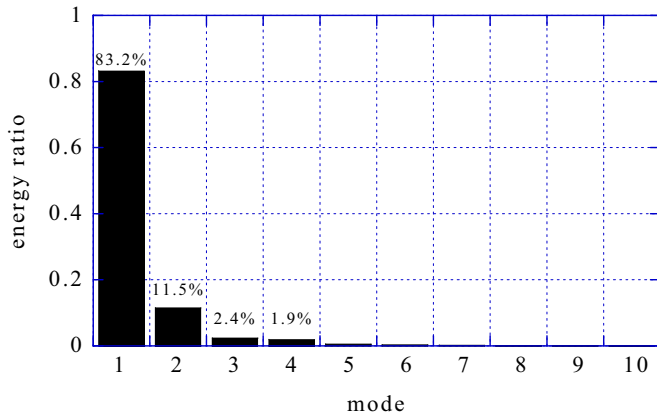


Fig. 37. Energy ratio of the principle modes.

airfoil as shown in Fig. 36. The number of grid points around an airfoil is 127.

5.3.3. Result

First, shape data is analyzed. Energy ratio of the 10 principal orthogonal base vectors (principal POD modes) to the total energy is presented in Fig. 37. The first mode is dominant (more than 83%) and the first four modes represent more than 99%. Fig. 38(a) presents the lift-to-drag maximum airfoil shape and orthogonal base vectors of the first four modes. This figure indicates that the

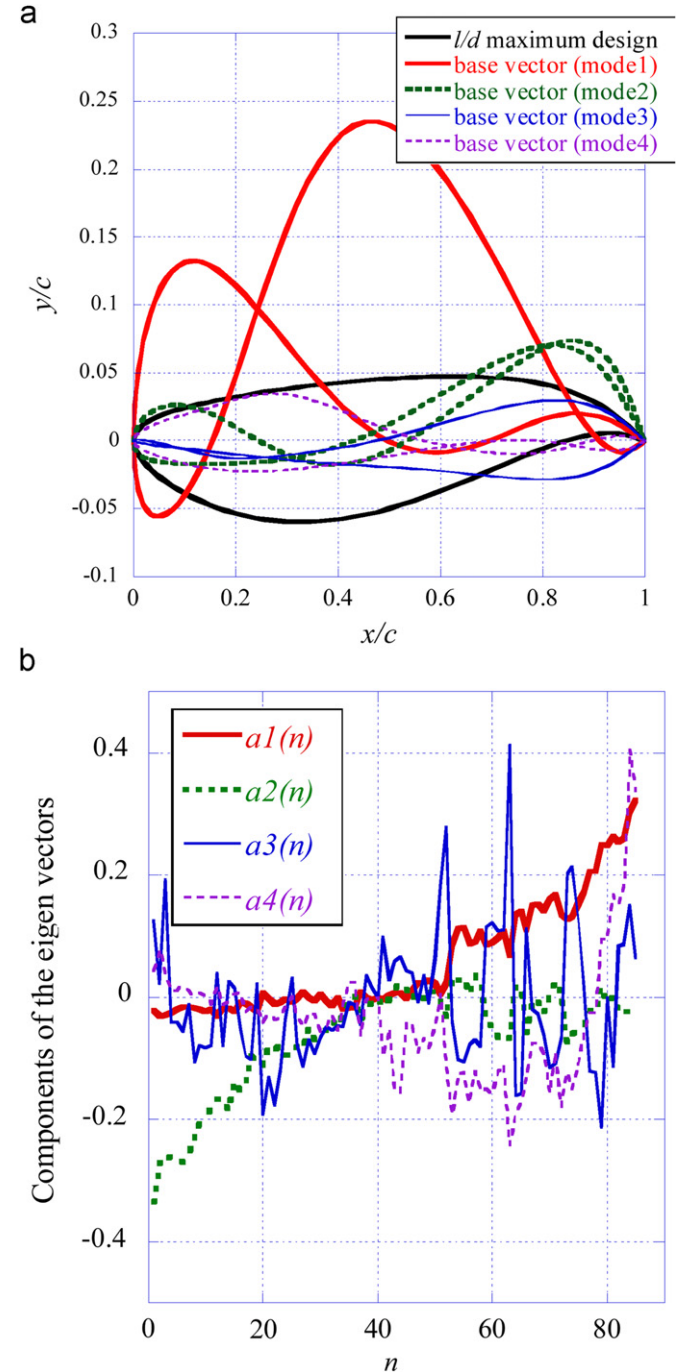


Fig. 38. Averaged airfoil shape and orthogonal base vectors (a) and eigenvectors of the first four modes against n (b).

mode 1 mainly contributes to lower surface near the mid-chord while the second mode mainly contributes to the camber near the trailing edge. Fig. 38(b) is eigenvectors of the first four modes against the index of the non-dominated solutions n . This figure indicates that the obtained non-dominated airfoil shapes are categorized into two groups, i.e., the low drag designs (roughly $n \leq 50$ and $C_d \leq 0.75$) and the high lift designs ($n > 50$ and $C_l > 0.75$). As for the low drag designs, the second mode is dominant as the eigenvector of the first mode is almost zero. This means that the Pareto-optimal low drag designs increase lift by changing the camber near the trailing edge. As for the high lift designs, the eigenvector of the second mode is almost zero. This indicates that the first mode is dominant: lift is increased by moving lower surface near the mid-chord upward without significant change in the trailing edge angle.

Acknowledgments

This work is supported by the NASA research Grant NAG3-2869 under the Ultra Efficient Engine Technology Program, managed by Carol Ginty. Meng-Sing Liou is also supported by the Subsonic Fixed Wing Project.

References

- Alexandrov NM, Dennis Jr. JE, Lewis RM, Torczon V. A trust-region framework for managing the use of approximation models in optimization. *Structural Optimization* 1998;15:16–23.
- Arnone A, Liou MS, Povinelli LA. Multigrid calculation of three-dimensional viscous cascade flows. NASA TM-105257; 1991.
- Arnone A. Viscous analysis of three-dimensional rotor flow using a multigrid method. *ASME Journal of Turbomachinery* 1994;116:435–45.
- Birger IA. Safety factors and diagnostics: problems of mechanics of solid bodies. Leningrad: Sudostroenie; 1970. p. 71–82 [in Russian].
- Benini E. Three-dimensional multi-objective design optimization of a transonic compressor rotor. *Journal of Propulsion and Power* 2004 559–65.
- Bishop CM. Neural network for pattern recognition. Oxford University Press; 2003.
- Booker LB. Improving search in genetic algorithms. In: Genetic algorithms and simulated annealing. San Mateo, CA: Morgan Kaufmann; 1987. p. 1–63.
- Carpenter WC. Effect of design selection on response surface performance. NASA CR-4520; 1993.
- Chorin AJ. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics* 1967;2:12–26.
- Cleveland WS. Elements of graphing data. California: Wadsworth Advanced Books and Software; 1985.
- Colozza A. Planetary exploration using biomimetics: an entomopter for flight on Mars. Final Report for Phase II Project NAS5-98051.
- Deb K. Multi-objective optimization using evolutionary algorithms. New York: Wiley; 2001.
- De Jong KA. An analysis of the behavior of a class of genetic adaptive systems. PhD dissertation, University of Michigan, Ann Arbor; 1975.
- Donald RJ, Matthias S, William JW. Efficient global optimization of expensive black-box function. *Journal of Global Optimization* 1998;13:455–92.
- DOT user's manual, Version 4.20, Vanderplaats Research & Development, Inc., Colorado Springs, CO, 1995.
- Duchon JP. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In: Schempp W, Zeller K, editors. Constructive theory of functions of several variables, Oberwolfach 1976. Berlin: Springer; 1977. p. 85–100.
- Enevoldsen I, Sorensen JD. Reliability-based optimization in structural engineering. *Structural Safety* 1994;15:169–96.
- Fonseca CM, Fleming PJ. Genetic algorithms for multiobjective optimization: formulation, discussion, and generalization. In: Proceedings of the fifth international conference on genetic algorithms, San Mateo, CA: Morgan Kaufmann; 1993. p. 416–23.
- Fujimoto K, Fujii K. Computational aerodynamic analysis of capsule configurations toward the development of reusable rockets. *Journal of Spacecraft and Rockets* 2006;43(1):77–83.
- Geoffrion AM, Dyer JS, Feinberg A. An interactive approach for multi-criterion optimization with an application to the operation of an academic department. *Management Science* 1972;19(4):357–68.
- Goldberg DE. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley Publishing Company, Inc.; 1989. p. 191–5.
- Giunta AA, Eldred MS. Implementation of a trust region model management strategy in the DAKOTA optimization toolkit. AIAA Paper 2000-4935, 2000.
- Haldar A, Mahadevan S. Probability, reliability and statistical methods in engineering design. New York: Wiley; 2000.
- Isogai K, Shinmoto Y, Watanabe Y. Effects of dynamic stall on propulsive efficiency and thrust of flapping airfoil. *AIAA Journal* 1999;37(10):1145–51.
- Ito T, Fujii K, Hayashi K. Computations of axisymmetric plug-nozzle flowfields: flow structures and thrust performance. *Journal of Propulsion and Power* 2002;18(2):254–60.
- Jameson A. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. AIAA Paper 1991-1596, 1991.
- Jeong S, Chiba K, Obayashi S. Data mining for aerodynamic design space. *Journal of Aerospace Computing, Information and Communication* 2005;2(11):452–69.
- Kawai S, Fujii K. Analysis and prediction of thin-airfoil stall phenomena with hybrid turbulence methodology. AIAA Journal 2005;43(5):953–61.
- Kaya M, Tuncer IH, Jones KD, Platzer MF. Optimization of flapping motion parameters for two airfoils in a biplane configuration. *Journal of Aircraft* 2009;46:583–92.
- Keane AJ. Wing optimization using design of experiment, response surface, and data fusion methods. *Journal of Aircraft* 2003;40:741–50.
- Kohonen T. Self-organizing maps, 2nd ed.. Heidelberg, Germany: Springer; 1997.
- Knill DL, Giunta AA, Baker CA, Grossman B, Mason WH, Haftka RT, Watson LT. Response surface methods combining linear and euler aerodynamics for supersonic transport design. *Journal of Aircraft* 1999;36:75–86.
- Lian Y, Kim N-H. Reliability-based design optimization of a transonic compressor. AIAA Journal 2006;44(2):368–75.
- Lian Y, Liou MS, Oyama A. An enhanced evolutionary algorithm with a surrogate model. In: Proceedings of genetic and evolutionary computation conference, Seattle, WA, 2004.
- Lian Y, Liou MS. Multiobjective optimization using coupled response surface model and evolutionary algorithm. AIAA Journal 2005;43(6):1316–25 [also AIAA Paper 2004-4323].
- Lian Y, Liou MS. Multiobjective optimization of a transonic compressor rotor using evolutionary algorithm. *Journal of Propulsion and Power* 2005;21(6):979–87 [also AIAA Paper 2005-1816, 2005].
- Lian Y, Liou MS. Aero-structural optimization of a transonic compressor blade. *Journal of Propulsion and Power*, 2006;22(4):880–8 [also AIAA Paper 2005-3634].
- Lian Y, Liou MS, Oyama A. An enhanced evolutionary algorithm with a surrogate model. In: Proceedings of genetic and evolutionary computation conference—GECCO 2004, Seattle, WA, 2004 [on CD-ROM].
- Madsen JJ, Shyy W, Haftka RT. Response surface techniques for diffuser shape optimization. AIAA Journal 2000;38:1512–8.
- Manas MS. Graphical methods of multi-criterion optimization. *Zeitschrift fur Angewandte Mathematik und Mechanik* 1982;62(5):375–7.
- McKay MD, Bechman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 1979;21:239–45.
- Michielson RC, Reece S. Update on flapping wing micro air vehicle research on going work to develop a flapping wing, crawling Entomopter. In: Proceedings of the 13th international reactor pressure vessel conference, 1998. p. 613–7.
- Muyl F, Dumas L, Herbert V. Hybrid method for aerodynamic shape optimization in automotive industry. *Computers & Fluids* 2000;33:849–58.
- Myers RH, Montgomery DC. Response surface methodology: process and product optimization using designed experiments. New York, NY: Wiley; 2002.
- Obayashi S, Chiba K. Data mining for multidisciplinary design space of regional-jet wing. *Journal of Aerospace Computing, Information and Communication* 2007;4(11):1019–36.
- Obayashi S, Chiba K. Knowledge discovery for flyback-booster aerodynamic wing using data mining. *Journal of Spacecraft and Rockets* 2008;45(5) 975–87.
- Ong YS, Nair PB, Keane A. Evolutionary optimization of computationally expensive problems via surrogate modeling. AIAA Journal 2003 687–96.
- Oyama A, Liou MS. Multiobjective optimization of rocket engine pumps using evolution algorithm. *Journal of Propulsion and Power* 2002;18(3):528–35.
- Oyama A, Liou MS, Obayashi S. Transonic axial-flow blade shape optimization using evolutionary algorithm and three-dimensional Navier–Stokes solver. *Journal of Propulsion and Power* 2004;20:612–9.
- Oyama A, Okabe Y, Fujii K, Shimoyama K. A study on flapping motion for MAV design using design exploration. AIAA Paper 2007-2878, 2007.
- Oyama A, Nonomura T, Fujii K. Data mining of pareto-optimal transonic airfoil shapes using proper orthogonal decomposition. AIAA Paper AIAA-2009-4000, AIAA, Reston, VA, 2009.
- Qu X, Haftka R. Reliability-based design optimization using probabilistic factor. *Journal of Structural and Multidisciplinary Optimization* 2004;27(5) 302–13.
- Qu X, Haftka R, Venkataraman S, Johnson T. Deterministic and reliability-based optimization of composite laminates for cryogenic environments. AIAA Journal 2003;41:2029–36.
- Rodríguez J, Renaud JE, Watson LT. Convergence of trust region augmented lagrangian methods using variable fidelity approximation data. *Structural Optimization* 1998;15:1–7.
- Roe PL. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics* 1981;43:357–72.

- [74] Sacks J, Welch WJ, Mitchell TJ, Wynn HP. Design and analysis of computer experiments. *Statistical Science* 1989;4:409–35.
- [75] Shanno DF. Conditioning of quasinewton methods for function minimization. *Mathematics of Computation* 1970;24:647–64.
- [76] Sirovich L. Turbulence and dynamics of coherent structures part 1: coherent structures. *Quarterly of Applied Mathematics* 1987;45(3):561–71.
- [79] Srinivas N, Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* 1994;12:221–48.
- [81] Strazisar AJ, Wood JR, Hathaway MD, Suder KL. Laser Anemometer measurements in a transonic axial-flow fan rotor. NASA Technical Paper 2879, 1989.
- [82] Sugimura K, Obayashi S, Jeong S. Multi-objective design exploration of a centrifugal impeller accompanied with a vaned diffuser. In: *Proceedings of fifth joint ASME/JSME fluid engineering conference*, 2007.
- [83] Sugimura K, Jeong S, Obayashi S, Kimura T. Multi-objective robust design optimization and knowledge mining of a centrifugal fan that takes dimensional uncertainty into account. In: *Proceedings of ASME turbo expo 2008*, 2008.
- [84] Tani N, Oyama A, Yamanishi N. Multiobjective design optimization of rocket engine turbopump turbine. In: *Proceedings of the fifth international spacecraft propulsion conference/second international symposium on propulsion for space transportation*, 2008 [on CD-ROM].
- [85] Tatsukawa T, Oyama A, Fujii K. Comparative study of data mining methods for aerodynamic multiobjective optimizations. In: *Proceedings of the eighth world congress on computation mechanics and the fifth European congress on computational methods in applied sciences and engineering*, 2008 [on CD-ROM].
- [86] Taylor GK, Nudds RL, Thomas ALR. Flying and swimming animals cruise at a Strouhal number tuned for high power efficiency. *Nature* 2003 707–11.
- [87] Tu J, Choi KK, Park YH. A new study on reliability-based design optimization. *ASME Journal of Mechanical Design* 1999;121(4):557–64.
- [88] Unal R, Lepsch RA, McMillin ML. Response surface model building and multidisciplinary optimization using D-optimal design. AIAA Paper 98-4759, 1998.
- [89] van Leer B. Towards the ultimate conservative difference scheme. A second-order sequel to Godunov's method. *Journal of Computational Physics* 1979;32:101–36.
- [90] Van Veldhuizen DA, Gary BL. Evolutionary computation and convergence to a pareto front. In: Koza JR, editor. *Late breaking papers at the genetic programming 1998 conference*, Stanford University, Stanford, 1998. p. 221–228.
- [92] Wang GG. Adaptive response surface method using inherited latin hypercube design points. *Journal of Mechanical Design* 2003;125:210–20.
- [94] Yamashiro D, Yoshikawa T, Furuhashi T. Visualization method of pareto solutions reflecting users' subjectivity in multi-objective optimization problems. In: *Proceedings of the 2007 international conference on parallel and distributed processing techniques and applications*, 2007.
- [95] Yoon S, Jameson A. Lower-upper symmetric-Gauss-Seidel method for the euler and Navier-Stokes equations. *AIAA Journal* 1998;26(9):1025–6.
- [96] Young J. Numerical simulation of the unsteady aerodynamics of flapping airfoils. PhD dissertation, School of Aerospace, Civil and Mechanical Engineering, The University of New South Wales, Australian Defense Force Academy, Sydney, Australia; 2005.