Multi-Objective Aerodynamic Optimization of Axial Turbine Blades Using a Novel Multilevel Genetic Algorithm

Özhan Öksüz
Department of Aerospace Engineering,
Middle East Technical University,
Ankara, Turkey;
TUSAŞ Engine Industry, Inc. (TEI),
Eskisehir 26003, Turkey
e-mail: ozhan.oksuz@tei.com.tr

İbrahim Sinan Akmandor
Department of Aerospace Engineering,
Middle East Technical University;
Pars Makina Ltd.,
ODTU-OSTIM Teknokent,
Ankara, Turkey
e-mail: sinan.akmandor@parsmakina.com

In this paper, a new multiploid genetic optimization method handling surrogate models of the CFD solutions is presented and applied for a multi-objective turbine blade aerodynamic optimization problem. A fast, efficient, robust, and automated design method is developed to aerodynamically optimize 3D gas turbine blades. The design objectives are selected as maximizing the adiabatic efficiency and torque so as to reduce the weight, size, and cost of the gas turbine engine. A 3D steady Reynolds averaged Navier–Stokes solver is coupled with an automated unstructured grid generation tool. The solver is verified using two well-known test cases. The blade geometry is modeled by 36 design variables plus the number of blade variables in a row. Fine and coarse grid solutions are respected as high- and low-fidelity models, respectively. One of the test cases is selected as the baseline and is modified by the design process. It was found that the multiploid multi-objective genetic algorithm successfully accelerates the optimization and prevents the convergence with local optimums. [DOI: 10.1115/1.3213558]

1 Introduction

The design of today's modern high performance turbo engine components requires fast but detailed design tools. On the other hand, the flow in a turbomachine is extremely complex because of the presence of blade boundary layers, interaction of rotating and nonrotating blades with the upstage wakes, development of secondary flow vortices, and overall periodic but still inherently unsteady flow.

Advanced CFD solvers are capable of analyzing 3D, viscous, transonic, and turbulent flows. The increase in the number of CFD calculations and the complexity of the geometries brings the fact that the design engineer needs to handle hundreds of design parameters and thousands of CFD solutions for different geometries at different boundary conditions. A way to solve this problem is to define an optimization problem, which translates the design engineer's decision criteria and experiences into a mathematical formulation that a computer can handle.

To solve this optimization problem, an automated, robust, and fast global optimization technique needs to be employed while preserving the accuracy of the optimization.

Multi-objective genetic algorithms (MOGAs) are the most popular among these global optimization techniques, including multi-objective aerodynamic shape optimization [1,2], multidisciplinary optimization of wings [3,4], rotor blade design [5], axial compressor optimization [6], turbine blade cooling [7], and thermodynamic optimization of turbojets [8].

Although MOGAs are able to converge to the global optimum, they require many objective function evaluations. For the optimization problem of 3D aerodynamic design of turbine blades, accurate objective function evaluations require computationally expensive high-fidelity Reynolds averaged Navier–Stokes (RANS) solutions. Hoping the rapid increase in computing power in the previous decade, genetic algorithm (GA) optimizations coupled with 3D RANS solvers appeared in the literature [9,10]. However, the computational power of computers converged to a degree where investments for further improvements are really infeasible. On the contrary, new sophisticated commercial CFD analysis tools demand more and more computational power to solve complex problems.

This drawback of genetic algorithms was avoided by using faster but less accurate low-fidelity 2D solvers such as a Euler [11] or a Euler/boundary layer coupled flow solvers [8] in the past. Additionally, parallel processing is implemented to reduce computational time of one GA optimization cycle.

Furthermore, several efforts have been made over recent years, particularly using MOGAs, applying surrogate models of accurate solvers. The most popular ones are response surface methodology, the Kriging model, design of experiments, neural networks, and the support vector machines. Ratle [12] examined a strategy for integrating GAs with Kriging models. This work uses a heuristic convergence criterion to determine when an approximate model must be updated. The same problem was revisited by El-Beltagy et al. [13], where the issue of balancing the design of experiments is addressed. Jin et al. [14] presented a framework for coupling GAs and neural network-based surrogate models. This approach uses both the expensive and approximate models throughout the search, with an empirical criterion to decide the frequency at which each model should be used. In Song [15], a real-coded GA was coupled with Kriging in structural optimization. A recent study of Pierret [16] presented the design of turbomachinery blades by means of function approximation. The concept is based on the use of online trained artificial neural network (ANN) as a surrogate model of a RANS solver with a GA optimization algorithm.

As long as a surrogate model is used interchangeably with the high-fidelity (exact solution) model during a MOGA optimization, the stationary optimization problem becomes dynamic. Despite these applications of surrogate models with MOGAs, evolutionary algorithms face a big convergence problem when solving dynamic optimization problems (DOPs) [17]. In order to enhance the performance of GAs for DOPs, several approaches have been developed [18], such as random immigrants [19,20], hypermutation [21], memory [22–26], and multipopulation schemes [27]. Further literature review about the subject can be found in Ref. [17]. All

Contributed by the International Gas Turbine Institute of ASME for publication in the Journal of Turbomachinery. Manuscript received March 25, 2009; final manuscript received April 15, 2009; published online May 4, 2010. Review conducted by David Wisler. Paper presented at the ASME Turbo Expo 2008: Land, Sea and Air (GT2008), Berlin, Germany, June 9–13, 2008.

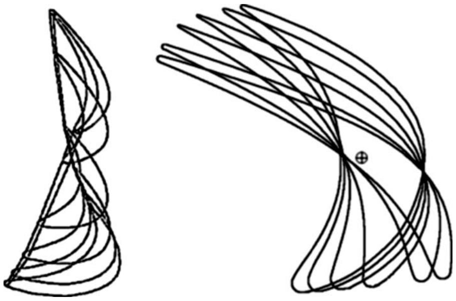


Fig. 1 (*a*) Stacking layers and (*b*) stacking layers on centroid

of these studies prove the fact that simple multi-objective GAs are not suitable as they are for dynamically changing environments without any modification to the GAs themselves. Therefore, MOGA optimization methods proposing interchangeable surrogate models with high-fidelity models have the problem of convergence.

The "multiploid" GAs have successfully solved the convergence problem of traditional GAs in dynamic environments [28–30]. However, the aerodynamic optimization problem differs from these multiploid GA applications in the fact that:

-the surrogate model already exists
-the problem environment is static

Consequently, in this paper, in order to use a limited computational budget efficiently, without degrading the optimization quality, the existing surrogate model is implemented with a modified version of multiploid GAs. First, the low-fidelity surrogate model available for the aerodynamic performance prediction of a turbine blade is selected as the coarse mesh solution. Then, a distance based MOGA is modified to yield a multiploid genetic representation in order to implement high- and low-fidelity models simultaneously and to avoid a convergence problem to the global optimum. Finally, the optimization is performed for two different maximizing objectives: adiabatic efficiency and torque.

2 3D Blade Shape Parameterization

The geometric model of the 3D blade profile must be defined in as few as possible parameters in order to simplify the optimization problem. On the other hand, a robust geometric model should be able to cover distinct blade profiles in every corner of the design space, almost entirely, while generating realistic blade profiles. Consequently, the parameterization of the blade profile should minimize the possibility of generating geometrically unrealistic profiles.

In that sense, the 3D blade is represented by several 2D cylindrical surface layers stacked along the centroids (center of gravities) based on a hub layer centroid location in the tangential direction, as shown in Fig. 1. Another choice would have been to stack the blade cross-sectional layer through their respective center of pressures, but stacking on centroids has the main advantage of satisfying mechanical constraints to reduce bending stresses under rotational loads. Additionally, since a linear stacking line passing through the centroids of the layers is used, additional design variables that may be required for defining a stacking curve are not necessary. This also helps to avoid increasing the number of design variables due to additional stacking curve parameters.

The number of layers, the spanwise locations of the layers, and the layer angles between the layer plane and the axial direction might be included in the design parameters. However, presetting these variables avoids a dramatic increase in design parameters. Moreover, for the design point of view, a focus on aerodynamic parameters such as blade angles rather than aforementioned geometric parameters is more suitable for the design engineer.

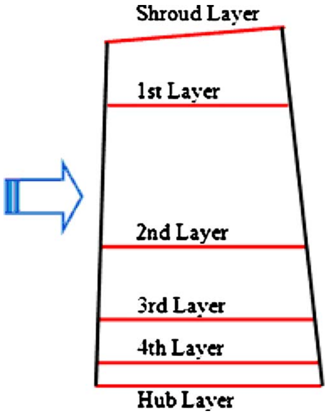


Fig. 2 Six layers in the meridional view

For these reasons, the number of layers is fixed and selected as six. The spanwise locations of the layers are taken to be constant. The midlayers' spanwise locations are set equal to that of the baseline blade, and the hub and shroud layers are set as the hub and tip sections of the blade, respectively. The shroud layer plane is set in the shroud axis whereas the remaining layers' planes are set in the axial direction in this study. Consequently, the selected layers in the meridional view and their names are shown in Fig. 2.

2.1 Parameterization of Layers.

The most basic and popular airfoil definition method is the NACA family [31] in which the blade is defined by a mean camber line and a thickness distribution. These airfoils cannot be applied to an optimization problem due to the limited number of airfoils. A recent study shows an airfoil modification algorithm for layers shaped by adding a thickness distribution to the camber line [32]. However, this method uses only four design parameters and relies on the baseline camber and thickness distribution.

Pritchard [33] implemented an 11 variable airfoil with a circular arc and third-order polynomial curves tangent to each other. However, polynomial curves suffer from discontinuities due to inflection points as well. Trigg et al. [34] improved this method by replacing the polynomial curves with Bezier curves and increased the number of variables to 17. Anders and Haarmeyer [35] represented the airfoil with two fifth-order Bezier curves, which requires 20 parameters. Yamamoto and Inoue [36] used cubic B-splines of camber line and a thickness distribution for wing sections.

In this study, the design engineers' traditional aerodynamic design parameters, such as leading and trailing edge radii, and inlet and outlet metal angles, are respected while generating the layer shapes, since these parameters are often imposed by manufacturing mechanical or aerodynamic constraints. The layer is parameterized using five angles, as shown in Fig. 3. These angles are the wedge angles of leading and trailing edges (W_{LE} and W_{TE}), the blade inlet and exit metal angles (β_{LE} and β_{TE}), and the stagger angle (ξ). Additionally, the circumferential rotation angle Θ is selected as the sixth design parameter since this angle sets the lean of the blade in three-dimensional geometry.

Consequently, there are six design parameters selected for a layer geometry shaping. Since there are six predetermined layers and one more design parameter, which is the number of blades, a total number of 37 design parameters is used to define one particular 3D blade profile.

In order to generate the layer profiles, four different curves are used. Two Bezier curves are selected to represent pressure and suction sides of the layer and the remaining two ellipse cuts for leading and trailing edges. The continuity up to the second-order derivative is ensured at the junction points of these curves. A complete description of curves and surfaces can be found in the

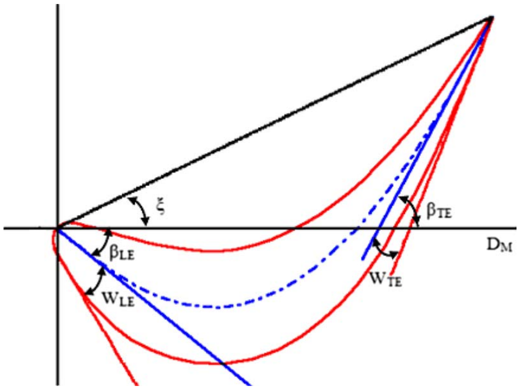


Fig. 3 Layer geometric parameterization

literature [37,38]. The main reason of using Bezier curves instead of polynomials is that although inflection points can occur with Bezier curves, they are far less than polynomials. Moreover, the curve definition is simpler, and the degree of the curve can be very easily increased. More importantly, the parametric form of the equations allows obtaining the coordinates of the blade points at any point [16].

2.2 Blade Profile Reshaping Algorithm. The blade profile reshaping algorithm is given in Fig. 4. According to the algorithm, there are two main steps. In the first step, the baseline blade geometry is parameterized. In the second step, the baseline blade geometry parameters are modified according to the design parameters.

3 Multifidelity Aerodynamic Models

A commercial RANS flow solver, ANSYS® CFX® software [39], is used in this study for evaluating the objective functions. The code is capable of solving a three-dimensional steady state compressible viscous Reynolds averaged Navier–Stokes equations in conservation form.

A zero-equation turbulence model is used in this study in order to keep robustness and to decrease sophistication [40]. It is simple to implement and use, is computationally low cost, can produce approximate results very quickly, and provides a good guess. The zero-equation turbulence model in the solver uses an algebraic equation to calculate the viscous contribution from turbulent eddies. A constant turbulent eddy viscosity is calculated for the entire flow domain.

For 3D meshing, a mesh generator, ANSYS® TURBOGRID™ software [41], is used for unstructured automatic volume meshing. Mainly, the program automatically generates a triangular surface mesh and the interior tetrahedral mesh using the Octree approach [42]. Automatic meshing of the volume is done by successive refinement until all grid density requirements are met. The resulting tetrahedral mesh is an adaptive mesh of nonuniform density.

In leading and trailing edge regions of high surface curvature, the mesh is automatically refined in order to maintain the necessary geometric resolution. This improves the robustness and eliminates the additional user input.

The flow adjacent to suction and pressure side walls are characterized by high flow variable gradients in the normal direction. Moreover, the boundary layer development in the turbine blade is the main factor affecting the efficiency of the blade. While tetrahedral cells can be used in such boundary layers, greater accuracy can be achieved using prism elements. The meshing software automatically arranges layers of prism elements near the boundary surfaces in order to appropriately model close to wall physics. Prisms are generated by extruding the triangular surface mesh. The prismatic mesh is inflated by a prescribed number of layers, which is set as ten layers in this study.

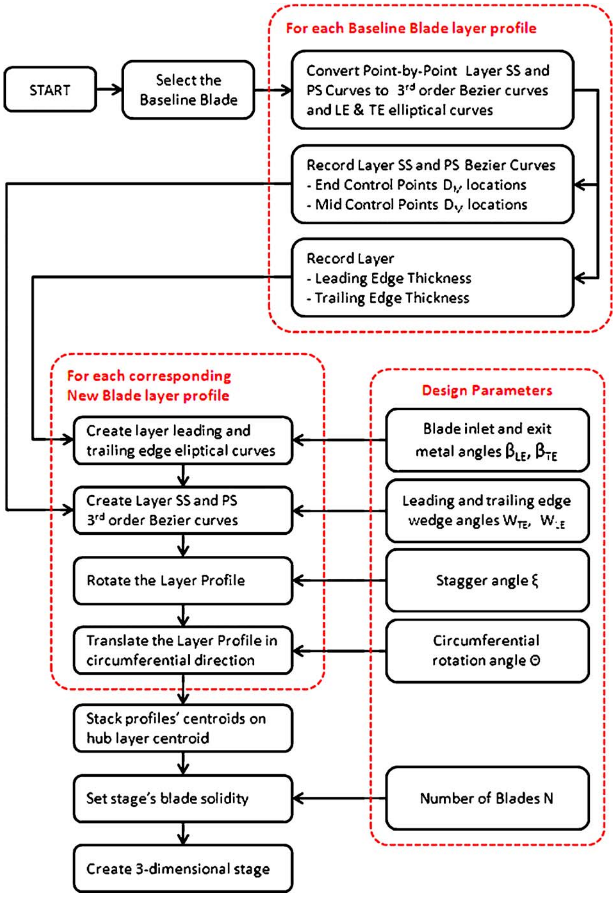

For each Baseline Blade layer profile
START
Select the Baseline Blade
Convert Point-by-Point Layer SS and PS Curves to 3rd order Bezier curves and LE & TE elliptical curves
Record Layer SS and PS Bezier Curves
- End Control Points D_V locations
- Mid Control Points D_V locations
Record Layer
- Leading Edge Thickness
- Trailing Edge Thickness
For each corresponding New Blade layer profile
Create layer leading and trailing edge eliptical curves
Create Layer SS and PS 3rd order Bezier curves
Rotate the Layer Profile
Translate the Layer Profile in circumferential direction
Stack profiles' centroids on hub layer centroid
Set stage's blade solidity
Create 3-dimensional stage
Design Parameters
Blade inlet and exit metal angles β_{LE}, β_{TE}
Leading and trailing edge wedge angles W_{TE}, W_{LE}
Stagger angle ξ
Circumferential rotation angle Θ
Number of Blades N

Fig. 4 Blade profile reshaping algorithm

As a result, a hybrid tetrahedral grid consisting of prism elements near the boundary surfaces and tetrahedral elements in the interior of the space with refined leading and trailing edge cells is automatically meshed.

Since the automatic meshing of the volume is done by successive refinement, a predefined grid refinement level is used to meet all grid density requirements. The mesh quality represents a tradeoff between the accuracy of the solution and computational cost.

In this paper, two different grid refinement levels is used for a low quality and a high quality mesh generation, which are called as *coarse grid* and *fine grid*, respectively. The low-fidelity objective function value is calculated using coarse grid RANS solutions, whereas high-fidelity value is calculated using fine grid.

The two selected experimental test cases are relevant to three-dimensional flow calculations for viscous flows in axial turbine calculations: VKI nozzle [43] and UH stage [44]. The VKI low speed annular turbine blade row test case is used to validate radial-tangential downstream plane flow variables and to investigate grid quality effect on the flow field. The UH four-stage low speed annular turbine blade row test case results are used to validate spanwise stage performance parameters for only the first stage. This test case is used to demonstrate the off-design performance predictions of the RANS solver.

3.1 VKI Nozzle Test Case. The VKI test case is a low speed, low aspect ratio, and high speed annular nozzle guide vane. The blades have a constant profile over the blade height and are untwisted. To account for differences in the upstream flow conditions for an inlet guide vane and an intermediate stage vane, the annular cascade was tested with skewed inlet end wall boundary layers. The inlet skew was generated by rotating the upstream hub

Table 1 Comparison of coarse/fine grid parameters

	Coarse grid	Fine grid
Number of nodes in surface mesh	7946	31,597
Number of faces in surface mesh	17,292	63,194
Number of nodes in volume mesh	81,885	454,785
Number of tetrahedral elements in volume mesh	263,107	1,955,673
Number of prismatic elements in volume mesh	64,420	206,570
Number of elements in combined mesh	327,527	2,162,243

Table 2 Comparison of low-/high-fidelity objective function calculation CPU times

	Low-fidelity	High-fidelity
Geometry generation (s)	10	10
Grid generation (s)	157	705
Solver run (s)	827	3564
Total CPU time (min)	16.6	71.3

end wall.

The nozzle is solved using test case boundary conditions of upstream inlet total temperature, total pressure, and swirl angle, and the downstream static pressure using coarse and fine grids. The coarse and fine grid generation parameters are listed in Table 1. For both of the cases, the automatic grid is generated with ten layers of prismatic boundary cells with an expansion factor of 1.2. The aspect ratio of all elements is below 5. Corresponding generated grids are shown for coarse and fine meshes in Figs. 5 and 6, respectively.

For the solution of the test case, standard air is used as the working fluid. The same settings and methods used in the optimization problems are set for the solver such as the turbulence model, wall roughness, and so on. The calculated average values of Reynolds number and turbulence are 1.5×10^5 and 7%, respectively.

The CPU times for the low- and high-fidelity objective function value evaluations are tabulated in Table 2. It is shown that a low-fidelity coarse mesh solution is more than four times faster than the high-fidelity fine mesh solution. Although the accuracy of the coarse mesh solution is lower than the fine mesh solution, the solution time is considerably faster for the former, therefore, a very suitable low-fidelity model of the objective function solution for use in the optimization method was developed in this study.

The VKI test case upstream flow conditions are measured at a distance of $X/C_{\text{ax}}=-0.70$, whereas the rotating hub extends from $X/C_{\text{ax}}=-0.15$ to $X/C_{\text{ax}}=-4.16$. The numerical model takes into account only the measured upstream flow conditions as the inlet conditions of the problem, and the rotating hub is not modeled at all. Therefore, the remaining part of the rotating hub after the measurement plane could not be included in the model in any way.

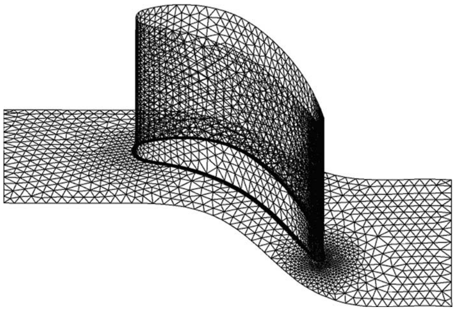


Fig. 5 VKI test case-coarse mesh

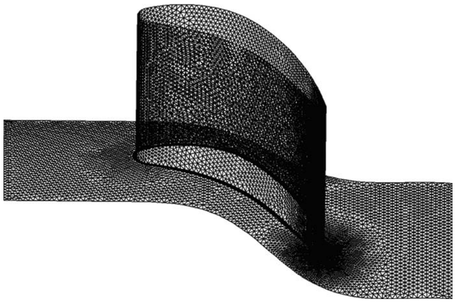


Fig. 6 VKI test case-fine mesh

The VKI test case measurement planes of $X/C_{\text{ax}}=0.86$ and $X/C_{\text{ax}}=1.11$ are shown in Fig. 7. Contour plots of the static and total pressure coefficients CP_s and CP_0 at the measurement planes of $X/C_{\text{ax}}=0.86$ and $X/C_{\text{ax}}=1.11$ are shown in Figs. 8 and 9, respectively. Contour plots of blade to blade exit flow angle β at the measurement planes of $X/C_{\text{ax}}=0.86$ and $X/C_{\text{ax}}=1.11$ are shown in Fig. 10. These figures represent the exceptionally strong three-dimensional features of the flow field.

According to Figs. 8 and 9, both CP_s and CP_0 values are in very good agreement with coarse and fine mesh solutions. The hub and shroud wall surface region measurements have steeper gradients with respect to calculated results. This is attributed to the simplicity of the selected wall roughness and the turbulence model of the solver. When compared with the fine mesh solution, the coarse mesh solution resolution is somewhat lower, but at an acceptable degree. The results have the same order of magnitude and topology. It can also be said that the measurement plane location has no significant effect on the flow parameter prediction.

According to Fig. 10, the exit flow angle β values are in very good agreement with the coarse and fine mesh solutions. When compared with the fine mesh solution, the coarse mesh solution resolution is lower but still at an acceptable degree. The results have the same order of magnitude and topology. It can also be said that the measurement plane location has no significant effect on the exit flow angle parameter prediction.

Consequently, the VKI test case demonstrates the ability to predict stage losses reasonably well, which in turn is used to calculate the efficiencies. Flow turning is directly related to torque calculations and therefore accurate predictions of the flow exit angles are essential. The solution method is validated against the VKI test case, which represents an exceptionally strong flow turning. The coarse/fine mesh calculations showed that although the resolution of the calculated contours are lower for the coarse mesh, the calculated flow field has the same topology with the same order of magnitude as of the fine mesh.

3.2 UH Four-Stage Low Speed Turbine. The UH turbine is designed for a rotational speed of 7500 rpm and a mass flow rate of 7.8 kg/s, which is set with the aid of bypass. The blading is of the free-vortex type with a 50% degree of reaction at the middle section of the last stage. A tip clearance of 0.4 mm is used for the rotor. Only throughflow radial traverse measurements at the up-

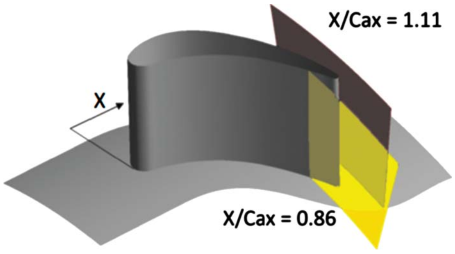
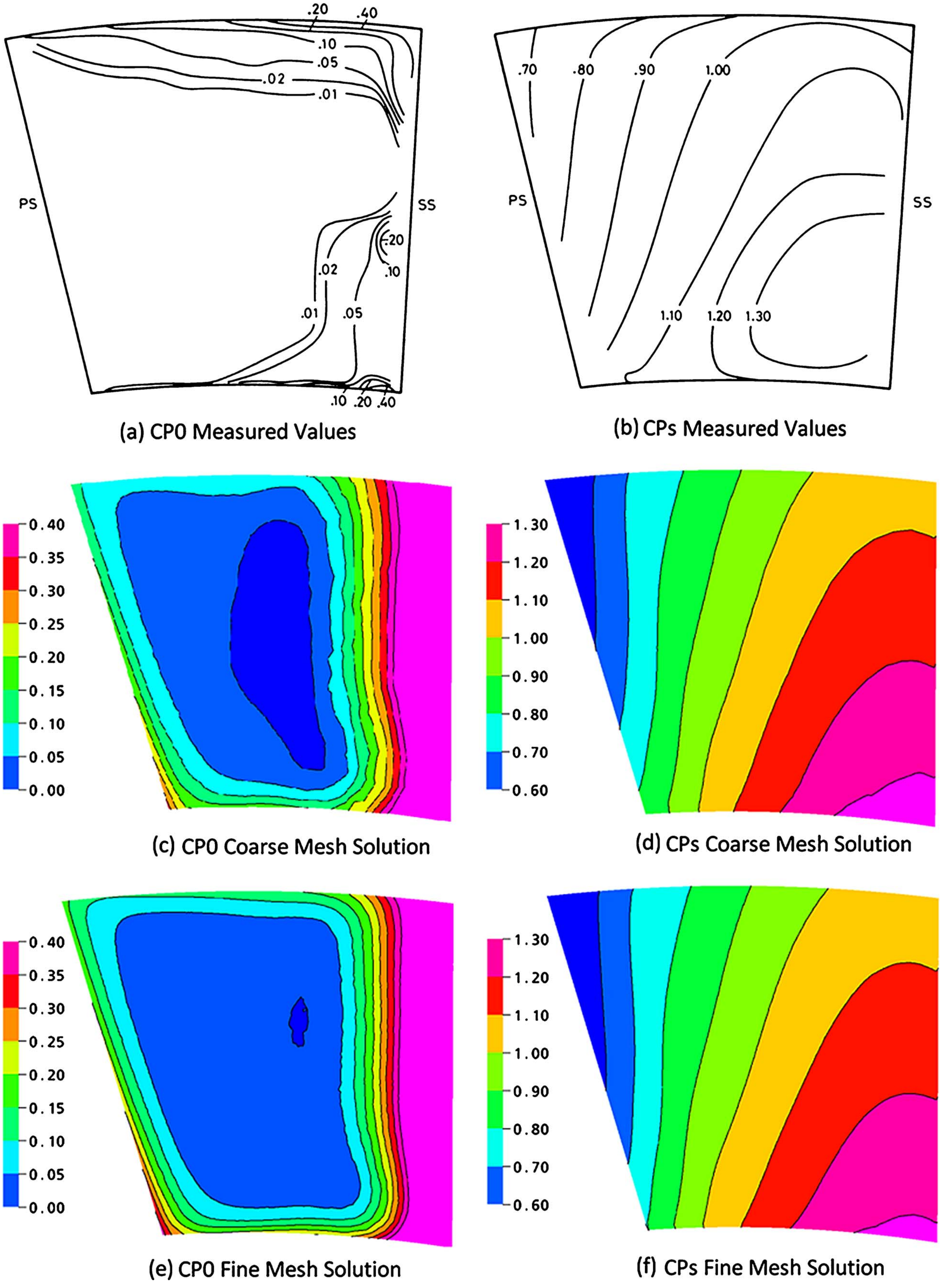


Fig. 7 VKI test case planes of $X/C_{\text{ax}}=0.86$ and $X/C_{\text{ax}}=1.11$



(a) CP0 Measured Values

(b) CPs Measured Values

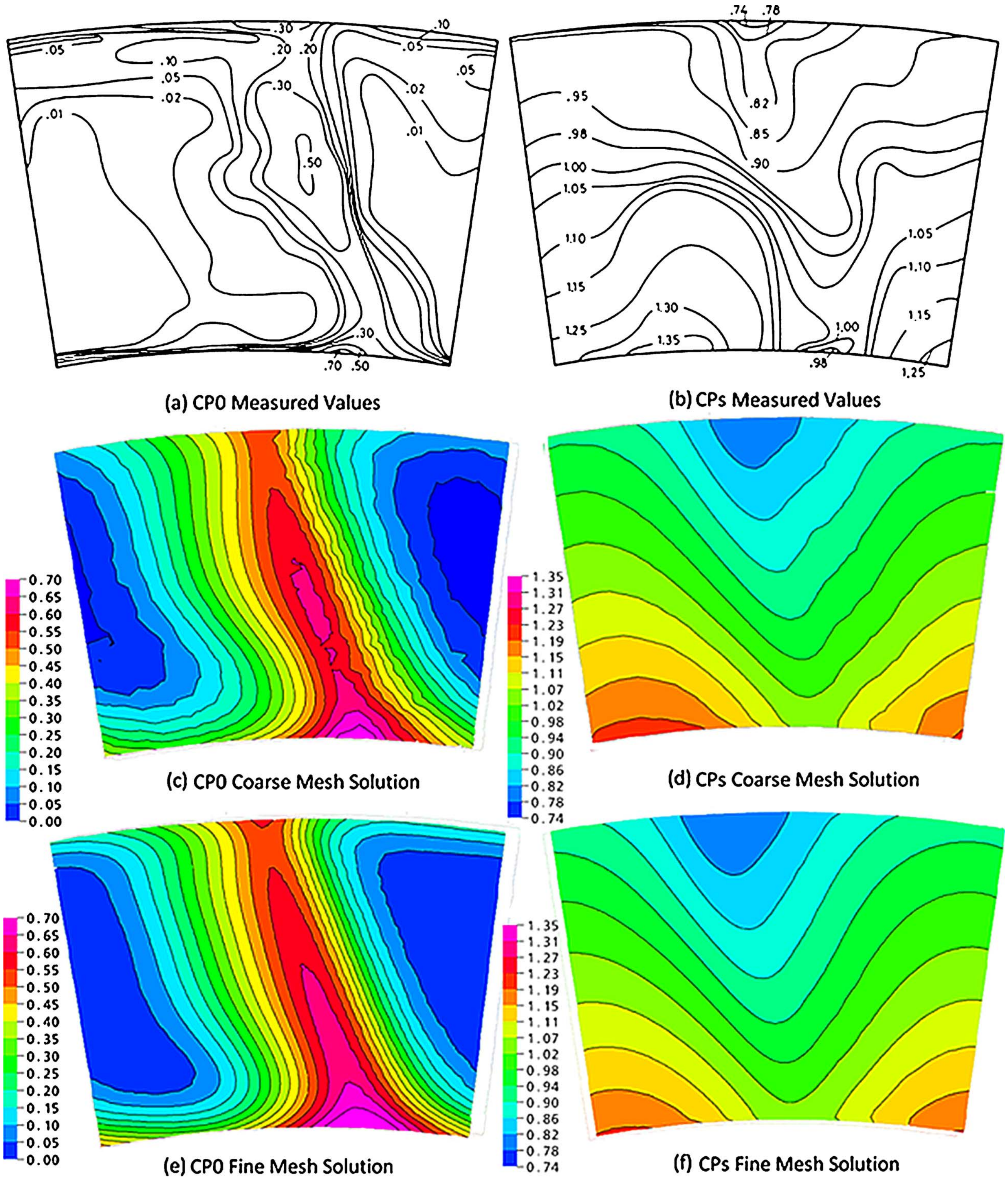
(c) CP0 Coarse Mesh Solution

(d) CPs Coarse Mesh Solution

(e) CP0 Fine Mesh Solution

(f) CPs Fine Mesh Solution

Fig. 8 The comparison of low-/high-fidelity CP_0 and CP_s values with measurements at plane $X/C_{ax}=0.86$



(a) CP0 Measured Values

(b) CPs Measured Values

(c) CP0 Coarse Mesh Solution

(d) CPs Coarse Mesh Solution

(e) CP0 Fine Mesh Solution

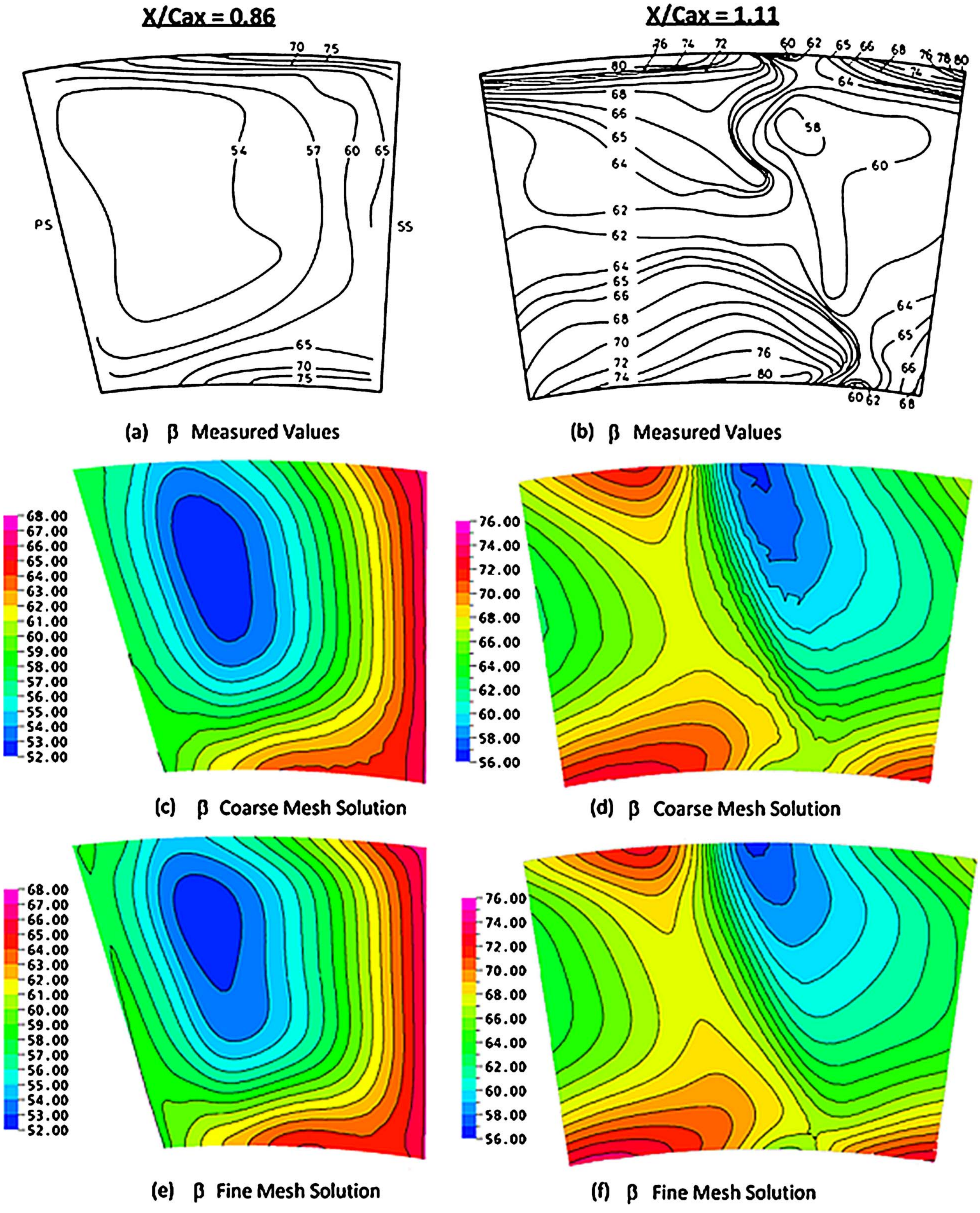
(f) CPs Fine Mesh Solution

Fig. 9 The comparison of low-/high-fidelity CP_0 and CP_s values with measurements at plane X/C_{ax}=1.11

stream of the stator and the downstream of the rotor were made. Total pressure p_t, static pressure p, total temperature T_t, and flow angle β are measured for design and off-design conditions. The stage is solved using test case boundary conditions measured at the upstream inlet total temperature, total pressure, and swirl angle, and the mass flow rate using frozen rotor approach. Only coarse mesh solution is used for the validation of the stage flow field, as shown in Fig. 11.

The stator is solved with test case specified inlet and exit boundary conditions, first. Then, the rotor is solved using outlet flow conditions of the stator as the upstream boundary conditions and setting the same mass flow rate. Finally, the test case is solved for design and six off-design conditions, as given in Table 3. For the solution of the test case, standard air is used as the working fluid. The calculated average values of Reynolds number and turbulence is 1.5×10^5 and 7%, respectively.

Figure 12 shows the measured and computed total pressure distributions along the span at the downstream of the rotor blade. The measurements are made in nine radial locations. It is seen that the



(a) β Measured Values

(b) β Measured Values

(c) β Coarse Mesh Solution

(d) β Coarse Mesh Solution

(e) β Fine Mesh Solution

(f) β Fine Mesh Solution

Fig. 10 The comparison of low-/high-fidelity β values with measurements at planes $X/C_{ax}=0.86$ and $X/C_{ax}=1.1$

predictions are 2% to 3% lower than the measured values. This is acceptable in the uncertainty region of the measured values, which is indicated as 2%.

Figure 13 compares the measured static pressure values with respect to the calculated values at on- and off-design conditions. In any case, the calculated values are slightly lower than the measured values, and the difference is well within the uncertainty level of 2%.

4 Multiploid Multi-Objective Genetic Algorithm: MOGAXL

The MOGAXL developed in this study is based on the MOGA, which is developed by the "distance based Pareto genetic algorithm" of Osyczka and co-worker [45,46]. The MOGA is a simple haploid genotype genetic algorithm in which the genotype is referred here as composed of a single chromosome.

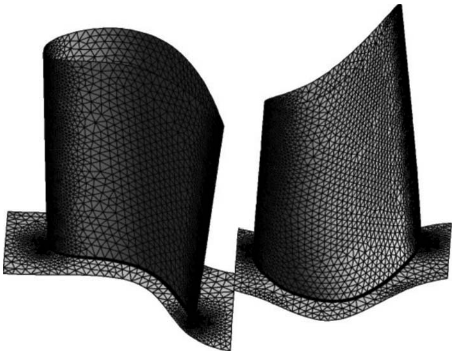


Fig. 11 UH test case-coarse mesh of stator and rotor

However, multiploid—mostly *diploid*—genotype also exists in nature, which contains two (diploid) or more sets of single chromosomes. Figure 14 shows the flowchart of a multiploid multifidelity genetic algorithm for multi-objective optimization problems (MOGAXL) developed in this study. There are major modifications to the distance based selection and crossover operators of a simple haploid MOGA to handle multifidelity objective function values. Additionally, there are two elite sets in the MOGAXL.

The XL abbreviation stands for "multilevel" fitness assignment. A higher level fitness means that the objective function values are calculated using a higher-fidelity model. On the contrary, a lower level fitness corresponds to a lower-fidelity model calculated values of the objective functions.

The initial population is randomly constructed by generating a prescribed number of blades. For each individual, a blade shape is constructed. An automatic coarse or fine grid is applied, and the RANS flow solver is run to get efficiency and torque values. The MOGAXL has two *elite member sets* (EMS-L and EMS-H), which are updated at each generation and are initially empty. The first individual of the first generation is evaluated using a low-fidelity coarse grid solution and directly copied to the low-fidelity elite member set (EMS-L), and the fitness is assigned. A *distance based fitness* value is calculated for the remaining individuals while the EMSs are updated simultaneously according to dominance principle. Only nondominated individuals are allowed in the EMSs. Finally, based on distance based low- and high-fidelity fitness values, MOGAXL operators of selection, crossover, and mutation are applied to generate the next population. The population generation cycle is repeated until a predefined maximum number of function evaluations or generations. At the end of the optimization, the EMS-H set consists of the Pareto optimal frontier individuals, which are to be presented to the decision maker.

Table 3 The UH test case experimental conditions

	Rotational speed (rpm)	Mass flow (kg/s)
On-design condition	7500	7.8
Off-design condition 1	7500	6.5
Off-design condition 2	7500	4.6
Off-design condition 3	7500	4.1
Off-design condition 4	5625	5.5
Off-design condition 5	5625	3.9
Off-design condition 6	5625	3.2

Similar to the MOGA, the MOGAXL requires an initial population of solutions (individuals) in the design space before the application of MOGAXL operators. Initiation of this population begins with the selection of two MOGA parameters and one MOGAXL parameter; they are mainly the population size and chromosome length, and the number of chromosomes, i.e., genetic structure, respectively. In this study, only a diploid genetic representation, which consists of two chromosomes, is used.

Each design variable set defines a blade configuration with two corresponding binary strings. The real value of each design variable is expressed as a string of binary digits, which is called as *binary coding*, e.g., 101101. The associated chromosomes for each blade are formed by placing the binary digits corresponding to each variable back to back in one string.

At the initial generation, the two chromosome strings are set equal to each other. The top chromosome string corresponds to a lower-fidelity fitness, and the bottom chromosome string corresponds to higher-fidelity. Consequently, for a diploid MOGAXL, the top chromosome can be interpreted as the geometry with coarse mesh, and the bottom chromosome can be interpreted as the same blade geometry with fine mesh. Note that only the lowest-fidelity fitness values are assigned at the initial generation.

4.1 Distance Based Multifidelity Fitness Assignment. Although a MOGA uses distance based fitness assignment of the objective function value as the fitness of an individual, since a MOGAXL handles multifidelity models to calculate objective function values, simple distance based fitness assignment is not possible. Therefore, a modified version of the distance based fitness assignment method is implemented in order to be able to make use of surrogate models in MOGAXL computations.

The MOGAXL has more than one elite member set. The num-

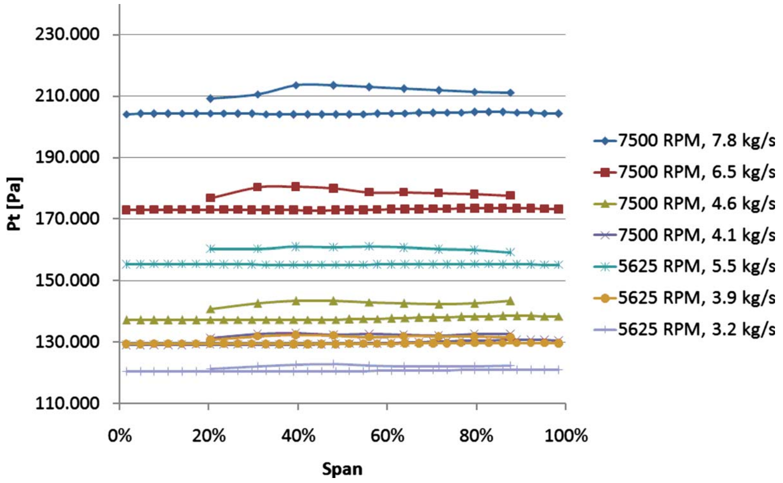


Fig. 12 The comparison of the spanwise total pressure distributions at the downstream plane of the rotor (measurements are shorter)

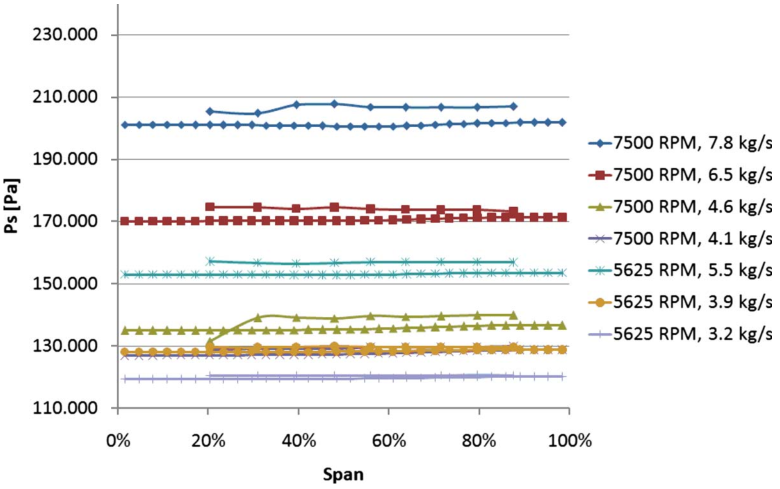


Fig. 13 The comparison of the spanwise static pressure distributions at the downstream plane of the rotor

ber of EMSs depends on the number of surrogate models. Since only one low-fidelity model is used in this thesis, two EMSs will be implemented, namely, the EMS-L and EMS-H, corresponding to lower-fidelity solution and the higher-fidelity solution, respectively. As in the case of a MOGA, only one standard population P_t exists where GA operators are performed. Among the EMSs E_t^L and E_t^H, the E_t^L contains all nondominated lower-fidelity solutions found so far, where t indicates the generation number. Similarly, E_t^H contains all nondominated higher-fidelity solutions found so far. According to the *multifidelity distance based fitness assignment method*, two (or more depending on the number of chromosomes) fitness values are assigned to each solution in P_t based on its farthest distance from the E_t^L and E_t^H members. For each EMS, the corresponding *relative distance* $d_k(\mathbf{x})$ of a P_t individual from an elite member k in EMS and the *latent potential value* p_k of an elite member k in EMS are calculated separately and are used to determine the low- and high-fidelity fitness values.

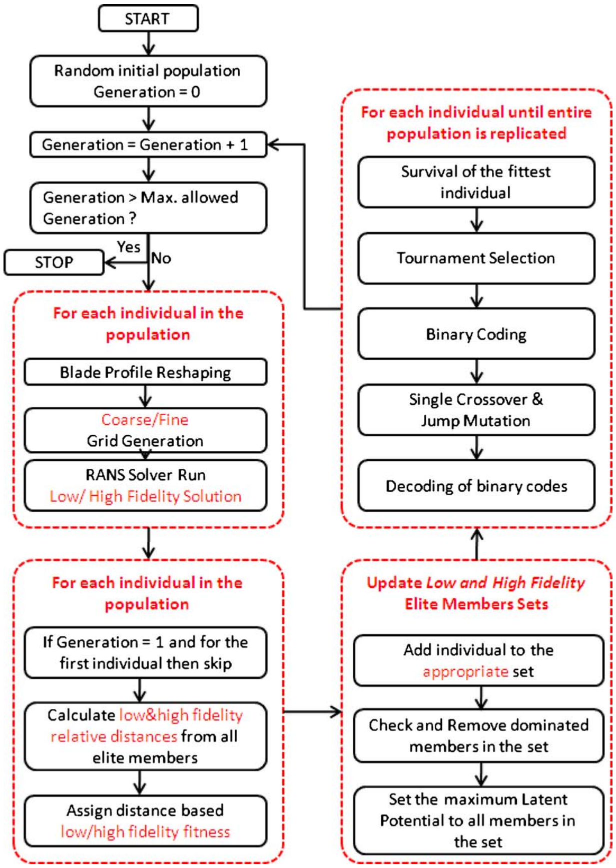

START
Random initial population
Generation = 0
Generation = Generation + 1
Generation > Max. allowed Generation ?
Yes
No
STOP
For each individual in the population
Blade Profile Reshaping
Coarse/Fine
Grid Generation
RANS Solver Run
Low/ High Fidelity Solution
For each individual in the population
If Generation = 1 and for the first individual then skip
Calculate low&high fidelity relative distances from all elite members
Assign distance based low/high fidelity fitness
For each individual until entire population is replicated
Survival of the fittest individual
Tournament Selection
Binary Coding
Single Crossover & Jump Mutation
Decoding of binary codes
Update Low and High Fidelity Elite Members Sets
Add individual to the appropriate set
Check and Remove dominated members in the set
Set the maximum Latent Potential to all members in the set

Fig. 14 Flowchart of the MOGAXL

The relative distances are calculated according to its distance from the elite set $E_t^l=\{\mathbf{e}_1^l,\mathbf{e}_2^l,\ldots,\mathbf{e}_K^l\}$, where K is the number of elite members in the lth-level EMS. The diploid MOGAXL has two l values corresponding to low-fidelity fitness and high-fidelity fitness, L and H, respectively.

For each individual in the P_t the relative distances from all elite members in E_t^l are calculated by

$$d_k(\mathbf{x}) = \sqrt{\sum_{i=1}^{I}\left(\frac{f_i^l(\mathbf{e}_k^l)-f_i^l(\mathbf{x})}{f_i^l(\mathbf{e}_k^l)}\right)^2} \quad \text{for } k=1,2,\ldots,K$$

where f^l terms represent the l-fidelity calculated values of the objective functions.

For the calculation of the l-fidelity fitness value of the individual $\mathbf{x}$, the minimum distant elite member in the E_t^l is considered, where the minimum distance is found as

$$d_{k*} = \min(d_k(\mathbf{x})) \quad \text{for all } k=1,2,\ldots,K$$

where the index k^* indicates which of the existing elite members in E_t^l is nearest to the individual $\mathbf{x}$.

After the determination of the minimum distance and the minimum distant elite member, the individual is checked whether it is dominated by the elite members of E_t^l or not. If the individual is a new Pareto solution, the E_t^l is updated by adding the new nondominated solution $\mathbf{x}$ and removing members $\mathbf{e}_k^l$, which are dominated by $\mathbf{x}$. The l-fidelity fitness and the corresponding latent potential values of the individual $\mathbf{x}$ (which is now an elite member in the E_t^l, too) are calculated using

$$F^l(\mathbf{x}) = p_{k*} + d_{k*}$$

$$p_k = F^l(\mathbf{x})$$

where $F^l(\mathbf{x})$ term represents the l-fidelity fitness values, and index k belongs to the new elite member in the E_t^l.

On the other hand, if the individual $\mathbf{x}$ is dominated by any elite member in E_t^l, then it is not accepted in the E_t^l and its l-fidelity fitness is calculated as follows:

$$F^l(\mathbf{x}) = \begin{cases} p_{k*} - d_{k*} & \text{for } F^l(\mathbf{x}) \geq 0 \\ 0 & \text{for } F^l(\mathbf{x}) < 0 \end{cases}$$

Note that in this equation, a minimum fitness value of zero is assigned for avoiding non-negative fitness values.

4.2 Regeneration and High-Fidelity Function Evaluation. At the initial generation, only low-fidelity objective function values are calculated for every individual and the first member of the initial population P_0 is directly copied to the empty $E_0{}^L$ set while a random latent potential value p_k is assigned to it. Now, there is an elite member with a latent potential value in the EMS-L, and this will help in setting the low-fidelity fitness values of the remaining members in the initial population P_0 and the new members of the next generations.

Once the low-fidelity fitness values of all individuals in the population are evaluated while the EMS-L is constantly updated at the initial generation, *multifidelity tournament selection* is carried out. None of the individuals in the first generation has high-fidelity fitness values assigned, and the $E_t{}^H$ set is left empty at the first generation.

At the end of the fitness assignment, the multifidelity selection method, full and level crossover operators, and mutation, which are discussed later, are implemented.

Before starting the fitness assignment calculations of the new generation, the corresponding latent potential values of all elite members in the EMS-L and EMS-H are replaced by that of maximum among the elite members. That is,

$$p_k = \max(p_k(\mathbf{e})) \quad \text{for all } k = 1,2, \ldots ,K$$

Consequently, all elite members in each EMS are assigned an equal latent potential value regardless of their previous values or position in the objective space. This ensures that no distinction is made between elite members in an EMS, since they are all nondominated. Note that the latent potential values of EMS-L and EMS-H are not the same.

The costly objective function evaluation is the main concern for regeneration in the MOGAXL. In the MOGA, high-fidelity objective function evaluation is carried out to assign fitness at each generation. However, since the main aim and also the advantage of MOGAXL is reducing the number of costly high-fidelity objective function evaluations, a decision must be made for which level of fitness will be assigned and when at each new generation.

As indicated at the initial generation, only the lowest level fitness values are assigned for each individual. Then, the selection is performed by comparing the lowest-fidelity level fitness values. For a given probability of occurrence, full crossover and mutation operators are implemented on selected individuals. In the next and subsequent generations, the assignment of a higher level fitness value for an individual is decided according to previous selection. If in the previous selection, the selected individuals' low-fidelity fitness values are close each other enough according to confidence level, then in the next generation, the same level and one level higher fitness values for both individuals are assigned. If in the previous selection, the selected individuals' fitness values are far away from each other according to confidence level, then only that level fitness value of the individual is assigned in the next generation. For the specific case of a diploid MOGAXL, the decision of high-fidelity fitness assignment is given in equation form as

If $\dfrac{|F^L(P_1)-F^L(P_2)|}{\min(F^L(P_1),F^L(P_2))} > CL$ then *evaluate* $F^L(C_1)$

Else *evaluate* $F^L(C_1)$ *and* $F^H(C_1)$

Endif

Note that if the individuals have already assigned the highest level fitness values, then repeated evaluation is unnecessary.

The EMS update works only for fitness assigned individuals. If a high-level fitness is not yet assigned, this individual is not included in the dominance check of EMS-H. On the other hand, if the individual is going to be assigned a high-level fitness, then dominance check procedure is applied for the EMS-H update. Note that if the EMS-H is empty, the first individual to be assigned a high-level fitness will be directly added to the EMS-H.

4.3 Multifidelity Tournament Selection. One of the biggest disadvantages of the MOGA is that although one can use a low-fidelity fitness model such as ANN or a coarse grid, the MOGA is not able to converge to the global optimum in dynamic problems. That is because of the assignment of low-fidelity fitness values to the chromosome of an individual at the initial generations, the genes (binary digits) coming from ancestors penetrate the children's chromosomes with low-fidelity fitness values. For a MOGA, the only operator to get rid of these low-fidelity well performing but high-fidelity bad performing genes is mutation, which has a very low probability of happening to an individual. Therefore, the MOGA will eventually converge to a local optimum rather than the global.

On the contrary to optimizing the problem in a dynamic environment using a MOGA, fitness values are separately assigned to each chromosome of an individual for each surrogate model calculation in a MOGAXL. Each fitness value is assigned using either the exact or approximate solution of the objective function.

The selection method, which is developed for multiploid MOGAXL, includes an additional input parameter called the *confidence level* (CL). Expressed in percentage, the confidence level indicates the expected accuracy of the surrogate model used. A confidence level of 100% is assigned for the highest-fidelity model solution of an objective function. Therefore, the bottom level fitness values have 100% confidence level automatically.

On the other hand, lower percentages of confidence levels are assigned for lower-fidelity fitness levels. There might not exist a unique error percentage between the high- and low-fidelity accuracies of the objective function values. Nevertheless, confidence level is set by finding the mean deviation of the calculated objective function values between surrogate and exact solutions, and it can be expressed as

$$CL \approx 1 - \frac{|f_{\text{exact}} - f_{\text{surrogate}}|}{f_{\text{exact}}}$$

This equation is calculated for a few points in the design space and the average value can be set as the confidence level. Note that each objective function surrogate model is assigned a unique confidence level throughout the optimization. In the case of this study, since diploid genetic structure is used, and only one surrogate model is used, only one confidence level is set for the low-fidelity level fitness value.

Multifidelity tournament selection is used in the MOGAXL. The population is sorted randomly, and a *pair* of blades is selected as $\mathbf{x}_1$ and $\mathbf{x}_2$ from the sorted list. Starting from highest level (most accurate) fitness to lower levels, the existence of fitness is checked. If any of the two individuals do not have a fitness assigned on a given level, lower-fidelity level existing fitness values are checked. For a diploid MOGAXL, if the high-fidelity fitness values exist (calculated at the previous generation) for both $\mathbf{x}_1$ and $\mathbf{x}_2$, then the selection is performed based on high-fidelity fitness values, $F^H(\mathbf{x}_1)$ and $F^H(\mathbf{x}_2)$, respectively. Else, the selection is performed based on low-fidelity fitness values $F^L(\mathbf{x}_1)$ and $F^L(\mathbf{x}_2)$.

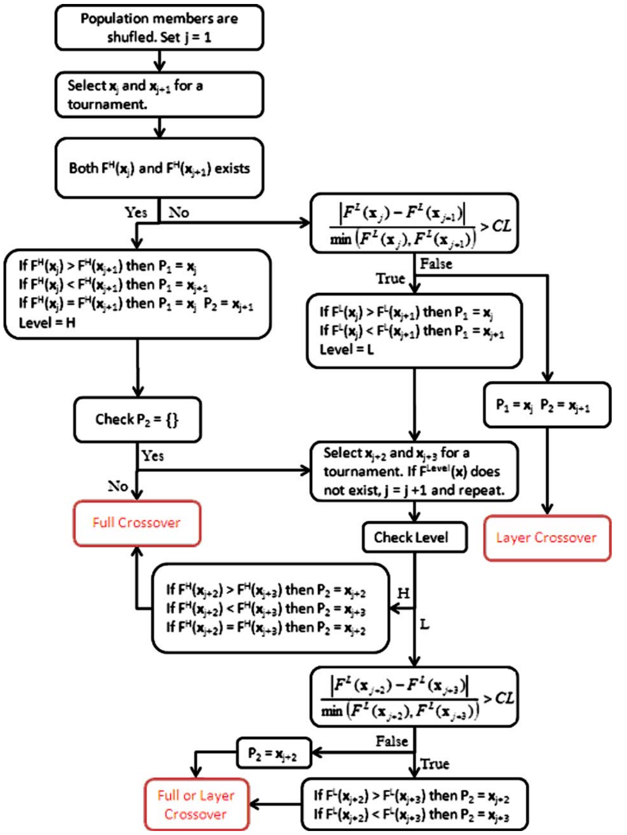

Population members are shufled. Set j = 1
Select x_j and x_{j+1} for a tournament.
Both $F^H(x_j)$ and $F^H(x_{j+1})$ exists
Yes
No
$\frac{|F^L(x_j)-F^L(x_{j+1})|}{\min(F^L(x_j),F^L(x_{j+1}))} > CL$
If $F^H(x_j) > F^H(x_{j+1})$ then $P_1 = x_j$
If $F^H(x_j) < F^H(x_{j+1})$ then $P_1 = x_{j+1}$
If $F^H(x_j) = F^H(x_{j+1})$ then $P_1 = x_j$ $P_2 = x_{j+1}$
Level = H
False
True
If $F^L(x_j) > F^L(x_{j+1})$ then $P_1 = x_j$
If $F^L(x_j) < F^L(x_{j+1})$ then $P_1 = x_{j+1}$
Level = L
$P_1 = x_j$ $P_2 = x_{j+1}$
Check P_2 = {}
Yes
No
Select x_{j+2} and x_{j+3} for a tournament. If $F^{Level}(x)$ does not exist, j = j +1 and repeat.
Full Crossover
Check Level
Layer Crossover
If $F^H(x_{j+2}) > F^H(x_{j+3})$ then $P_2 = x_{j+2}$
If $F^H(x_{j+2}) < F^H(x_{j+3})$ then $P_2 = x_{j+3}$
If $F^H(x_{j+2}) = F^H(x_{j+3})$ then $P_2 = x_{j+2}$
H
L
$\frac{|F^L(x_{j+2})-F^L(x_{j+3})|}{\min(F^L(x_{j+2}),F^L(x_{j+3}))} > CL$
False
$P_2 = x_{j+2}$
True
Full or Layer Crossover
If $F^L(x_{j+2}) > F^L(x_{j+3})$ then $P_2 = x_{j+2}$
If $F^L(x_{j+2}) < F^L(x_{j+3})$ then $P_2 = x_{j+3}$

Fig. 15 Multifidelity tournament selection

The detailed flowchart of the process is given in Fig. 15. At the end of the selection operator, two parents P_1 and P_2 will be ready to be mated by the crossover operator to obtain a child.

4.4 Full/Layer Crossover Operators. Once the parents to be mated are determined, crossover is applied to the parents according to a preselected probability. The crossover operator determines whether the information coming from the genes of the selected parents will be transferred to the next generation. Therefore, for a proper flow of information from past to future generations, and in order to prevent the mixing of low- and high-fidelity information, the crossover operator of a simple MOGA is carefully modified. Consequently, in a MOGAXL, two different kinds of crossover operators are used simultaneously in which both of them implements the single point crossover technique.

There are mainly two quality levels of information flowing from ancestors to the next generation in a MOGAXL when compared with a MOGA in which only fixed quality (accuracy) information is transferred. These are the better performing genes according to exact fitness values (highest-fidelity fitness levels), and the better performing genes according to surrogate fitness values (lower-fidelity fitness levels).

If the selected individuals' lower level fitness values are far away when compared with the confidence level, the information coming from the lower level genes can be used to guide higher level chromosomes. Therefore, "full crossover" is used in this case. On the other hand, if the selected individuals' lower level fitness values are within the confidence level, the information coming from the lower level genes cannot be used to guide higher level chromosomes, since one cannot decide if the better performing individual according to low-fidelity fitness is really performing better when exact fitness is compared. Therefore, "level crossover" is used in that case. For the specific case of a diploid GA, the decision of full versus level crossover is shown in Fig. 15 and given in equation form as

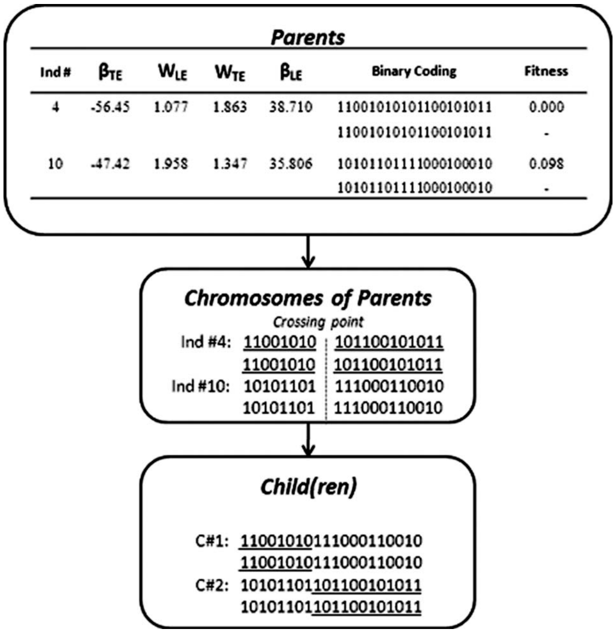


Fig. 16 Diploid MOGAXL full crossover illustration (for two children setting)

If $\dfrac{|F^L(P_1)-F^L(P_2)|}{\min(F^L(P_1),F^L(P_2))} > CL$ then full crossover

Else layer crossover
Endif

The full crossover operator requires chromosomes of all levels of selected individuals, which are crossed over at the same point simultaneously. This ensures that the better gene information is passed on all levels. The illustration of a full crossover is given in Fig. 16. The crossing point is chosen randomly.

The level crossover operator works only on one layer. This is the layer in which the selection of the individuals is performed according to this layer's fitness. Level crossover prevents any false information to be passed to higher level chromosomes. The crossover is performed on only the chromosome of the selection performed level. The illustration of a level crossover is given in Fig. 17 for the initial population.

Mutation is a bit change in a chromosome that occurs during the crossover process. In a MOGAXL, the mutation operator is only applied to the chromosomes of selection performed levels. Mutation probability is set as the probability of performing mutation on a bit of the chromosome in which the selection is performed on.

5 Optimization Results

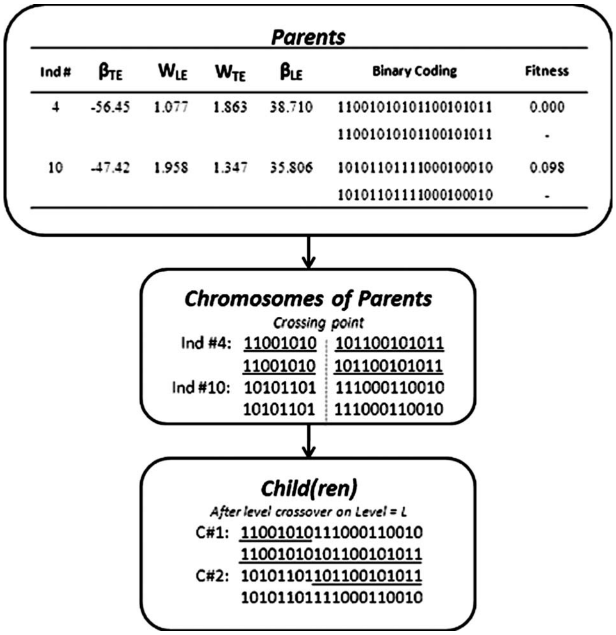
5.1 Optimization Problem Definition. Two objective functions are selected for the current optimization procedure: the blade adiabatic efficiency and the torque. For maximization of the both objective functions, two optimization cycles are run using the MOGA and the MOGAXL separately, and the design space is constrained by the same boundaries given in Table 4.

The total adiabatic efficiency η is calculated by the RANS analysis solution using the following equation:

$$\eta = \frac{1 - T_{0ABS,2}/T_{0ABS,1}}{1-(P_{0ABS,2}/P_{0ABS,1})^{\gamma-1/\gamma}}$$

Total temperature T and total pressure P are the mass flow weighted average quantities at the inlet (1) and exit (2) planes.

Similarly, torque T is calculated from



Parents

Ind#	β_{TE}	W_{LE}	W_{TE}	β_{LE}	Binary Coding	Fitness
4	-56.45	1.077	1.863	38.710	11001010101100101011	0.000
					11001010101100101011	-
10	-47.42	1.958	1.347	35.806	10101101111000100010	0.098
					10101101111000100010	-

Chromosomes of Parents
Crossing point
Ind #4: 11001010 101100101011
11001010 101100101011
Ind #10: 10101101 111000110010
10101101 111000110010
Child(ren)
After level crossover on Level = L
C#1: 11001010111000110010
11001010101100101011
C#2: 10101101101100101011
10101101111000110010

Fig. 17 Diploid MOGAXL level crossover illustration

$$T = N \sum p \cdot A$$

In this equation, pressure p is multiplied by tangential projection of the blade surface area of the cell faces A and summed up all of the cell faces on the blade surface. Then, the total torque is found from multiplying this result with the number of blades in a row.

5.2 Results. The baseline UH rotor blade is optimized using a haploid MOGA and a diploid MOGAXL for comparing the performance of a MOGAXL. Table 5 shows the MOGA and MOGAXL parameter values used in the optimization processes.

The high-fidelity Pareto-optimal frontiers after 100 generations of MOGA and MOGAXL are shown in Fig. 18. The figure represents the tradeoff between efficiency and torque and the decision maker may chose one blade shape out of all individuals in the Pareto-optimal frontier according to higher level information, such as stress calculations or rotor/stator interactions. In this figure, baseline rotor blade is also pointed for comparison. As long as maximizing torque and efficiency, the baseline blade is nondominated by all MOGA optimum solutions in the Pareto-front, but the baseline blade does not dominate any MOGA optimum solutions either. This indicates that after 100 generations, the MOGA did not converge sufficiently to the global optimum frontier yet.

On the other hand, the MOGAXL succeeded in finding as many as 11 nondominated optimum blades that also dominates the baseline blade. Although the Pareto-optimal frontier of the MOGAXL may not be the global frontier of the problem, the MOGAXL will eventually converge to it by increasing the number of generations of the optimization cycle.

Table 5 Optimization parameters of the MOGA and the MOGAXL

Parameter	MOGA	MOGAXL
Number of chromosomes	1	2
Design parameters	37	37
Chromosome length	219	219
Number of child	1	1
Maximum generation	100	100
Population size	100	100
Elitism	Yes	Yes
Crossover probability	90%	90%
Mutation probability	2%	2%
Initial latent potential p_k	2	2
Low-fidelity confidence level	-	50%

The diversity of the optimal solutions is a main concern in multi-objective optimization problems. In the present case, the nondominated individuals are broadly spread from a maximum torque value of 227 N m to a maximum efficiency value of 86.8%. Although it is desired to find many nondominated solutions in the objective space, because of the complexity of the problem, only 11 optimum solutions could be achieved. In case of increasing the number of generations of the optimization loop, the MOGAXL may converge to a better Pareto-frontier with more nondominated blade shapes.

From the beginning of the first generation until the end of the optimization, the number of nondominated individuals in the elite member set is given in Fig. 19. Only high-fidelity fitness assigned elite members of the MOGAXL are counted per generation. At the initial generations, the MOGAXL found twice as much elite members compared with the MOGA. Besides, the MOGA has 15 elite members at the end of the generation, and all of these members are dominated by MOGAXL elite members.

The second objective of this paper is the acceleration of the optimization cycle in order to decrease the total computational cost. This cost is evaluated by counting the number of high-fidelity objective function evaluations in the optimization process. Figure 20 shows the number of high-fidelity evaluations with respect to the generation number of the first ten generations.

The MOGAXL operates exactly the same as the MOGA after all of the individuals in the population are assigned their high-fidelity fitness values. Because, in this specific case, the MOGAXL assigned the high-fidelity fitness values to every individual of the tenth generation population, Fig. 20 is plotted until the tenth generation. After the tenth generation, both the MOGA and MOGAXL will perform 100 evaluations at each generation, which corresponds to the population size.

At the initial nine generations, the MOGAXL used very little computational resources when compared with the MOGA. In

Table 4 Design space constraints

	Minimum/maximum values					
Layer	ξ	Θ	β_{LE}	β_{TE}	W_{LE}	W_{TE}
1 hub	10/20	0.1/3	−60/−50	60/75	17/35	3/9
2	15/25	−2/−0.1	−55/−45	60/75	17/35	3/9
3	20/30	−3.5/−1.5	−45/−35	60/75	17/35	3/9
4	30/40	−5/−3.5	−35/−20	60/75	17/35	3/9
5	35/45	−7/−5	0/10	60/75	17/35	3/9
6 tip	50/60	−8/−6	10/20	60/75	17/35	3/9
N			26/33			

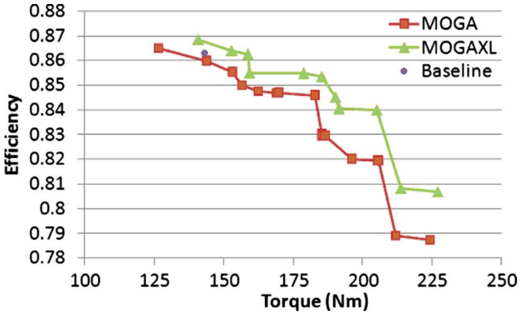


Fig. 18 Pareto-optimal frontiers of the MOGA and the MOGAXL after 100 generations

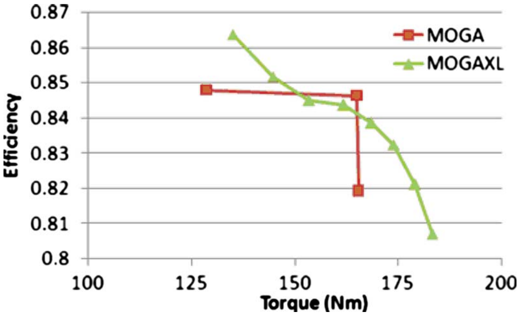


Fig. 21 Pareto-optimal frontiers after three generations of MOGA and nine generations of MOGAXL

these 9 generations 293 and 900 high-fidelity RANS solutions are requested by the MOGAXL and MOGA, respectively. Consequently, the ninth generation of the MOGAXL costs less than the third generation of MOGA in terms of computational expenses. Therefore, in order to compare the Pareto-optimal frontiers of both algorithms fairly, the ninth generation EMS-H of MOGAXL and the third generation EMS of MOGA are compared in Fig. 21.

According to Fig. 21, the MOGA could only find three elite members in the Pareto-optimal frontier, and only one of them is nondominated by the MOGAXL Pareto-optimal frontier. Whereas, the MOGAXL already has eight nondominated members, and only two of them are dominated by the MOGA. Additionally, the MOGAXL has an excellent distribution of elite members when compared with the diversity of the simple MOGA.

6 Conclusion

A multiploid distance based MOGA aerodynamic shape design optimization tool capable of handling surrogate models for a turbine blade has been successfully developed. The two objectives of the study were addressed by implementing a multiploid chromosome structure, two elite preserving sets, multifidelity fitness assignment, and modified crossover, selection, and mutation operators. The two-objective maximization problem demonstrates the success of the optimization tool regarding the accelerated global optimization. This study showed that the developed multiploid genetic algorithm optimization method offers a promising tool to multi-objective optimization problems involving surrogate models, while shortening design cycle and reducing design costs in the near future.

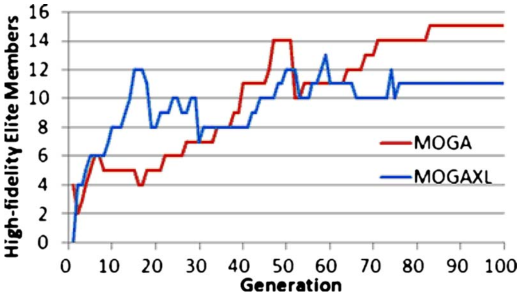


Fig. 19 Number of elite members in the EMS-H

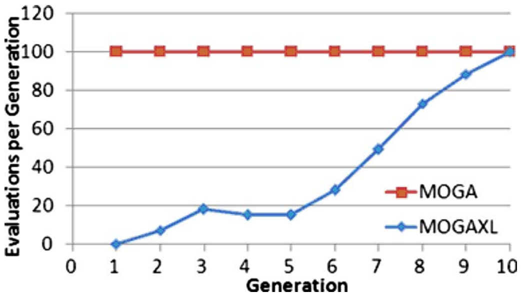


Fig. 20 Computational cost comparison

References

[1] Poloni, C., 1995, "Hybrid GA for Multi-Objective Aerodynamic Shape Optimization," *Genetic Algorithms in Engineering and Computational Science*, Wiley, New York, pp. 397–415.

[2] Nemec, M., Zingg, D. W., and Pulliam, T. H., 2002, "Multi-Point and Multi-Objective Aerodynamic Shape Optimization," Paper No. AIAA 2002-5548.

[3] Chiba, K., Obayashi, S., and Nakahashi, K., 2006, "Design Exploration of Aerodynamic Wing Shape for Reusable Launch Vehicle Flyback Booster," J. Aircr., **43**(3), pp. 832–836.

[4] Obayashi, S., and Sasaki, D., 2004, "Multiobjective Aerodynamic Design and Visualization of Supersonic Wings by Using Adaptive Range Multiobjective Genetic Algorithms," *Applications of Multi-Objective Evolutionary Algorithms*, World Scientific, Singapore, pp. 295–315.

[5] Lee, J., and Hajela, P., 1996, "Parallel Genetic Algorithm Implementation in Multidisciplinary Rotor Blade Design," J. Aircr., **33**(5), pp. 962–969.

[6] Ghisu, T., Parks, G. T., Jarrett, J. P., and Clarkson, P. J., 2006, "Multi-Objective Optimisation of Aero-Engine Compressors," *Proceedings of the Third International Conference on the Future of Gas Turbine Technology*, Brussels.

[7] Tiwari, A., and Roy, R., 2002, "Application of Generalised Regression GA for Designing a Turbine Blade Cooling System," Genetic and Evolutionary Computation Conference, Workshop Program, New York, pp. 108–113.

[8] Atashkari, K., Nariman-Zadeh, N., Pilechi, A., Jamali, A., and Yao, X., 2005, "Thermodynamic Pareto Optimization of Turbojet Engines Using Multi-Objective Genetic Algorithms," Int. J. Therm. Sci., **44**(11), pp. 1061–1071.

[9] Oyama, A., Liou, M.-S., and Obayashi, S., 2004, "Transonic Axial-Flow Blade Optimization Using Evolutionary Algorithms and a Three-Dimensional Navier-Stokes Solver," J. Propul. Power, **20**(4), pp. 612–619.

[10] Oksuz, O., and Akmandor, I. S., 2005, "Turbine Blade Shape Aerodynamic Design Using Artificial Intelligence," ASME Paper No. GT2005-68094.

[11] Oksuz, O., and Akmandor, I. S., 2001, "Turbine Cascade Optimization Using an Euler Coupled Genetic Algorithm," Paper No. ISABE-2001-1234.

[12] Ratle, A., 2001, "Kriging as a Surrogate Fitness Landscape in Evolutionary Optimization," Artif. Intell. Eng. Des. Anal. Manuf., **15**(1), pp. 37–49.

[13] El-Beltagy, M. A., Nair, P. B., and Keane, A. J., 1999, "Metamodelling Techniques for Evolutionary Optimization of Computationally Expensive Problems: Promise and Limitations," *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 196–203.

[14] Jin, Y., Olhofer, M., and Sendhoff, B., 2002, "A Framework for Evolutionary Optimization With Approximate Fitness Functions," IEEE Trans. Evol. Comput., **6**(5), pp. 481–494.

[15] Song, W. B., 1998, "Shape Optimisation of Turbine Blades," Ph.D. thesis, University of Southampton, Southampton, England.

[16] Pierret, S., 1999, "Designing Turbomachinery Blades by Means of the Function Approximation Concept Based on Artificial Neural Network, Genetic Algorithm, and the Navier-Stokes Equations," Ph.D. thesis, VKI, Belgium.

[17] Yang, S., 2006, "On the Design of Diploid Genetic Algorithms for Problem Optimization in Dynamic Environments," IEEE Congress on Evolutionary Computation, Vancouver.

[18] Branke, J., 2002, *Evolutionary Optimization in Dynamic Environments*, Kluwer, Dordrecht, The Netherlands.

[19] Grefenstette, J. J., 1992, "Genetic Algorithms for Changing Environments," *Parallel Problem Solving From Nature 2*, Elsevier, Brussels, Belgium, pp. 137–144.

[20] Yang, S., 2005, "Memory-Based Immigrants for Genetic Algorithms in Dynamic Environments," *Proceedings of the Genetic and Evolutionary Computation Conference*, Vol. 2, pp. 1115–1122.

[21] Cobb, H. G., and Grefenstette, J. J., 1993, "Genetic Algorithms for Tracking

Changing Environments," *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 523–530.

[22] Branke, J., 1999, "Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems," *Proceedings of the Congress on Evolutionary Computation*, Vol. 3, pp. 1875–1882.

[23] Louis, S. J., and Xu, Z., 1996, "Genetic Algorithms for Open Shop Scheduling and Re-Scheduling," *Proceedings of the 11th ISCA International Conference on Computers and Their Applications*, pp. 99–102.

[24] Mori, N., Kita, H., and Nishikawa, Y., 1997, "Adaptation to Changing Environments by Means of the Memory Based Thermodynamical Genetic Algorithm," *Proceedings of the Seventh International Conference on Genetic Algorithms*, pp. 299–306.

[25] Simoes, A., and Costa, E., 2003, "An Immune System-Based Genetic Algorithm to Deal With Dynamic Environments: Diversity and Memory," *Proceedings of the Sixth International Conference on Neural Networks and Genetic Algorithms*, pp. 168–174.

[26] Trojanowski, K., and Michalewicz, Z., 1999, "Searching for Optima in Nonstationary Environments," *Proceedings of the Congress on Evolutionary Computation*, pp. 1843–1850.

[27] Branke, J., Kaussler, T., Schmidth, C., and Schmeck, H., 2000, "A Multipopulation Approach to Dynamic Optimization Problems," *Proceedings of the Adaptive Computing in Design and Manufacturing*, pp. 299–308.

[28] Goldberg, D. E., and Smith, R. E., 1987, "Nonstationary Function Optimization Using Genetic Algorithms With Dominance and Diploidy," *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 59–68.

[29] Ng, K. P., and Wong, K. C., 1995, "A New Diploid Scheme and Dominance Change Mechanism for Non-Stationary Function Optimization," *Proceedings of the Sixth International Conference on Genetic Algorithms*.

[30] Ryan, C., 1994, "The Degree of Loneness," *Proceedings of the 1994 ECAI Workshop on Genetic Algorithms*.

[31] Abbott, I. H., Von Doenhoff, A. E., and Stivers, L. S., 1945, "Summary of Airfoil Data," NACA Report No. 824.

[32] Oksuz, O., and Akmandor, I. S., 2001, "Aerodynamic Optimization of Turbomachinery Cascades Using Euler/Boundary Layer Coupled Genetic Algorithms," Paper No. AIAA-2001-2577.

[33] Pritchard L. J., 1985, "An Eleven Parameter Axial Turbine Airfoil Geometry Model," ASME Paper No. 85-GT-219.

[34] Trigg, M. A., Tubby, G. R., and Sheard, A. G., 1999, "Automatic Genetic Optimization Approach to Two-Dimensional Blade Profile Design for Steam Turbines," ASME J. Turbomach., **121**(1), pp. 11–17.

[35] Anders, J. M., and Haarmeyer, J., 1999, "A Parametric Blade Design System," *Turbomachinery Blade Design Systems*, von Karman Institute for Fluid Dynamics, Brussels, Belgium.

[36] Yamamoto, K., and Inoue, O., 1995, "Applications of Genetic Algorithm to Aerodynamic Shape Optimization," Paper No. AIAA-95-1650.

[37] Farin, G., 1993, *Curves and Surfaces for Computer Aided Geometric Design*, Academic, New York.

[38] Faux, I. D., and Pratt, M. J., 1979, *Computational Geometry for Design and Manufacture*, Ellis Horwood, West Sussex, England.

[39] ANSYS® CFX®, http://www.ansys.com

[40] Rodi, W., 1975, "A Note on the Empirical Constant in the Kolmogorov-Prandtl Eddy-Viscosity Expression," ASME J. Fluids Eng., **97**, pp. 386–389.

[41] ANSYS® TURBOGRID™, http://www.ansys.com/

[42] Lefebvre, S., Hornus, S., and Nyret, F., 2005, "Octree Textures on the GPU," *Programming Techniques for High-Performance Graphics and General-Purpose Computation*, Addison-Wesley Professional, pp. 595–613.

[43] Boletis, E., Sieverding, C. H., and van Hove, W., 1990, "Test Case E/TU-1 Low Speed Annular Turbine Blade Row," AGARD Report No. AR-275.

[44] Evers, B., and Kötzing, P., 1990, "Test Case E/TU-4 4-Stage Low Speed Turbine," AGARD Report No. AR-275.

[45] Osyczka, A., and Kundu, S., 1995, "A New Method to Solve Generalized Multicriteria Optimization Problems Using the Simple Genetic Algorithm," Struct. Optim., **10**(2), pp. 94–99.

[46] Osyczka, A., 2002, *Evolutionary Algorithms for Single and Multicriteria Design Optimization*, Physica-Verlag, New York, pp. 93–113.