AIAA JOURNAL
Vol. 43, No. 6, June 2005

Multiobjective Optimization Using Coupled Response Surface Model and Evolutionary Algorithm

Yongsheng Lian*
Ohio Aerospace Institute, Cleveland, Ohio 44142
and
Meng-Sing Liou†
NASA John H. Glenn Research Center at Lewis Field, Cleveland, Ohio 44135

In this work we develop an efficient approach for computationally expensive multiobjective design optimization problems. In this approach we bring together design of experiment, a response surface model, a genetic algorithm, and computational-fluid-dynamics analysis tools to provide an integrated optimization system. We use an improved hypercube sampling to preselect an array of design points on which the computational-fluid-dynamics code will run. Then a computationally cheap surrogate model is constructed based on response surface approximation. A real-coded genetic algorithm is then applied on the surrogate model to perform multiobjective optimization. Representative solutions are chosen from the Pareto-optimal front to verify against the computational-fluid-dynamics code. This proposed method is used in the redesign of a single-stage turbopump, a two-stage turbopump, and the NASA rotor67 transonic compressor blade. For the single-stage pump optimization problem, we can improve the total head rise by 1.2% with the same power input; for the multistage pump problem, we can improve the total head rise by 0.5% at the same power input; for the rotor67 compressor blade design, we can increase the pressure ratio by 1.8% or reduce the entropy generation by 6.2%. We achieve these with a much reduced computational cost.

I. Introduction

EVOLUTIONARY algorithms (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.[1] applied an EA in their redesign of the NASA rotor67 transonic compressor blade. Benini implemented an EA to improve the performance of the NASA rotor37 blade.[2] Oyama and Liou[3] and later Lian et al.[4] utilized EAs to the redesign of rocket turbo pumps. Another thrust behind these broad applications is that EAs are particularly suitable for multiobjective optimization problems, which are often encountered in aerospace designs. For example, in the turbopump design problems the objectives are to maximize the total head rise and to minimize the input power.[3,4] These two objectives are competing. Improving one objective will inevitably deteriorate the other. Unlike a single-objective optimization problem, a multiobjective optimization problem does not have a solution that is better than others in terms of all objectives. Instead, at the end, we expect a set of compromised solutions, each of which is better than the others in one objective but might be worse in the other objectives. These solutions are largely known as nondominated solutions or Pareto-optimal solutions or compromised solutions. In the case of multiobjective optimization problems, the use of classical methods usually converts them into multiple single-objective problems by introducing parameters such as weight vectors.[5] Each optimal solution is associated with a particular vector. To find multiple Pareto-optimal solutions, we have to choose different weight vectors and solve the resulting single-objective optimization problem repeatedly. On the other hand, EA's population approach can be exploited to emphasize all nondominated solutions in a population equally and to preserve a diverse set of multiple nondominated solutions using a niche-preserving operator.[5] As a consequence, EAs eliminate the need of any parameters using in the single-objective formulation and can find as many Pareto-optimal solutions as possible in one run.

One drawback of EAs is that they suffer a slow convergence because they use probabilistic recombination operators to control the step size and searching direction. The required population size and generation size usually demand a 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, and the overall turnaround time was 2000 h using a four-processor workstation.[2] Oyama et al. used an EA to redesign the NASA rotor67; with a population size of 64 and generation size of 100 the turnaround time was about two months using a 64-processor cluster.[1] Therefore, when expensive function evaluations are involved the required CPU time impedes practical applications of EAs even with today's computing power. 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,[6] Madsen et al. optimized a diffuser shape,[7] and Keane improved a wing design.[8]

The work reported here is aimed at performing multiobjective optimization for computationally expensive design problems. Our strategy is to hybridize the desirable characteristics of EAs and response surface approximations. For that purpose we integrate design of experiment (DOE), response surface approximations, genetic algorithms, and computational-fluid-dynamics (CFD) tools to provide an efficient design optimization system. The proposed approach is then used in the redesign of a single-stage turbopump, a two-stage turbopump, and the NASA rotor67 transonic compressor blade. This paper is organized in the following way: first, we

Received 24 August 2004; presented as Paper 2004-4323 at the AIAA/ISSMO 10th Multidisciplinary Analysis and Optimization Conference, Albany, NY, 30 August–1 September 2004; revision received 2 December 2004; accepted for publication 14 December 2004. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/05 $10.00 in correspondence with the CCC.

*Senior Researcher; currently Visiting Scholar, Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109; ylian@umich.edu. Member AIAA.
†Senior Scientist, MS 5-11. Associate Fellow AIAA.

introduce the concepts of RSMs and DOE; then, a real-coded genetic algorithm is introduced; next, we present the CFD codes used in our work. This is followed by our integrated optimization approach. Last, we present and discuss the numerical results for the three test problems.

II. Response Surface Methodology

RSM is a popular surrogate model to approximate computationally expensive simulation codes usually with low-order polynomials. 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.

The response surface approximation is normally chosen to be a low-order polynomial. The second-order model is widely used because of its flexibility and ease of use.[9] A second-order response surface model with d variables can be written as the following:

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

where x_i are the predictor variables, β are the p regression coefficients, and 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. The fitting model written in matrix form is

$$\hat{\boldsymbol{y}} = \boldsymbol{Xb} \quad (2)$$

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

$$\mathrm{SS}_E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (3)$$

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 rms error RMSE. Their definitions involves partitioning the total sum squares

$$\mathrm{SS}_T = \sum_{i=1}^{n} (y - \bar{y})^2$$

into a sum of squares caused by model and a sum of squares caused by error,

$$\mathrm{SS}_T = \mathrm{SS}_R + \mathrm{SS}_E \quad (4)$$

where

$$\bar{y} = \sum_{i=1}^{n} \frac{y_i}{n}$$

is the mean of response. The R^2 statistic is defined as

$$R^2 = 1 - \mathrm{SS}_E / \mathrm{SS}_T \quad (5)$$

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

$$R^2_{\mathrm{adj}} = 1 - [(n-1)/(n-p)](1 - R^2) \quad (6)$$

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:

$$\sigma = \sqrt{\mathrm{SS}_E/(n-p)} \quad (7)$$

III. Latin Hypercube Designs

The central composite design (CCD) is a popular approach with practitioners in DOE when the number of design variables is 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. However, in a deterministic simulation model the output responses are entirely deterministic quantities the values of which are determined by the mathematical models. When deterministic models are used, there is no variability in analysis outputs. Sacks et al.[10] found that in 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)[11] 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 a LHD sample.[12] In our work we apply the improved-hypercube-sampling (IHS) algorithm,[13] which differs from the traditional LHD in that it tries to pick a solution which has the property that the points are spread out as evenly as possible. It does this by determining an optimal even spacing and using the duplication factor to allow it to choose the best of the various options available to it.

There is no comprehensive theory of the number of design points required to construct the response surface models with LHD. Relevant work was conducted by Carpenter[14] and Unal et al.,[15] 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% overdetermined D-optimal designs was a good choice.

IV. 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. In their applications to function optimization, EAs start with a random population of chromosomes (candidates); 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 the candidate. In general, penalty is put on infeasible candidates to cause all infeasible solutions to have a

worse fitness than that of any feasible solution. Simply put, EAs involve three operators, that is, selection, crossover, and mutation. The primary purpose of the selection operator is to make duplicates of good candidates and eliminate bad candidates in a population while keeping constant the population size.[5] Tournament selection, proportionate selection, and ranking selection are common methods to achieve the task. In our work we choose ranking selection because of its computational simplicity. For single-objective optimization problems, the ranking is based on the fitness of a candidate. For multiobjective optimization problems, the ranking is based on Fonseca's nondominated 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 of the objective functions.[16] After ranking, the N best candidates, the same size as the initial population, are chosen from both the current and previous generations and placed in the mating pool. Here, we adopt the elitist strategy[17] 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. We also use fitness sharing[18] in the multiobjective optimization to maintain a uniform distribution on the Pareto-optimal front. 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

$$p_s = c(1-c)^{(\text{rank}-1)} \tag{8}$$

where c is a user-defined parameter. That is, the selection probability is reduced by a factor of $(1-c)$ each time when the rank is increased by 1. We choose $c = 0.01$ in our work. Then, a pair of parents are selected by using either stochastic universal sampling or roulette-wheel section, with which the same candidate can be selected more than once to become a parent. As a consequence, selection operator ensures that good candidates are preserved at the cost of bad candidates.

A crossover operator is applied next to create offsprings. In our work we apply the blend crossover (BLX-α) operator to the two mating parents, randomly chosen from the mating pool. For the two parent candidates, parent 1 and parent 2, the two offspring are expressed as

$$\text{Child } 1 = \gamma \text{ Parent } 1 + (1-\gamma) \text{ Parent } 2 \tag{9}$$

$$\text{Child } 2 = (1-\gamma) \text{ Parent } 1 + \gamma \text{ Parent } 2 \tag{10}$$

where

$$\gamma = (1+2\alpha)u - \alpha \tag{11}$$

and 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.[19]

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

$$x_i^{\text{new}} = x_i[1 + (r_i - 0.5)\Delta_i] \tag{12}$$

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

V. CFD Codes

The meanline pump flow modeling method[20] based on the Euler equations coupled with empirical correlations for rotor efficiency is used to analyze turbopumps. This method provides a fast capability for modeling turbopumps of cryogenic rocket engines. The meanline pump code can predict the performance of pumps at off-design operating conditions, given the loss of the diffusion system. The design point rotor efficiency and slip factor are obtained from empirical correlations to rotor specific speed and geometry. This method has been validated by simulating a variety of very different pumps such as Mark48 liquid-hydrogen pump, XLR-129 liquid-hydrogen pump, and orbital-transfer-vehicle liquid-oxygen pump.

The NASA rotor67 blade[21] analysis is performed based on the TRAF3D code,[22] which solves the three-dimensional Reynolds-averaged Navier–Stokes equations in generalized curvilinear coordinates. The space discretization uses a second-order cell-centered scheme with eigenvalue scaling to weigh the artificial dissipation terms. The system of equations is advanced in time using an explicit four-stage Runge–Kutta scheme. The two-layer eddy-viscosity model of Baldwin and Lomax is used for the turbulence closure. The capability of this code was validated by comparing the computational results with experiments for the Goldman annular vane, the low-speed Langston linear cascade, and the transonic rotor67.[22,23] For each simulation, a converged solution is assumed when the residuals of the continuity equation, momentum equations, and energy equation all drop by an order of four.

In our blade calculations, the inlet and outlet boundary conditions are obtained in a manner consistent with subsonic internal flows. Along the upstream boundary, the inlet flow angle, total pressure, and total temperature are specified, and the velocity components are extrapolated from the interior. The presence of an inlet boundary layer on the hub and tip endwalls is accounted for by giving a total pressure and a total temperature profile whose distribution simulates the experimental one. At the subsonic-axial outlet, the average value of the static pressure at the hub is prescribed, and the density and velocity components are extrapolated together with the circumferential distribution of pressure. The radial equilibrium equation is used to determine the spanwise distribution of the static pressure. On the solid wall, the pressure is extrapolated from the interior points, and the no-slip condition and an adiabatic wall temperature condition are used to compute density and total energy. In the present work we assume a constant temperature on the wall equal to the total inlet one. We set periodic phantom cell values to overimpose the periodic condition from blade passage to blade passage. We use phantom cells to overlap the real ones where the grid is not periodic. The values of dependent variables in the phantom cells are computed using linear interpolations.

The rotor67 geometry is parameterized with 14 surfaces of revolution from a surface at the hub to a surface at the casing of the machine. At each surface 35 point coordinates are measured from the leading edge to the trailing edge, leading to 510 points in total to define the compressor-blade geometry. The CFD grid is generated based on these points. For design optimization purpose, it is impractical to parameterize the blade using 510 design variables. In our design optimization process we define a new rotor-blade geometry by superimposing perturbation on the original baseline rotor67 blade. The perturbation is obtained by linear interpolation of four blade profiles along the span (hub, 31% span, 62% span, and tip). Each blade profile can be defined by a mean camber line and thickness distributions, and they are parameterized by a third-order B-spline curves. The camber is determined by three points and the thickness distribution by five points. As a result, eight design variables are required to represent such perturbation profile, resulting in 32 design variables in total. By doing this, we can recover the baseline geometry by setting all of the design variables as zero. To make the optimization results comparable to those of the baseline configuration, we maintain the chord distribution along the span and fix the meridional contour of the hub, casing, the sweep, and lean. For a single passage, the chordwise, tangential, and spanwise grid point numbers are 201, 53, and 57, respectively. This leads to 0.6×10^6 nodes per single passage. The structured CFD grid is shown in Fig. 1. We only compute one passage while imposing periodic boundary conditions to the other passages.

VI. Coupled Model

The proposed approach has the following steps: 1) sampling design points using IHS 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, and 5) choosing representative solutions along the Pareto-optimal front obtained from step 4 using the K-means clustering algorithm 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. We need to verify these solutions against the CFD code. To verify the whole set of Pareto-optimal solutions can be time demanding. Instead, we can choose a subset of the solutions. For that purpose we use K-means clustering algorithm.

K-means clustering algorithm is a popular clustering technique to find a subset of points that 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 $\{\boldsymbol{x}\}$ into K disjoint subsets S_j containing N_j data points. We wish to find a set of K representative vectors $\{\boldsymbol{\mu}_j\}_1^K$ in such a way as to minimize the sum-of-squares clustering function given by

$$J = \sum_{j=1}^{K} \sum_{n \in S_j} \left\| \boldsymbol{x}^n - \boldsymbol{\mu}_j \right\|^2 \tag{13}$$

where $\boldsymbol{\mu}_j$ 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.[24]

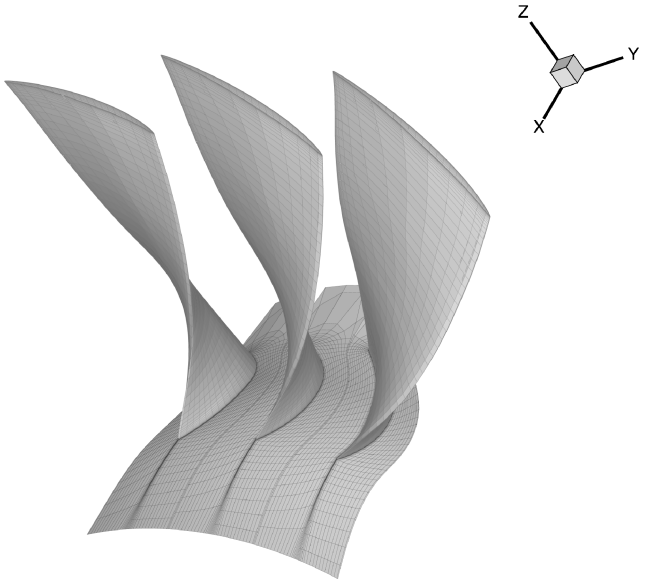


Fig. 1 Structured grid for single passage with 0.6×10^6 nodes.

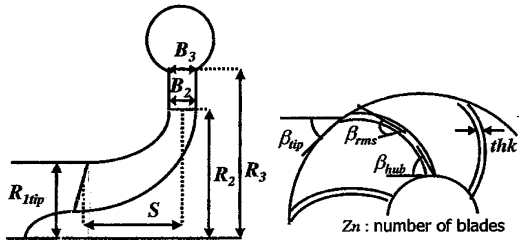


Fig. 2 Design parameters of the single-stage centrifugal turbopump design problem (from Oyama and Liou[3]).

VII. Numerical Results

We apply the proposed approach to the redesign of a single-stage centrifugal pump. The schematic of such a pump is shown in Fig. 2. The pump has a shaft rotative speed of 5416.7 rpm and a mass flow rate of 85.67 kg/s. At the inlet the flow total temperature is 29.6°C, and total pressure is 35,200 kgf/m^2. At the design point the pump produces a total head rise of 26.57 m with 441.6-kW input power. For this single-stage pump design the objectives are to minimize the input power and to maximize the total head rise. We select 11 design variables. The design space and the baseline design values are listed in Table 1.

Based on IHS, we sample 121 design points, an 50% overdetermined design. During this process, no interrelationship between design variables is assumed. These sampling points are evaluated with the meanline code.[20] Two response surface models are then constructed, one for the total head rise and the other for the power input. The accuracy of the response surface approximation is evaluated by statistical measures shown in Table 2. The values of R^2_{adj} indicate that the quadratic response surface models give accurate representations of the response functions.

Because we apply the generic algorithm on a computationally cheap surrogate model, we can readily increase the population and generation sizes without considerably increasing the total computational cost. Different settings of population and generation sizes are tested. With a population size of 110, solution converges after 1000 generations. In total, there are 1569 Pareto-optimal solutions. It is noted, however, that these solutions from the surrogate model might not satisfy the meanline code. We need to verify them using the meanline code. For that purpose we chose 20 representative designs from these Pareto-optimal solutions using the K-means clustering algorithm. We call these selected designs clustered points. The clustering is based on the total head rise and power input. By using the K-means clustering technique, input variables are scaled to the range

Table 2 Statistical measures of quadratic response surface approximation for the single-stage pump problem with 121 design points

Error statistics	Head rise	Power input
R^2	0.9994	0.9994
R^2_{adj}	0.9965	0.9965
RMSE	0.1155	2.295
Mean of response	26.30	437.1

Table 1 Design space for the single-stage pump design problem

Lower limit	Baseline	Upper limit	Quantity
8.928	9.30	9.672	$R_{1,\text{tip}}$: rotor leading-edge tip radius from centerline, cm
13.06	13.6	14.14	R_2: rotor trailing-edge tip radius from centerline, cm
14.40	15.0	15.60	R_3: hub radius, cm
1.978	2.06	2.142	B_2: blade span at trailing edge normal to the direction of flow, cm
2.218	2.31	2.402	B_3: blade span at diffuser leading edge, cm
9.792	10.2	10.608	S: axial length of the blade at the rms diameter from leading to trailing edge, cm
33.60	35.0	36.40	β_{hub}: blade trailing-edge angle at the hub, deg
33.60	35.0	36.40	β_{rms}: blade trailing-edge angle at the rms radius, deg
33.60	35.0	36.40	β_{tip}: blade trailing-edge angle at the tip, deg
0.122	0.127	1.321	thk: blade normal thickness, cm
7	12	17	Z_{n2}: number of blades at the rotor

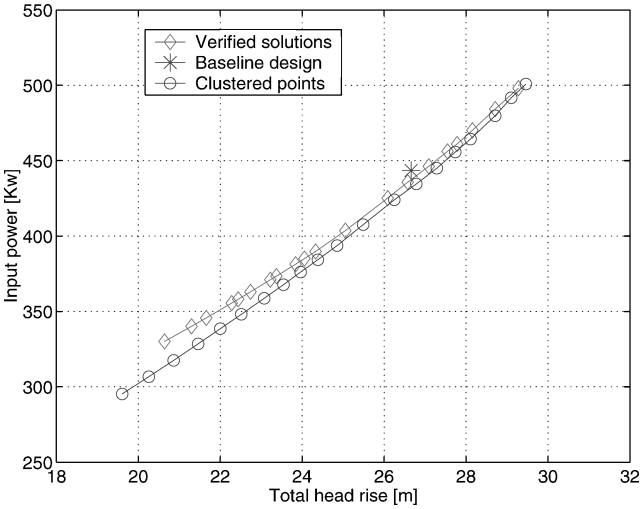


Fig. 3 Verified solutions with 121 design points and 20 clustered points.

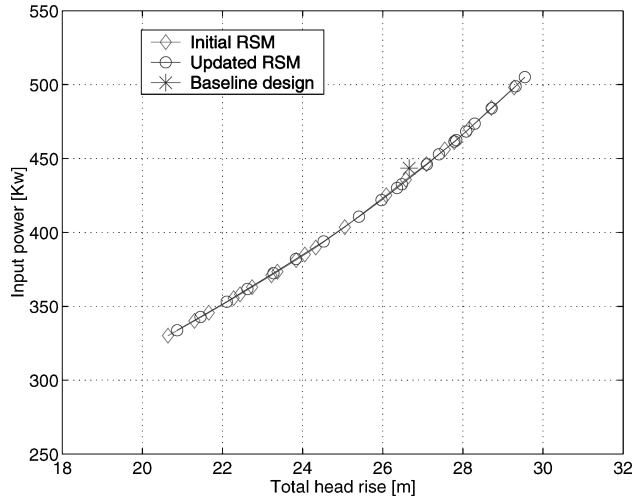


Fig. 4 Pareto-optimal solutions by the search on the updated response surface approximations.

of [0 1] so that no single variable dominates the sum of squares in Eq. (13). This is a consequence of the fact that the sum of squares is determined by the Euclidean distance between the input vector and the mean of the data points. If one of these input variables has a much larger range of values than the others, the clustering function will be very sensitive to that variable. Figure 3 demonstrates the distribution of the clustered points. It shows an almost uniform distribution over the range, indicating a good representation of the whole Pareto-optimal front.

The verified solutions are also shown in Fig. 3, together with the baseline design. Some solutions outperform the baseline by increasing the total head rise by 1.2% with the same power input as the baseline. Oyama and Liou[3] also studied this problem by directly coupling a real-coded genetic algorithm with the meanline code. They achieved 1% improvement in the total head. However, they called the meanline code 3000 times, whereas we used it for fewer than 200 times.

Keane suggested updating the data set used to build the response surface by adding new points in areas where good designs were achieved.[8] With this update he improved the initial design from the original response surface. We add the 20 verified designs to the initial design points and construct response surfaces based on the new set of 141 design points. The statistical measures are shown in Table 3. The optimization results are compared with those from the initial design in Fig. 4. With the updated response surface model no evident improvement is made beyond the initial model. There should be no surprise for this because our initial approximation is

Table 3 Statistical measures of the updated quadratic response surface approximation for the single-stage pump problem

Error statistics	Head rise	Power input
R^2	0.9988	0.9987
R^2_{adj}	0.9964	0.9920
RMSE	0.1253	2.540
Mean of response	25.87	408.1

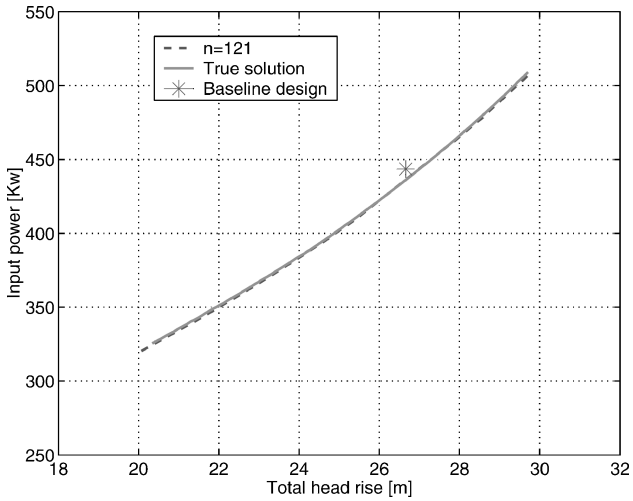


Fig. 5 Comparison between the true solutions and the RSM solutions.

already very good, with a value of R^2_{adj} larger than 0.99. Adding the 20 design points only slightly changes the value of R^2_{adj}. In the case reported by Keane, the correlation coefficient is 0.4903, which does not guarantee a good predictive capability, and adding new points will improve the prediction capability.

Oyama and Liou[3] approximated the true Pareto-optimal front by applying a genetic algorithm on the meanline code with population size of 400 and generation size of 300. In Fig. 5 we compare our result from 121 design points with the "true" solution, which is obtained based on the same approach and same parameter setting reported by Oyama and Liou. At both ends of the head rise, our solutions outperform the true Pareto-optimal solutions; in the middle, our solutions match the true solutions well. Notwithstanding, readers should not be misled by this comparison. Optimizations based on accurate models commonly yield optimal solutions as good as RSM approach if enough search evaluations are taken. The seemingly conflicting phenomenon in Fig. 5 implies that true optimal solutions have not been reached yet. This is quite normal because EAs typically have slow convergence rate, especially when dealing with high-dimensional problems.

The number of design points impacts the accuracy of the response surface approximation and therefore the final optimization result. Figure 6 compares performance with three different numbers of design points. In all cases 20 clustered points are chosen to verify from the respective Pareto-optimal front. The realization with 121 design points outperforms the other two. The performance with 156 design points, a 100% overfitting design, is not as good as that with either 93 or 121 points, suggesting that overfitting the response surface model might not a good practice. Even though the number of design points is important, there is no comprehensive theory in this respect. In most cases it is problem dependent. From our experience we suggest a 20–80% overfitting design.

Next, we apply our approach on the redesign of the RL10A-3-3A liquid-oxygen pump, which is a liquid-oxygen/liquid-hydrogen expander cycle engine and has been used for the expandable Atlas Centaur upper-stage vehicles since 1984 (see Fig. 7). The pump has one inducer and a single centrifugal impeller, followed by a vaneless diffuser and conical exit volute. At the design point, the pump has a shaft speed of 12,900 rpm and a mass flow rate of 18.1 kg/s. At

Table 4 Design parameter spaces for the two-stage turbopump redesign problem

Upper limit	Baseline	Lower limit	Quantity
			First-stage parameters
2.93	2.87	2.81	$R_{1,\mathrm{tip}}$: rotor leading-edge tip radius from centerline, cm
1.24	1.22	1.20	$R_{2,\mathrm{hub}}$: rotor trailing-edge tip radius from centerline, cm
43.86	43.0	42.14	$\beta_{2,\mathrm{hub}}$: blade trailing-edge angle at the hub, deg
27.84	27.3	26.75	$\beta_{2,\mathrm{rms}}$: blade trailing-edge angle at the rms radius, deg
22.03	21.6	21.17	$\beta_{2,\mathrm{tip}}$: blade trailing-edge at the tip, deg
0.104	0.102	0.100	thk_2: blade normal thickness, cm
			Second-stage parameters
5.44	5.33	5.22	R_3: rotor trailing-edge radius from centerline, cm
0.246	0.241	0.236	$R_4 - R_3$: radial seal clearance for shrouded rotor, cm
0.651	0.638	0.625	B_3: blade span at the trailing edge, cm
1.04	1.02	1.00	B_4: blade span at diffuser leading edge, cm
0.0777	0.0762	0.0747	thk_3: blade normal thickness, cm
2.27	2.23	2.19	S: axial length of the blade at the rms diameter from leading to trailing edge, cm
14	12	10	Z_{n3}: number of blades at the rotor
90.0	90.0	88.2	β_3: blade trailing edge angle, deg

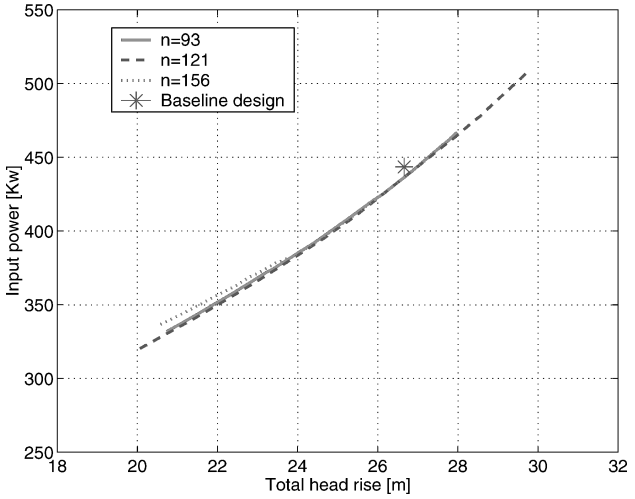


Fig. 6 Performance based on different sizes of sampling points.

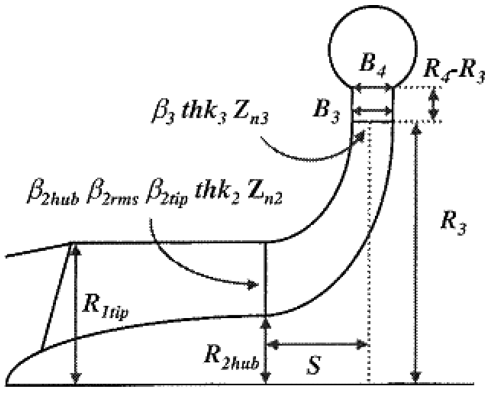


Fig. 7 Design parameters of the two-stage turbopump design problem.

the first stage, the inlet flow total temperature is −175.9°C, and total pressure is 275,790 Pa; at the second stage, the corresponding values are −175.8°C and 499,629 Pa, respectively. Our objectives in the redesign are to maximize the total head and to minimize the input power. The 14 design variables and their ranges are listed in Table 4.

We chose 180 design points to fit the response surface model, which is a 50% overfitting design. Two response surface models are constructed, one for the head rise and the other for the power input. The statistical measures of the fitting are shown in Table 5. The

Table 5 Statistical measures of quadratic response surface approximation for the two-stage pump problem

Error statistics	Head rise	Power input
R^2	0.9892	0.9890
R^2_{adj}	0.9678	0.9673
RMSE	0.2865	1.066
Mean of response	30.70	91.47

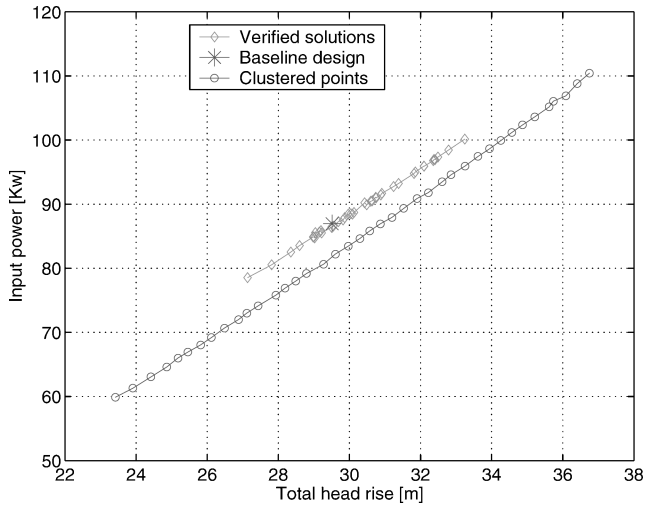


Fig. 8 Verified solutions with RSM using 180 design points.

constructed models are then optimized with a genetic algorithm. With a population size of 140, converged solution is obtained after 2000 generations. Forty clustered points are selected from the 958 Pareto-optimal solutions to verify with the meanline code. The final optimization results are plotted in Fig. 8. Comparing with the baseline design, some new designs improve the total head rise by 0.5% at the same power input. By directly combining a real-coded genetic algorithm with the meanline code, Oyama and Liou[3] achieved designs that improved both total head and input power by as much as 1% after evaluating the meanline code 3600 times.

The third problem we consider here is the redesign of the NASA rotor67 compressor,[21] which is a transonic axial-flow fan rotor and is the first stage of a two-stage fan. At the design point, rotor67 has a pressure ratio of 1.63 at a mass flow rate of 33.25 kg/s. The operating rotational speed is 16,043 rpm, which generates a tip speed of 429 m/s and an relative inlet tip Mach number of 1.38. The rotor

has a low aspect ratio of 1.56 and totally has 22 blades. It has a varied solidity from 3.11 at the hub to 1.29 at the tip. The inlet and exit hub/tip radius ratios are 0.375 and 0.478, respectively. The Reynolds number based on the chord length at the hub is 1.797×10^6.

The purpose of our redesign is to minimize the entropy generation and to maximize the stage pressure ratio p_{03}/p_{01} while maintaining the mass flow rate. We require that the relative difference in mass flow rate between the new design and the rotor67 is less than 0.5%. We sample 1024 design points with the IHS to construct the response surface, which represents an 80% overdetermined design. With eight Intel Intanium processors of 1.3 GHz, the turnaround time to evaluate these 1024 cases with the CFD code RAF3D is about 128 h (5.3 days). We built two response surface models, one for the entropy generation and the other for the stage pressure ratio. Again, the accuracy of the response surface approximation is shown in Table 6.

We set the population size and generation size as 320 and 2000, respectively. We choose 12 representative solutions from the Pareto-optimal front to verify using the TRAF3D CFD code. The verified solutions together with the baseline design are shown in Fig. 9. These solutions represent different tradeoffs between the stage pressure ratio and entropy generation. Comparing with the baseline design, all of the tradeoff solutions reduce the entropy generation and increase the stage pressure ratio. Some solutions increase the pressure ratio as much as 1.8% and reduce the entropy production by 6.2%. Table 7 lists the performance of two typical designs, representing designs

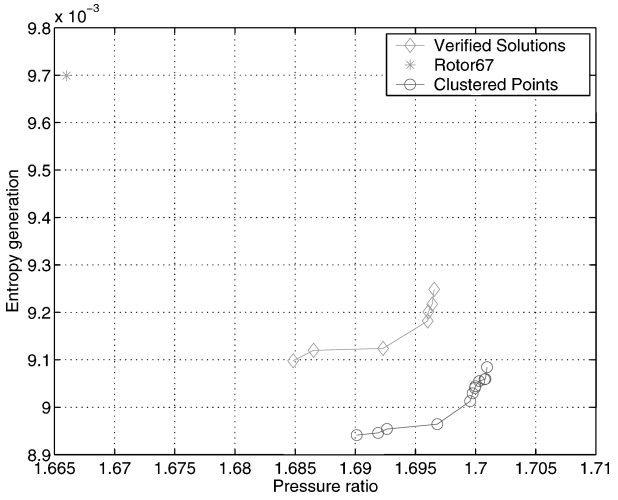
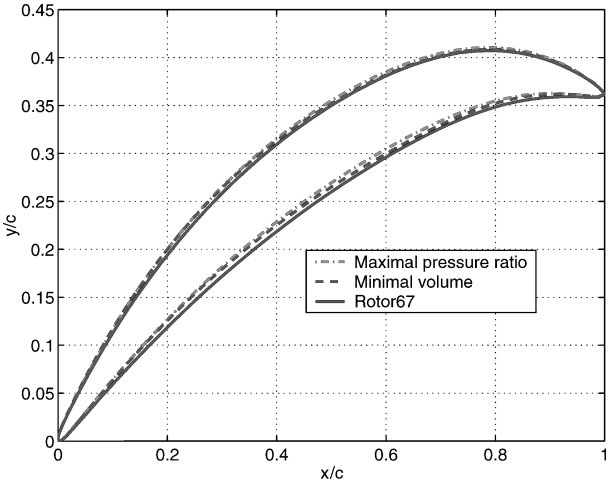


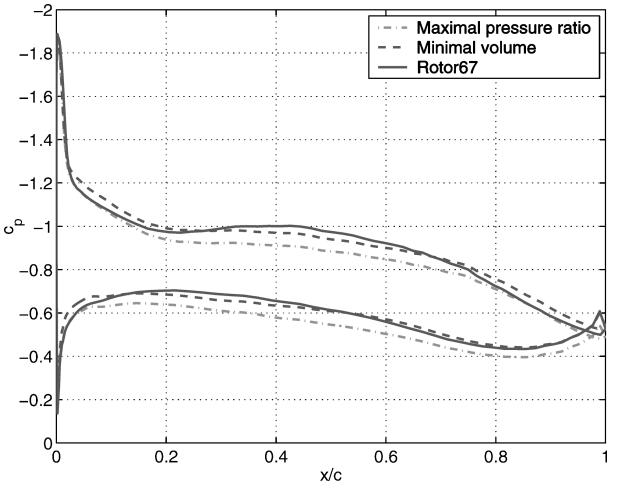
Fig. 9 Pareto-front optimal solutions for the rotor67.

Table 6 Statistical measures of quadratic response surface approximation for the rotor67 problem

Error statistics	p_{02}/p_{01}	Entropy generation
R^2	0.9949	0.9639
R^2_{adj}	0.9888	0.9205
RMSE	0.6e-3	0.7e-5
Mean of response	1.6673	3.37e-4

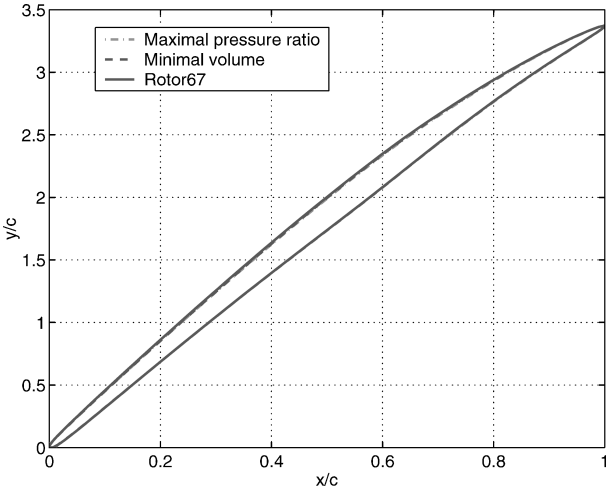


a) Airfoil shape

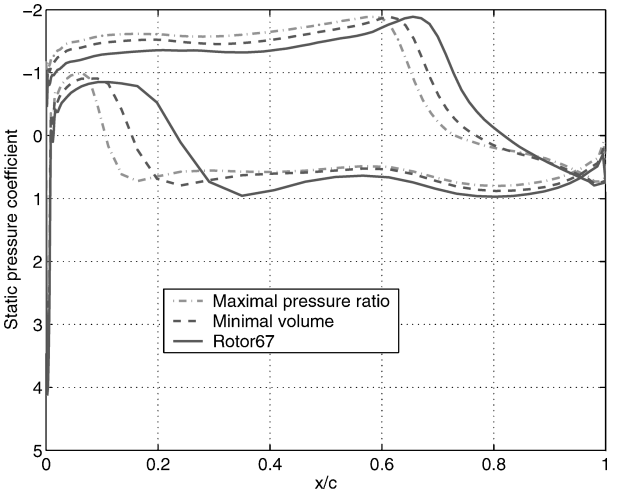


b) Pressure distribution

Fig. 10 Comparison between the new designs and rotor67 at the 10% span from the hub.

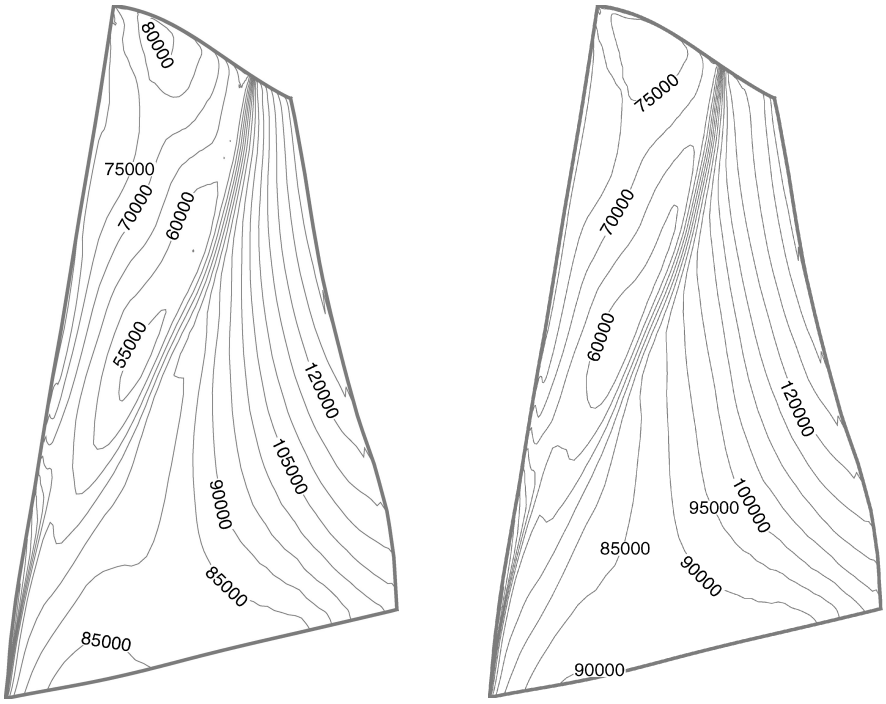


a) Airfoil shape



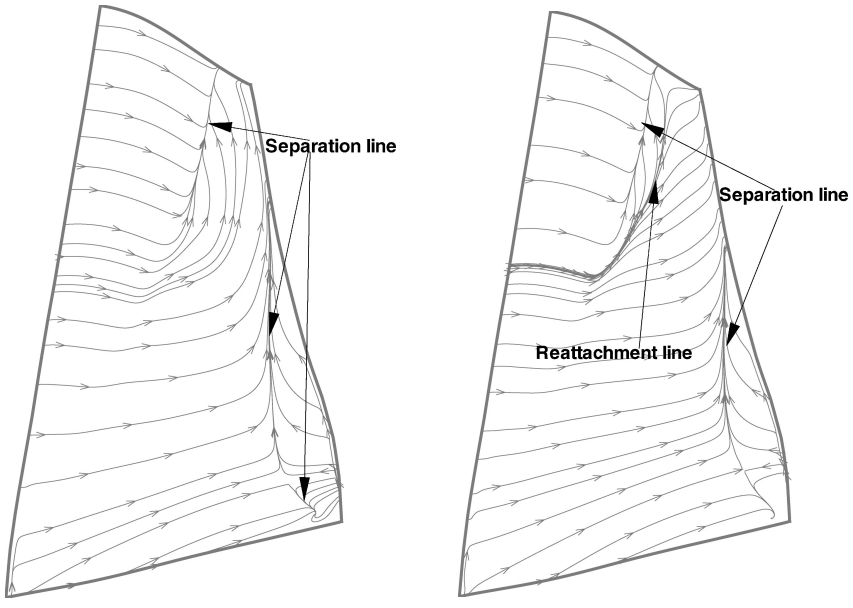
b) Pressure distribution

Fig. 11 Comparison between the new designs and rotor67 at the 90% span from the hub.



a) Rotor67 b) High-pressure-ratio design

Fig. 12 Pressure contours on the blade suction side.



a) Rotor67 b) High-pressure-ratio design

Fig. 13 Streamlines close to the blade suction side.

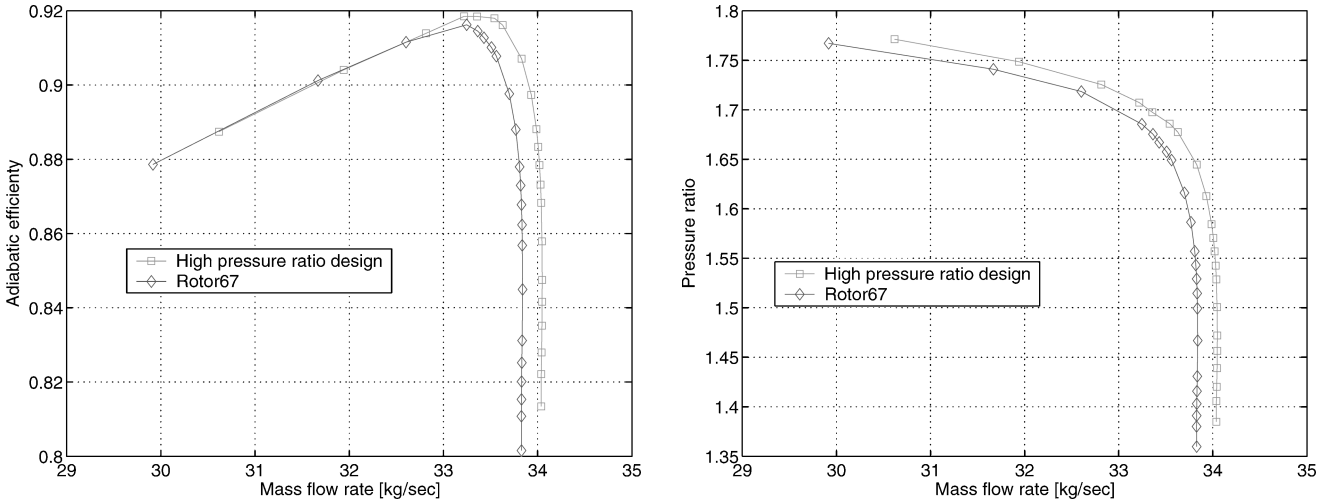
Table 7 Rotor67 design performance

Design	Mass flow, kg/s	Isentropic efficiency	Pressure ratio	Entropy production
Low-pressure ratio	33.515	0.91938	1.6848	0.009107
High-pressure ratio	33.415	0.91897	1.6965	0.009248
Rotor67	33.446	0.91234	1.6661	0.009699

at the low and high ends of the pressure ratio. Both designs have a higher isentropic efficiency than the rotor67. The high-pressure-ratio design has slightly less mass flow rate than the rotor67 design while the low-pressure-ratio design has a larger one.

Airfoil profiles at two spanwise positions are shown in Figs. 10 and 11, representing the 10% span and 90% span, respectively. The static-pressure coefficient profiles are also shown. At the 10% span, the high-pressure-ratio design has a larger camber but smaller thickness than the rotor67 design. At the 90% span, the high-pressure-ratio design has a smaller camber than the rotor67 and has a thinner airfoil. The shape change is large at the root, and it gradually decreases when approaching the tip. However, the change of pressure distribution is large at the tip and small at the root. This is because that transonic flow is more sensitive to the shape change than subsonic flow. Figure 11 shows that the new designs have a more forward shock than the rotor67 design.

We use the pressure distribution on the suction side to interpret the flow patten. Figure 12 compares the pressure distribution between the high-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 with our finding in Fig. 11, the new design has a more forward shock than the rotor67. After the shock, flow separates as a result of shock-boundary-layer interaction. This separation is



a) Isentropic efficiency **b) Pressure ratio**

Fig. 14 Performance map comparison between rotor67 and the high-pressure-ratio design.

evident in Fig. 13, too. 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 design, in which the flow does not reattach at the trailing edge after separation, the new design has a smaller separation zone after the shock, which is partially responsible for its higher-stage pressure ratio.

Figure 14a demonstrates the performance map of the rotor67 and the high-pressure-ratio design. Close to the operating condition, the new design has higher isentropic efficiency than the rotor67 design. Near the stall condition, the new design allows larger mass flow rate than the rotor67 design. Figure 14b shows that at the same flow rate the pressure ratio of the new design is larger than that of the rotor67 design.

VIII. Conclusions

We brought together design of experiment, response surface method, a genetic algorithm, and computational-fluid-dynamics analysis codes to provide an efficient system for multiobjective design problems. The improved hypercube sampling was used for the design of experiments that were characterized with system error. The genetic algorithm facilitated the exploration of multiple Pareto-optimal solutions and maintained the global optimization properties. The response surface approximation greatly reduced the computational cost while still maintaining a good predictive capability. Instead of verifying all of the Pareto-optimal solutions from the response surface approximation, we selected a subset of solutions using the K-means clustering algorithm, which accurately reflected the solution distribution. The proposed system was used in the redesign of a single-stage turbopump, a two-stage turbopump, and the NASA rotor67 blade. For the single-stage pump design, we gained an improvement of 1.2% in the total head rise, and for two-stage pump we achieved an improvement of 0.5%. Both improvements were made with the same power input as the original designs. For the rotor67 design we achieved an improvement of 1.8% in the pressure ratio and 6.2% decrease in entropy production. This improvement was partially realized through reducing the separation zone in the trailing edge. With our proposed approach, these improvements were made at a much reduced computational cost.

Acknowledgments

The authors thank the Computational Thermo-Fluids Laboratory at the University of Florida for providing the computational resources. This work is supported by NASA Research Grant NAG3-2869 under the Ultra Efficient Engine Technology Program, managed by Carol Ginty. The reviewers gave good suggestions to improve our work.

References

[1]Oyama, A., Liou, M. S., and Obayashi, S., "Transonic Axial-Flow Blade Shape Optimization Using Evolutionary Algorithm and Three-Dimensional Navier–Stokes Solver," *Journal of Propulsion and Power*, Vol. 20, 2004, pp. 612–619.

[2]Benini, E., "Three-Dimensional Multi-Objective Design Optimization of a Transonic Compressor Rotor," *Journal of Propulsion and Power*, Vol. 20, No. 3, 2004, pp. 559–565.

[3]Oyama, A., and Liou, M. S., "Multiobjective Optimization of Rocket Engine Pumps Using Evolution Algorithm," *Journal of Propulsion and Power*, Vol. 18, No. 3, 2002, pp. 528–535.

[4]Lian, Y., Liou, M. S., and Oyama, A., "An Enhanced Evolutionary Algorithm with a Surrogate Model," *Proceedings of Genetic and Evolutionary Computation Conference–GECCO 2004* [CD-ROM], Springer-Verlag, 2004.

[5]Deb, K., *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, New York, 2001.

[6]Knill, D. L., Giunta, A. A., Baker, C. A., Grossman, B., Mason, W. H., Haftka, R. T., and Watson, L. T., "Response Surface Methods Combining Linear and Euler Aerodynamics for Supersonic Transport Design," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 75–86.

[7]Madsen, J. I., Shyy, W., and Haftka, R. T., "Response Surface Techniques for Diffuser Shape Optimization," *AIAA Journal*, Vol. 38, No. 9, 2000, pp. 1512–1518.

[8]Keane, A. J., "Wing Optimization Using Design of Experiment, Response Surface, and Data Fusion Methods," *Journal of Aircraft*, Vol. 40, No. 4, 2003, pp. 741–750.

[9]Myers, R. H., and Montogomery, D. C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Wiley, New York, 2002.

[10]Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–435.

[11]McKay, M. D., Bechman, R. J., and Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.

[12]Wang, G. G., "Adaptive Response Surface Method Using Inherited Latin Hypercube Design Points," *Journal of Mechanical Design*, Vol. 125, No. 2, 2003, pp. 210–220.

[13]Beachkofski, B., and Grandhi, R., "Improved Distributed Hypercube Sampling," AIAA Paper 2002-1274, 2002.

[14]Carpenter, W. C., "Effect of Design Selection on Response Surface Performance," NASA CR-4520, 1993.

[15]Unal, R., Lepsch, R. A., and McMillin, M. L., "Response Surface Model Building and Multidisciplinary Optimization Using D-Optimal Design," AIAA Paper 98-4759, 1998.

[16]Fonseca, C. M., and Fleming, P. J., "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion, and Generalization," *Genetic Algorithms: Proceedings of the 5th International Conference*, edited by S. Forrest, Morgan Kaufmann, San Mateo, CA, 1993, pp. 416–423.

[17]De Jong, K. A., "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," Ph.D Dissertation, Univ. of Michigan, Ann Arbor, MI, 1975.

[18]Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learing*, Addison Wesley Longman, Reading, MA, 1989, pp. 191–195.

[19]Booker, L. B., "Improving Search in Genetic Algorithms," *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann, San Mateo, CA, 1987, pp. 1–63.

[20]Veres, J. P., "Centrifugal and Axial Pump Design and Off-Design Performance Prediction," NASA TM-106745, 1994.

[21]Walter, S. C., William, S., and Donald, C. U., "Design and Performanceof a 427-Meter-Per-Second-Tip-Speed Two-Stage Fan Having a 2.40 Pressure Ratio," NASA TP-1314, 1978.

[22]Arnone, A., Liou, M. S., and Povinelli, L. A., "Multigrid Calculation of Three-Dimensional Viscous Cascade Flows," NASA TM-105257, 1991.

[23]Arnone, A., "Viscous Analysis of Three-Dimensional Rotor Flow Using a Multigrid Method," *Journal of Turbomachinery*, Vol. 116, No. 3, 1994, pp. 435–445.

[24]Bishop, C. M., *Neural Network for Pattern Recognition*, Oxford Univ. Press, 2003, pp. 187–189.

E. Livne
Associate Editor