

Optimisation of Transonic Gas Turbine Blades with Evolution Strategies

Markus OLHOFFER*
Toyotaka SONODA**

Toshiyuki ARIMA**
Bernhard SENDHOFF*

Yaochu JIN*

ABSTRACT

要 旨

Evolution strategies belong to the class of evolutionary algorithms which are direct pseudo-stochastic search methods which mimic the principles of Neo-Darwinian evolution. They are known to be powerful optimisation techniques in particular for the solution of multi-modal, noisy and non-differentiable complex problems. For problems where a continuous representation is “natural”, like the geometry of gas turbine blades, evolution strategies are particularly suitable. Besides the choice of the search operators, the representation is one of the most crucial elements in optimisation. The advantages and disadvantages of different approaches are discussed in this work. Furthermore, to evaluate different geometric alternatives the fluid-dynamical properties of the blades have to be estimated. We propose a framework in which a Navier-Stokes solver and a neural network approximation model are combined. In this way, the computationally expensive evaluation of the fluid dynamics can be reduced while guaranteeing correct convergence of the optimisation algorithm. Finally, the optimisation results are discussed and the performance of the optimised and the original blades are compared both for the simulation method and for the transonic windtunnel experiment. In both results it can be seen, that a high performance increase is achieved.

進化的戦略(ES)はネオダーウィニズム的な進化論の原理を模倣した、直接的に擬似確率探索を行う進化的アルゴリズムの一つである。ESは多峰性でノイズのある微分不可能な複雑問題の解に対して特に強力な最適化技術であることが知られている。最適化問題において、表現形の記述手法は探索アルゴリズムと共に極めて重要な要素であるが、ESはガスタービン翼形状のように連続的な表現形の記述が必要な問題に特に有効である。本論では幾つかの異なった表現形記述手法についてそれらの得失を議論する。次に、異なったブレード形状の空気力学的特性を効率的に評価するための手法として、ナヴィエ・ストークスソルバーとニューラルネットワークを結合した手法を提案する。本手法は探索アルゴリズムの最適解への収束を保証しながら、計算コストの高い計算流体力学(CFD)による空気力学的特性の評価回数を短縮することができる。最後に、CFDシミュレーションと遷音速風洞実験の結果により、最適化された翼の特性について議論し、従来設計法による翼の性能を比較する。その結果、ESによる最適化翼の高い性能が実証された。

1. Introduction

The application of optimisation methods to complex systems in control, manufacturing and design has significantly increased in recent years. This increase is due to the development of new powerful optimisation methods which can handle nonlinear, multi-modal, non differentiable, dynamic, noisy and mixed combinatorial/continuous optimisation problems. Furthermore, the sharp increase of the available computing power makes

extensive simulations like in computational fluid-dynamics feasible even in an optimisation framework.

Evolutionary algorithms and the evolution strategy, which is employed in this work for the problem of design optimisation, are among the most powerful and most versatile of these new optimisation methods.

Examples for the application of evolutionary algorithms can be found in very different areas ranging from chip design⁽¹⁾⁻⁽³⁾ and the optimisation of mechanical structures^{(4), (5)} to the generation of neural structures^{(6), (7)} and even to

* Honda R&D Europe (Germany) GmbH

**和光基礎技術研究センター

the optimisation of the aesthetic of a design⁽⁸⁾. Applications for the optimisation of aerodynamic properties can be found in different areas of the aircraft design. Whereas Obayashi et al.⁽⁹⁾ and Sobieszanski-Sobieski et al.⁽¹⁰⁾ focus on the optimisation of the plane body, Vicini et al.⁽¹¹⁾, Doorly⁽¹²⁾ and Jones et al.⁽¹³⁾ target the optimisation of the inner air flow in gas turbines.

Although a solid theoretical basis of this type of optimisation algorithms is available, still constructive methods on how the algorithm has to be adapted to a particular problem and how the overall system has to be designed are sparse. Particular attention has to be paid to the representation of the optimisation problem, e.g. the geometry of a gas turbine blade and the way in which changes are inflicted on the representation in order to obtain alternative solutions. Since the computational cost of the fluid-dynamics simulations is very high, measures have to be taken to reduce the number of evaluations of alternative solutions. We approach this problem by employing a specific strategy which exploits information from past optimisation steps to minimise the necessary number of calculations. Furthermore, another recently developed method from computational intelligence, the neural network is used as an approximation tool to *learn* the fluid-dynamical properties of certain geometries. Some geometries can then be evaluated without the use of the computational fluid dynamics (CFD) solver.

In this paper, we will outline the set-up of a successful optimisation project consisting of several modules which address the questions raised in the previous paragraph. The core modules are the optimisation method and the CFD solver. The problem of representation is discussed in the context of the design of gas turbine blades and a framework for the integration of neural networks as approximation tools into the overall optimisation approach is described. In the last sections, we discuss the results of the optimisation both with regard to the results of the CFD calculations, as well as, to physical measurements in a transonic windtunnel.

2. Evolutionary Algorithms

2.1. Principles

Evolutionary algorithms are direct pseudo-stochastic search methods which mimic the principles of Neo-Darwinian evolution, see e.g. by Fogel⁽¹⁴⁾. A population of possible solutions (e.g. a vector of continuous parameters, the objective variables, describing a blade geometry) is adapted to solve a given problem (e.g. minimisation of pressure loss) over several generations. The adaptation occurs by variation of solutions contained in a population and by selection of the best solutions for the next generation. The variations can be classified as purely stochastic (usually called mutation) and combinatoric/stochastic (usually called recombination or in the context of genetic algorithms crossover). Formally, an evolutionary algorithm can be described as follows, schematically the evolution cycle is shown in Fig. 1:

ALGORITHM

$t := 0;$

initialise $P_g(0) := \{g_1(0), \dots, g_\mu(0)\}$

decode and evaluate $P_g(0) : \Psi(c_{\theta_c}^{-1}(P_g(0)))$

do

recombine $P'_g(t) := r_{\theta_r}(P_g(t))$

mutate $P''_g(t) := m_{\theta_m}(P'_g(t))$

decode $P''_p(t) := c_{\theta_c}^{-1}(P''_g(t))$

evaluate $P''_p(t) := \Psi(P''_p(t))$

select $P_p(t+1) := s_{\theta_s}(P''_p(t) \cup Q)$

code $P_g(t+1) := c_{\theta_c}(P_p(t+1))$

$t := t + 1$

until stop

NOMENCLATURE

$P_g(t)$	genotype population, generation t
$P_p(t)$	phenotype population, generation t
Q	additional set of individuals, e.g. $P_p(t)$
$g_1(t)$	genotype of the first individual
Ψ	fitness function
μ, λ	population size (parents, offspring)
r_{θ_r}	recombination operator
m_{θ_m}	mutation operator
$c_{\theta_c}, s_{\theta_s}$	coding and selection operator

In the formal description, the coding and decoding processes have been explicitly included. They describe the transition from the genotype, which is also called the representation or the blueprint, to the phenotype, which is evaluated. The representation is one of the most important factors in the design of evolutionary algorithms. Together with the search operators most of the current algorithms can be classified with respect to the representation they use, e.g. genetic algorithms, evolution strategies, genetic programming and evolutionary programming.

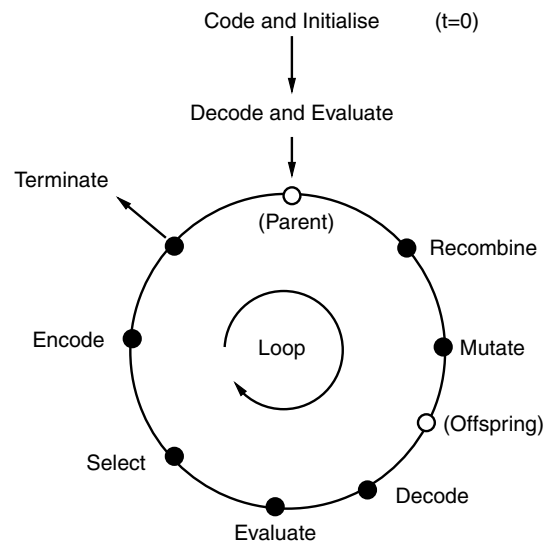


Fig. 1 The evolution cycle.

2.2. Evolution strategy

In the evolution strategy (ES) the essential variations during the evolutionary search are mutations which are realized by adding normal distributed random numbers to the objective variables. In Fig. 2 different normal distributions are shown; they determine the distribution of the offspring around the parent. The variances σ^2 of the normal distribution are called the strategy parameters of the search process and their values determine the width of the search. The variances have to adapt during the process to the local topology of the search space. This process of *self-adaptation* is one of the key principles of ES. It relies on a “second-order” or indirect selection of the strategy parameters which are part of each individual. In Fig. 3 a bird’s eye view of different normal distributions $N(\mathbf{a}_1, (\sigma_1^1, \sigma_1^1))$, $N(\mathbf{a}_2, (\sigma_2^1, \sigma_2^1))$ and $N(\mathbf{a}_3, (\sigma_3^1, \sigma_3^2))$ is shown for a two dimensional representation space. The “probability axis” would be perpendicular to the paper and the circles (ellipse) correspond to lines of equal probability (due to the symmetry of the normal distribution). These lines are also called iso-density lines. Thus, if the parent vector is given by $\mathbf{a}_1$, ca. 46% of all offspring will lie within the shown circle with radius σ_1^1 (91% for radius $2\sigma_1^1$).

In the standard evolution strategy only a diagonal covariance matrix is used, i.e. the mutation of the objective x is carried out by adding $N(0, \sigma_i^2)$ distributed random numbers z_i to each component x_i . The “step-sizes” σ_i are also subject to mutations (log-normal distributed) both for each component separately (parameterised by τ) and overall (parameterised by τ'). Thus, the individual consists of both the objective and the step-size vector. Formally, the standard evolution strategy with self-adaptation can be expressed as follows: (see e.g. Schwefel⁽¹⁵⁾)

$$\sigma_i(t) = \sigma_i(t-1) \exp(\tau' z) \exp(\tau z_i) \quad (1)$$

$$\mathbf{x}(t) = \mathbf{x}(t-1) + \tilde{\mathbf{z}} \quad (2)$$

$$z_i, \mathbf{z} \sim N(\mathbf{0}, 1), \tilde{\mathbf{z}} \sim N(\mathbf{0}, \tilde{\sigma}(t)^2).$$

The selection in ES is typically deterministic, i.e. from μ parent individuals, λ offsprings are produced and from these the best μ individuals are selected to form the parent population of the next generation. This method is usually abbreviated to (μ, λ) -selection.

Evolution strategies are often used with some kind of recombination operator (discrete or intermediate), which is basically the same as the crossover operator in genetic algorithms, however, it usually plays a minor role for the search process and is completely omitted in this work.

Covariance matrix adaptation The derandomized Covariance Matrix Adaptation (CMA)⁽¹⁶⁾ differs from the standard ES mainly in three respects:

- Instead of the variances (σ_i^2) , the full covariance matrix of the probability density function

$$f(\mathbf{z}) = \frac{\sqrt{\det(\mathbf{C}^{-1})}}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}(\mathbf{z}^T \mathbf{C}^{-1} \mathbf{z})\right). \quad (3)$$

is adapted for the mutation of the objective parameter vector. The (σ_i^2) can only describe probability distributions with the main axes parallel to the coordinate axes. Using the full covariance matrix $\mathbf{C}^{-1}$, rotated distributions, like the blue ellipse in Fig. 3, can be represented. Since $\mathbf{C}^{-1}$ has to be positive definite with $\det(\mathbf{C}^{-1}) > 0$, the different matrix entries cannot be determined independently and the detailed adaptation algorithm is a little more involved.

- In order to reduce the stochastic influence on the self-adaptation, one stochastic source for both the adaptation of the objective and of the strategy parameters is used. Note, that in eqs. (1) and (2), the stochastic elements (z, z_i) and $\tilde{\mathbf{z}}$ are independent. In the derandomized approach, the *actual* step

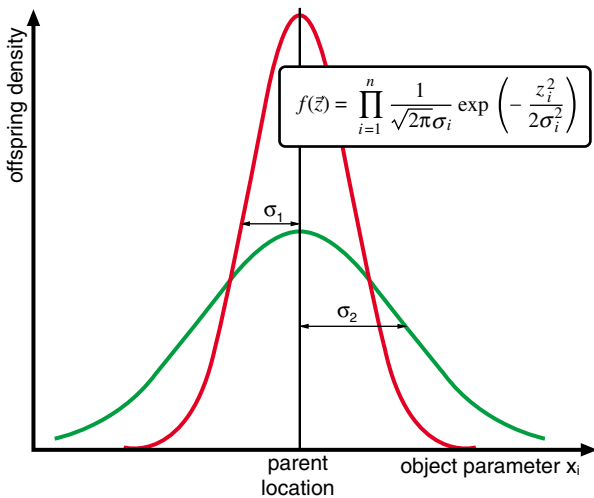


Fig. 2 The normal distribution $N(0, \sigma^2)$ for different variances σ^2 . For the n -dimensional case, $f(\mathbf{z}) \sim N(\mathbf{0}, \sigma^2)$ is given.

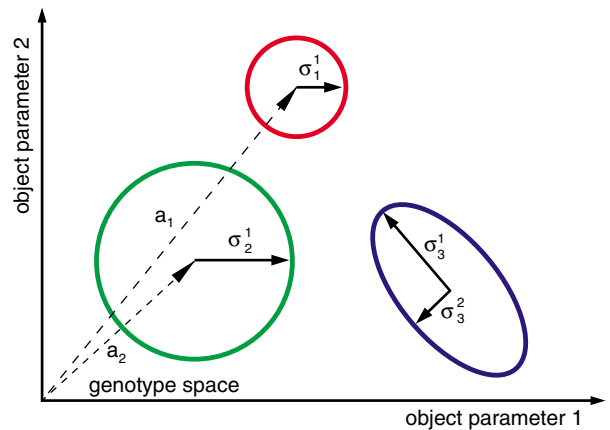


Fig. 3 Iso-density lines for three different two dimensional normal distributions.

length in the objective parameter space is used to adapt the strategy parameter.

- The second method is the introduction of the cumulative step size adaptation. Whereas the standard evolution strategy extracts the necessary information for the adaptation of the strategy parameters from the population, the cumulative step size adaptation additionally relies on information collected during successive generations. This leads to a reduction of the necessary population size.

3. Overview of the gas turbine engine

3.1. A single gas turbine engine

The simplest gas turbine engine for an aircraft is a turbojet. As shown in Fig. 4, the turbojet engine is essentially designed for the purpose of producing high-velocity gas at the jet nozzle. An axial flow compressor is located in the forward section of the engine. The compressor is followed by a combustion chamber. Behind which is an axial flow turbine that produces the power needed to drive the compressor. Lastly, there is the jet nozzle. The compressor raises the pressure of the incoming air before it enters the combustion chamber. Fuel is sprayed through nozzles into the front of the chamber. The resulting mixture of fuel and air is burned and the expanding gas rushes into the turbine section, causing the turbine rotors to rotate. On leaving the turbine section, the gas is expelled to the outside air through the jet nozzle. The power that the turbine rotors extract from the gas is used to drive the compressor. The turbine must produce sufficient power to turn the compressor and yet leave enough energy in the exhaust gas to generate a worthwhile amount of forward thrust. Superior compressor and turbine combinations eventually led to successful engines.

3.2. Axial flow compressor blades

The axial flow compressor consists of a multi-stage unit comprising of a row of rotor blades followed by a row of stator blades. As shown in Fig. 5 (a), the absolute velocity is initially accelerated by the rotor blades, and then decelerated in the stator blade passages wherein the kinetic energy transferred in the rotor is

converted to the static pressure. The process is repeated until the required pressure rise is achieved. The rotor and stator blades consist of airfoil sections with the passage between adjacent blades forming a divergent duct. Therefore, the flow velocity relative to the rotor and stator is decreased. The stage is usually designed to give a pressure gradient along its length to ensure that the fluid maintains a reasonably uniform axial velocity as the density increases toward the downstream. That is why there is a gradual reduction of the air annulus area in the compressor meridional passage, see Fig. 4.

3.3. Axial flow turbine blades

As mentioned earlier, the turbine has the function of providing the power to drive the compressor. To produce the driving power, the turbine consists of fewer stages, each stage comprising of a row of stator blades followed by a row of rotor blades, as shown in Fig. 5 (b). However, the passage between adjacent blades forms a convergent duct. The flow is relatively stable due to a negative pressure gradient, as compared with the positive pressure gradient for the compressor blades. This is why a single stage turbine can drive a large number of compressor stages. When the gas expands in the combustion process, it forces its way onto the turbine stator blades where it is accelerated to about the speed of sound due to the convergent shape of the blade. At the same time the gas is swirled by the stator blades in the direction of the rotation of the rotor blades. Due to the impact of the gas on the rotor blades and due to the subsequent flow of the gas through the rotor blades, energy is absorbed which causes the rotor blades to rotate at high speed and in turn provides the power for driving the compressor.

3.4. Seeking a new design principle

In order to reduce the engine weight and the manufacturing cost, it is very attractive to reduce considerably the number of stages needed in the compressor and/or turbine part of the engine while maintaining the same level of efficiency. However, in order to realize such a reduction, the aerodynamic engineer has to uncover new design principles for very highly loaded gas turbine blades. Many aerodynamic

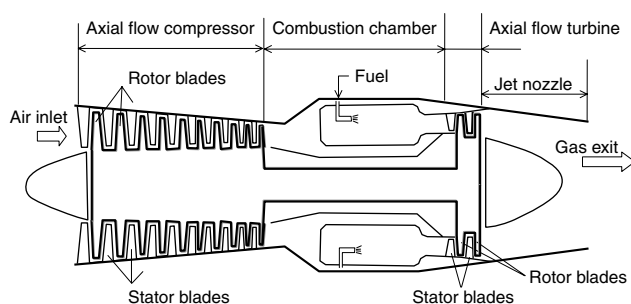


Fig. 4 Schematic view of a typical single gas turbine engine.

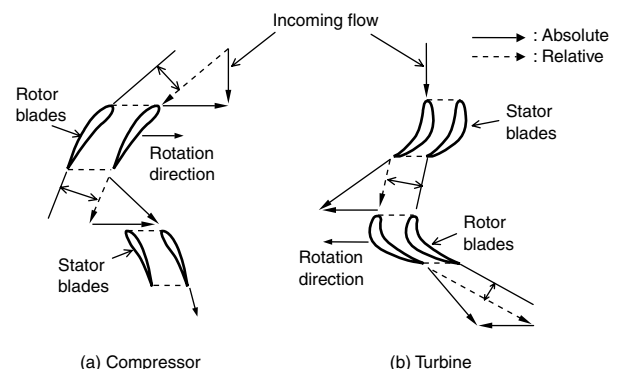


Fig. 5 Comparison of typical forms of (a) the compressor and (b) the turbine blades.

properties have to be taken into account for the design of very high loaded blades. The designer has to pay attention to the steady pressure loss due to *shock waves themselves*, due to *an interaction of the shock waves with boundary layers*, due to *the secondary flow* and due to *the tip clearance flow*. Furthermore, the designer has to consider the effect of the Reynolds number on the compressor and on the turbine. The Reynolds effect will be a critical problem especially for a low-pressure turbine in a twin-spool small turbofan engine. Other potential sources of pressure losses are *unsteady phenomena*, like an interaction of stator blades with rotor blades.

By concentrating on conventional designs, the “expert’s way of thinking” can be too restricted. In these cases, stochastic optimisation methods like evolution strategies can uncover new principles due to their “inherently unbiased” approach.

4. Optimisation of gas turbine blades

In order to apply evolution strategies to the optimisation of blade geometries, we have to define a suitable representation. This set of objective variables describes a blueprint for the construction of a blade and is stored in the individual, see Fig. 10. Of course different blueprints or blade models are possible, all with specific drawbacks and merits. Examples of models which have been used previously for the optimisation of gas turbine blades are described in Section 4.1.

In order to evaluate the quality of a geometry, computational fluid dynamic (CFD) calculations are used which are described in more detail in Section 4.2 and which allow an accurate prediction of the streaming conditions around the blade for realistic transonic constraints.

Due to the high computational demand of CFD calculations, additionally an approximation model has been used to speed up the estimation of the blade quality, see Section 4.3. Finally, in Section 4.4. the overall system architecture is described and in Section 4.5 some results of the optimisation are given.

4.1. Representation of the blade geometry

In Figs. 6, 7 & 8, three blade models are described. The main difference is the degree of freedom which they allow in order to modify the geometry. A representation, which allows a detailed description of the geometry, tends to need a large number of parameters. On the other hand, the number of parameters determines the dimension of the search space and significantly influences the number of iterations which the optimisation methods need to find an optimal or near optimal solution. In general, an optimal representation of a geometry is flexible enough to describe any geometrical property, using only a minimal number of parameters, i.e. it is compact and flexible. Compactness of a blade model can be achieved by using expert knowledge about the problem and about known solutions.

An example for such a kind of model defined for sub-sonic stationary blades is given in Fig. 6. Only

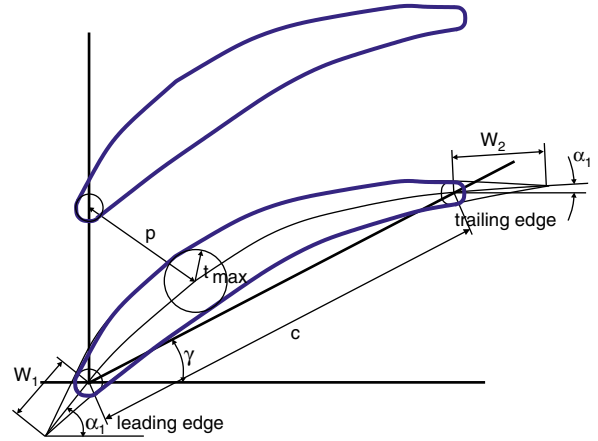


Fig. 6 Encoding of a compressor stator blade which is strongly based on a priori knowledge.

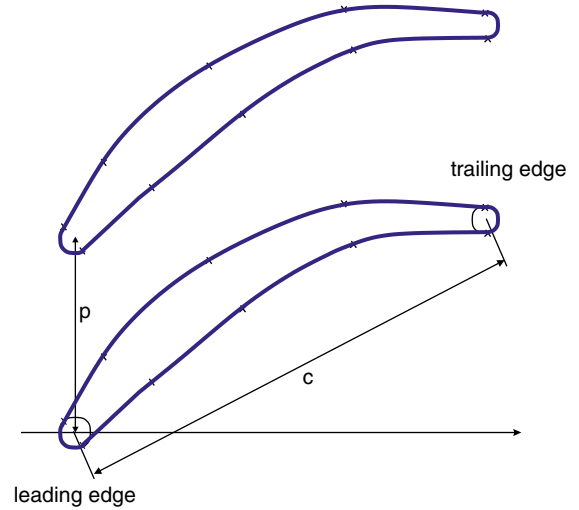


Fig. 7 Extended encoding for the suction and pressure side of the blade.

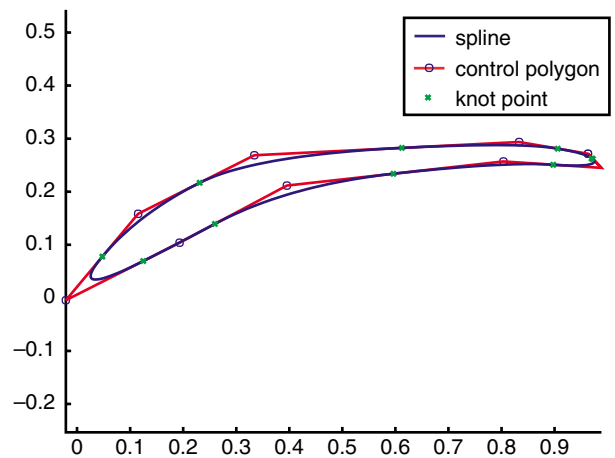


Fig. 8 Spline encoding of the compressor stator blade with increased degree of freedom at the cost of a higher dimensional search space.

parameters which are assumed to be very important for both the flow conditions and the stability of the design are included in the model. The curves describing the suction and pressure side of the blade are defined by 3rd order Bézier splines. The splines are fixed by angles and radii at the leading and trailing edge and by a defined radius in the middle section of the blade which defines for example the thicknesses and wedge angles. This model allows a highly compact representation of the design, however, the freedom for modifications of the geometry is restricted. In turn this restricts the possible outcome of the optimisation.

A more detailed description of the curves can be achieved by increasing the degree of freedom at the middle part of the blade. Such an extended model is shown in Fig. 7 where the curves at the suction and pressure side are defined by higher order polynomials which are tangential to the circles at the leading and trailing edge. In this case, a more detailed description of the blade is realised at the cost of a higher dimensional search space. This representation has been successfully employed for the optimisation of gas turbine blades with evolution strategies by Olhofer et al.⁽¹⁷⁾.

An even more flexible model describing the cross section of a blade is shown in Fig. 8. The description is based on a cyclic Non-uniform Rational B-spline (NURBS). Besides elementary constraints like closeness, continuity and differentiability only the number of control points is predetermined, which influences the smoothness of the spline and at the same time the degree of details which can be represented. The degree of freedom of this model can be increased or decreased by inserting or removing control points in the spline description. A priori knowledge can still be represented by defining a suitable initialisation of the blade geometry. This can be seen in Fig. 9, where the same model is used to describe a rotating turbine blade. In particular due to the changeable degree of freedom this model is used in the optimisation discussed in the remainder of this paper. In Fig. 10 the complete representation of the blade and the strategy parameters, which are contained in each individual, are shown.

4.2. Notes on the CFD calculations

In this section the flow solver used in this work and its most important features will be described. The performance of the solver was investigated in detail and demonstrated by Arima et al.⁽¹⁸⁾

The basic equations governing the viscous flow through blade rows are the three-dimensional, Reynolds-averaged, Navier-Stokes equations. To complete the system of governing equations, the Reynolds stress tensor, which results from the Reynolds-average for the Navier-Stokes equations, is evaluated using a low Reynolds number (κ - ϵ) equation proposed by Chien et al.⁽¹⁹⁾.

The governing equations are solved using an implicit time-marching finite-difference scheme to obtain a steady-state solution. The implicit approximate factored (IAF) scheme based on the diagonal alternating-

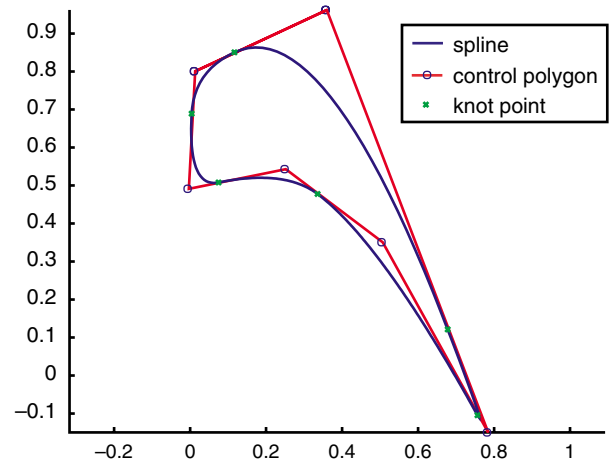


Fig. 9 Spline encoding of a turbine rotor blade using the same model as in figure 8.

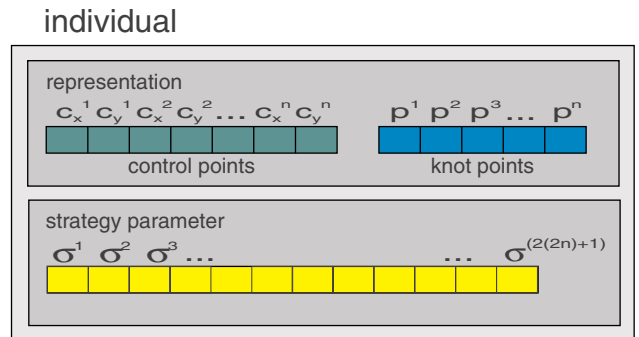


Fig. 10 Set-up of each individual, consisting of the representation of the blade, which are the control and knot vectors of the splines, and the strategy parameters of the CMA evolution strategy.

direction-implicit (ADI) scheme developed by Pulliam and Chaussee⁽²⁰⁾ has been applied for the time integration. The high-accuracy total-variation-diminishing (TVD) formulation proposed by Chakravarthy and Osher⁽²¹⁾ has been used for the inviscid terms. The viscous terms have been evaluated using standard second-order, central-difference formulae. The turbulence quantities are obtained by lagging, once the mean flow equations have been updated.

The boundary conditions are needed to calculate the flow field in blade rows. Along the blade surfaces, the no-slip condition is imposed on the velocities. The pressure is extrapolated from the adjacent grid points and the density is computed using the adiabatic wall condition. The turbulent kinetic energy, κ and the isotropic dissipation rate, ϵ are set to zero along the solid boundaries.

At the inflow boundaries, the total pressure and temperature profiles and two inlet-flow angles are specified; and the static pressure is extrapolated from the interior flow field domain. The turbulent kinetic energy and dissipation rate imposed at the inflow boundary are

derived from the specified free-stream turbulence intensity and the unit eddy viscosity at the inflow boundary. At the outflow boundaries, the static pressure is specified. The density and velocity components, κ and ε are extrapolated from those at interior points. At the periodic boundaries upstream of the leading edge and downstream of the trailing edge, the values at two corresponding points should be equal. The detailed numerical procedures have been previously reported by Arima et al.⁽¹⁸⁾.

4.3. Neural network models and evolution control

Since the computational cost of numerically solving the Navier-Stokes equations is very high, it is important in aerodynamic optimisation to reduce the number of CFD simulations. One way to achieve this is to build an approximate model with a priori knowledge from both design engineers and previous design optimisation. To this end, artificial neural network models, described for example in Bishop⁽²²⁾, can be used.

Artificial neural networks are parallel computational systems consisting of a large number of simple nonlinear processing elements (known as neurons) connected by weights. One of the most important features of neural networks is their ability to learn from human knowledge and experimental data like described e.g. by Jin et al.⁽²³⁾ by adjusting the structure of networks and the strength of the weights. After the learning (also called training) process has been completed, the neural network model can be used to predict an unseen value for a given input. In this work, neural networks are used to approximate the task of a Navier-Stokes solver. Although the applicability of the neural network will be restricted to geometries which are similar to the ones it has learned before, it can evaluate the different designs at a much faster speed than the CFD method. In other words, given the parametric description of the blade geometry, the neural network is able to predict the performance of the blade at a much lower computational cost than a Navier-Stokes solver. Of course, it is assumed that the neural network has been successfully trained.

Feedforward multi-layer perceptron (MLP) neural network models are employed in this work. As discussed in Section 4.1, Non-uniform Rational B-Splines (NURBS) have been used to represent the blade geometry. Therefore, the input-output relationship of the neural network model can be described as

$$\mathbf{P} = NN(\mathbf{CP}), \quad (4)$$

where $\mathbf{P}$ is a vector of performance, such as pressure loss and deviation angle, and $\mathbf{CP}$ is a vector of control points of the NURBS. The structure of the neural network model used in this work is illustrated in Fig. 11, where

$$\mathbf{CP} = (x_1, y_1, \dots, x_n, y_n) \quad (5)$$

are the x and y coordinates of the control points of the B-splines; $\mathbf{P} = (\omega, \alpha_2)$ are the pressure loss and the deviation angle.

It is unlikely that the neural network model will show

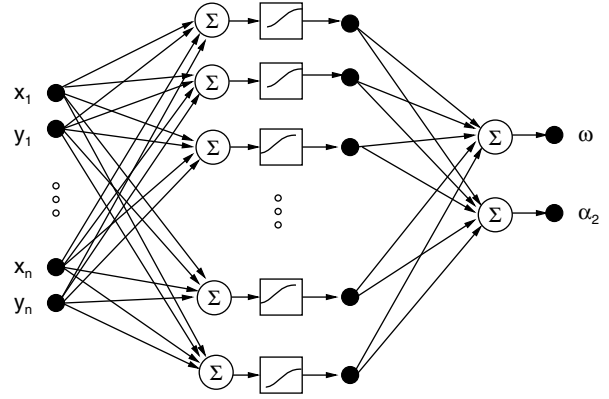


Fig. 11 The neural network model.

the same performance as the CFD solver. This is mainly due to the limited a priori knowledge, sparse data available for training, and the high dimensionality of the input space, i.e. large n in eq. (5). For example, in our work, 26 control points have been used to describe the blade geometry. For a two-dimensional description of a blade, the input space of the network is therefore 52 dimensional. On the other hand, the data is usually limited because of the high computational cost for their generation. As found by Jin et al.⁽²⁴⁾, the evolutionary algorithm may converge incorrectly when false minima are present in the neural network model. This problem can be addressed in two respects. Instead of a global model, a local model should be used which approximates only a part of the search space. In this way, it is much easier to get a high quality model with limited training data. Furthermore, the network model should not be used to replace the CFD method, but to supplement it. This approach has been called “evolution control”⁽²⁴⁾ and will be introduced in the following.

Individual-based and generation-based evolution control schemes have been suggested by Jin et al.⁽²⁴⁾. A key issue in evolution control is how often the approximate model should be used, which is usually called model management in optimisation. So far, this has not received much attention in evolutionary optimisation with approximate models. Either the approximate model is assumed to be correct and no evolution control is considered like in the work of Redmond et al.⁽²⁵⁾, or the approximate model is used at a fixed frequency like by Ratle et al.⁽²⁶⁾. To use the neural network model more effectively, a framework for model management in evolutionary optimisation has been used here, see Fig. 12 and by Jin et al.⁽²⁷⁾. The main idea behind the framework is that the better the fidelity of the neural network model is, the more frequently the model should be used. In other words, the evolution control frequency is adjusted adaptively during optimisation based on the model quality, which can be estimated using the past data both from the Navier-Stokes solver and from the neural network model.

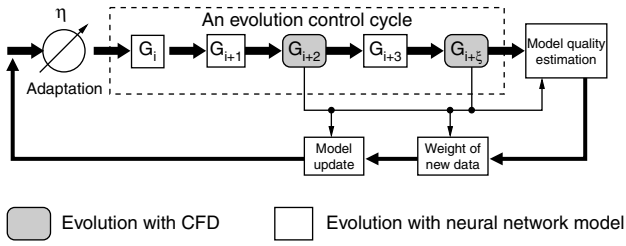


Fig. 12 A framework for evolution control. In an evolution cycle, there are ξ generations, among which η generations use the CFD solver, and $(\xi - \eta)$ generations use the neural network model for fitness evaluation.

From Fig. 12, it can be seen that in an evolution cycle, there are ξ generations in total, among which η generations use the CFD solver, and $(\xi - \eta)$ generations use the neural network model for fitness evaluation. Once the CFD solver is used, new data become available. The training data will be used both for the estimation of the model quality and during on-line learning (model update). Three steps are implemented after each evolution control cycle. Firstly, the model quality is estimated, which is achieved by comparing the results from the CFD solver with the ones from the neural network model. Secondly, the covariance matrix of the CMA evolution strategy is exploited to assign a weight to the new data samples for on-line learning. This is useful because the covariance matrix contains information on the search direction of the evolution strategy. The more likely it is that the evolution strategy will visit a particular region in search space in the next steps, the larger the weights will be that are assigned to the samples from this region. The larger the weight of a sample the more important this sample will be during on-line learning. Thirdly, the neural network model will be updated using the new samples collected within the last evolution control cycle. Suppose that in an evolution control cycle, N new samples have been collected, then the cost function for on-line neural network learning with weighting is described as follows:

$$E = \frac{1}{E} \sum_{i=1}^N w(i) (\omega(i) - \hat{\omega}(i))^2, \quad (6)$$

where $\omega(i)$ is the i -th new sample calculated from the CFD solver and $\hat{\omega}(i)$ is the one predicted by the neural network model for the same blade geometry; $w(i)$ is the weight for the i -th sample obtained in the following way:

$$w(i) = \exp\left(-\frac{1}{2} (\mathbf{P}_A(i))^T \mathbf{C}^{-1} \mathbf{P}_A(i)\right) \quad (7)$$

$$\mathbf{P}_A(i) = \mathbf{C}\mathbf{P}(i) - \mathbf{C}\mathbf{P}_R(i),$$

where $\mathbf{C}\mathbf{P}(i)$ is the input vector to the neural network (the geometry) for the i -th sample, $\mathbf{C}\mathbf{P}_R(i)$ is a reference vector, usually the best geometry in the last generation in which the CFD solver has been used for fitness

evaluation, and $\mathbf{C}^{-1}$ is the covariance matrix of the evolution strategy discussed in Section 2.

4.4. Overall system architecture

The overall system architecture is shown in Figure 13. All parts of the program including the optimisation algorithm, as well as, the quality evaluation are realised as modules which can easily be modified or exchanged according to the needs of the optimisation task.

The algorithm is initialised with a known geometry, which is the starting point for the optimisation. This assures that basic properties and constraints are fulfilled so that the flow can be calculated using the CFD method described in Section 3. In particular for the optimisation of the blade geometry in this work, a proper initialisation is important, since non-stationary flow fields cannot be evaluated.

After the initialisation of the algorithm the iteration process – over different generations – starts with the reproduction of the initial parent population.

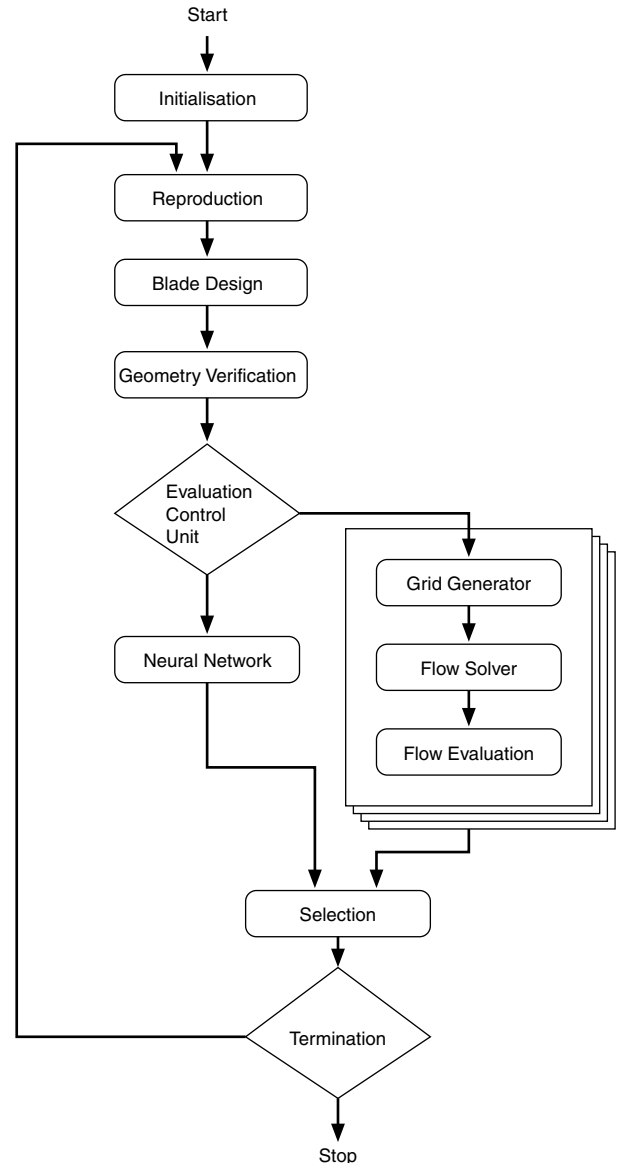


Fig. 13 The overall, modular system architecture.

The representations of the geometries – the blueprints, which are generated in the reproduction step, are mutated according to the used method, see Section 2. Next, in the “design module” the blade models are generated from the representations. In order to determine the quality, different criteria have to be taken into account. In addition to the flow conditions, constraints concerning the mechanical and thermal stability have to be evaluated in the “geometry verification” module.

Thereafter, two different modules are used to estimate the fluid-dynamics properties of the blade. On the one hand, the Navier-Stokes flow solver is used. The results are in very good agreement with physical measurements, however, at the drawback of a high calculation time. Therefore, this part of the system is parallelised: the quality of different geometries is evaluated in parallel on different machines of a computing cluster which was built for this purpose. In this work, the communication between different processes has been organised with the message passing library PVM described by Geist et al.⁽²⁸⁾. On the other hand, a neural network model is used to further increase the calculation speed. The frequency with which the neural network model is used is determined by the evolution control method, which was outlined in the previous section.

After the evaluation of all geometries within the actual population the best ones are selected and used as parents in the next reproduction step – the next generation. The algorithm is terminated after it has converged to an optimised geometry or after a given time limit has been reached.

4.5. Optimisation results

Firstly, we will present the results from optimisation runs using the CMA evolution strategy, see Section 2, without using the neural network module.

In a second part, the speed up due to the use of a neural network is demonstrated by an example of an optimisation of a transonic turbine blade.

In the last part, the comparison of the performance of an optimised turbine blade with the previously available blade geometry is shown. Additionally, results from a physical measurement in the transonic wind tunnel for plane cascades of the DLR (Deutsches Zentrum für Luft und Raumfahrt e.V / German Aerospace Centre) Göttingen are presented in the figure in order to allow a comparison of the simulation results and the physical measurements both of the optimised and the initial geometry.

Evolution strategy The presented results were obtained with a $(\mu, \lambda)=(1,12)$ evolution strategy. The population was initialised with an existing blade geometry. In Fig. 14 the history of the overall quality during the optimisation process is shown. The number of quality evaluations (calculations) rather than generations is shown on the x -axis of all plots. Figures 15 (a) & (b) show the development of the pressure loss

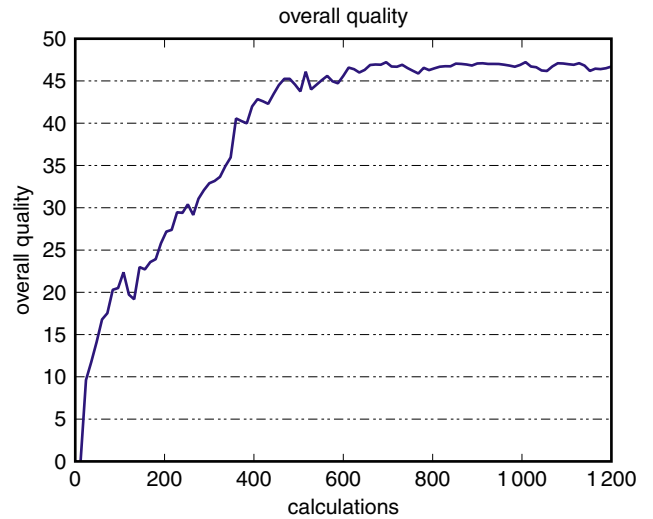


Fig. 14 Blade Quality.

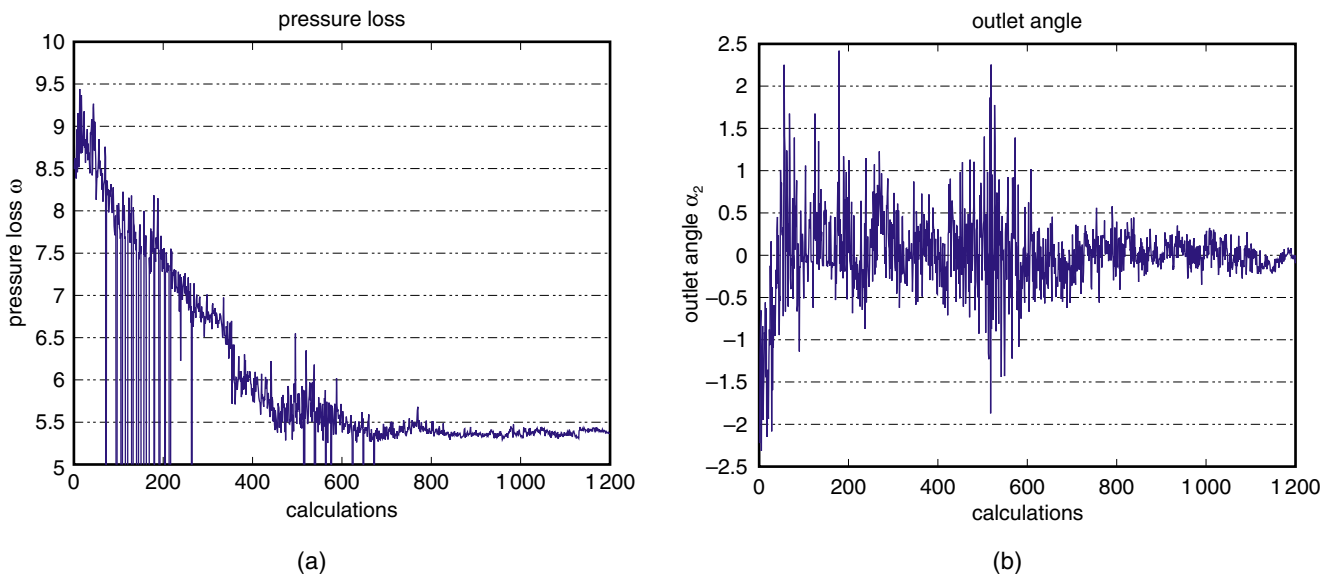


Fig. 15 (a) Total pressure loss and (b) outlet angle.

ω and the deviation angle α_2 during the optimisation, which are the main criteria for the optimisation. The outliers in Fig. 15 correspond to simulations which did not converge mainly due to unstable non-stationary flow conditions. These geometries were removed from the population.

In order to generate valid geometries, we introduced in the quality function a strong bias towards geometries which fulfil all constraints. Therefore, during the first generations mainly the deviation angle is adapted to the desired value. Thereafter, the deviation angles of the individuals fluctuate within the given tolerance range and the optimisation focuses on the pressure loss.

In order to observe the self-adaptation of the strategy parameters, the global step size δ and the condition of the covariance matrix during the optimisation are shown in Figs. 16 (a) & (b). The condition of a matrix is the relation between its largest and smallest eigenvalue. If

all eigenvalues are equal, i.e. the condition of the matrix is one, the iso-density lines of the probability density function for the mutation of the parameters are circles and no adaptation of the strategy parameters has taken place. The larger the difference between the eigenvalues the more pronounced is the “cigar-shape” of the iso-density lines and the more distinct is the preference to one direction. The magnitude of the mutation is controlled by the global step size. In these figures it can be seen, that the global step size as well as the preferred search direction, given by the covariance matrix of the normal distribution, are adapted very early and continuously during the optimisation to the local structure of the quality function. To draw general conclusions, e.g. on the nature of the search space, from these observations which only rely on one particular optimisation run is difficult. However, we can conclude that a self-adaptation of the strategy parameters indeed

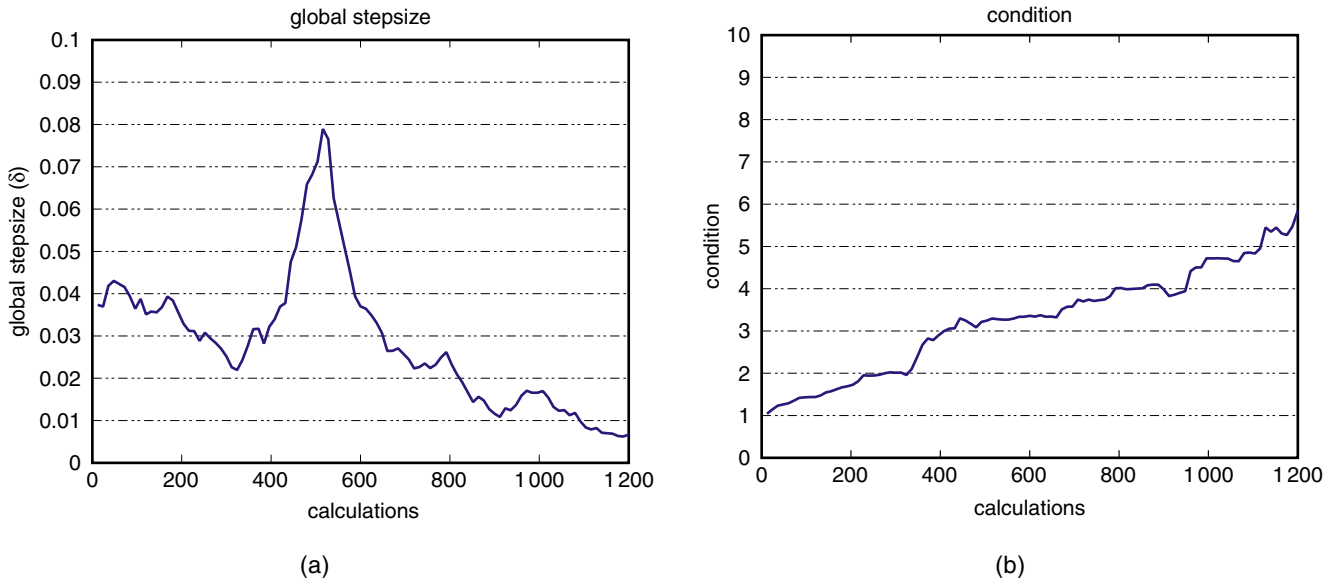


Fig. 16 (a) Global step size of the CMA algorithm and (b) condition of the covariance matrix.

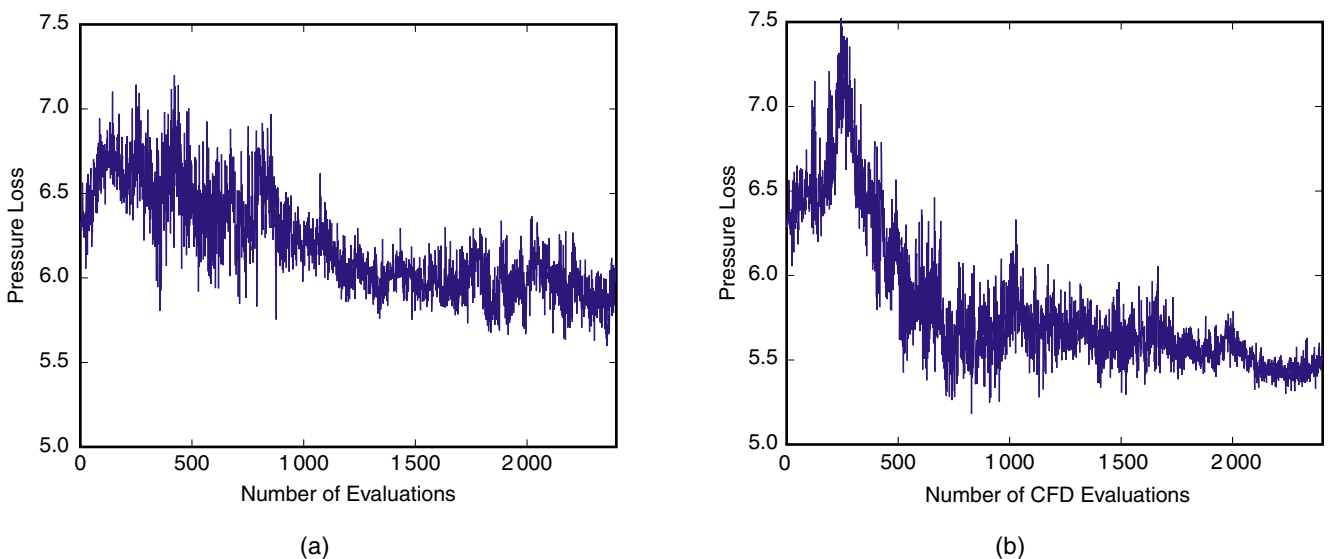


Fig. 17 Optimisation without (a) and with (b) the neural network.

occurs and that it is guiding the process of evolution to regions of the search space where high quality geometries can be identified.

Neural network approximation Due to the extremely high computational cost of CFD simulations it is beneficial to use approximation models for the estimation of the quality of geometries. Besides being universal approximators, neural networks are particularly suitable for this application because they can be adapted online to new search regions during the optimisation.

Even in case of a lower accuracy of the estimated quality compared to CFD simulations the overall calculation time for optimisations can be reduced often by a factor of two. In Fig. 17 (a) the pressure loss of generated geometries during an optimisation run is shown without using a neural network. In Fig. 17 (b) the result of the same optimisation is shown, but with additional use of the approximation model. On the x -axis of both figures the number of CFD evaluations is plotted. Since this is the time consuming part in the optimisation, this number is proportional to the overall computation time. Therefore, using the approximation model, we can increase the total number of evaluations while maintaining a constant overall computation time. In the example the quality which is reached after about 2000 CFD simulations can be reached with the help of an approximation model after some hundred CFD evaluations.

Physical measurement Besides high performance at the design point, another aim of the optimisation of turbine parts is good performance under variations of the working conditions. In Fig. 18 the total pressure loss ω for the optimised blade is plotted for different outlet velocities m_2 . The variations of m_2 are a result of the variations of the thrust in turbines used for the propulsion of aircrafts. The design point which was the target for the optimisation is at ca. $m_2 = 1.136$ *mach* which is marked by the black vertical line in the figure.

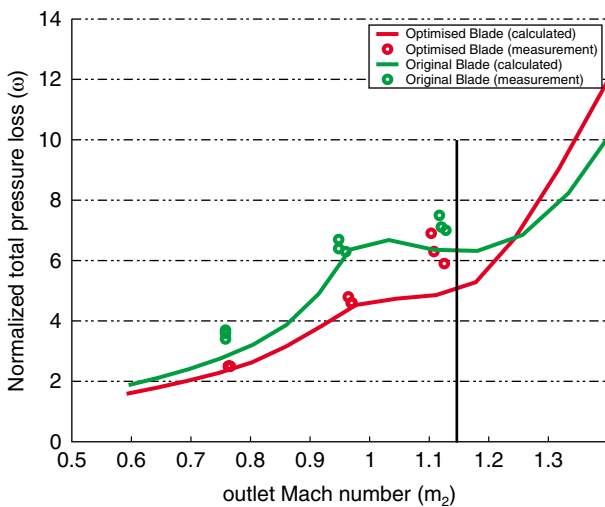


Fig. 18 CFD simulation and physical measurement results for the initial and the optimised blade geometry.

The green line shows the pressure loss for the initially available, original turbine blade whereas the red line shows the pressure loss for the optimised design.

In the simulation the total pressure loss at the design point of the optimised blade geometry is ca. 20% less compared to the original blade. The performance for larger and smaller velocities around the design point is also considerably improved, for smaller m_2 even to a larger degree than at the design point. This tendency is supported by the physical measurement at the DLR transonic wind tunnel for plane cascades in Göttingen, Germany; the data is shown as circles. Both the measurements, as well as, the simulations are very complicated and error-prone, therefore, slight variations like at the design point are to be expected. In addition to the substantial reduction in the pressure loss, the optimised blades did also show unique and previously unknown geometric properties, which exploited some nonlinearities in the flow around the blade to minimise the loss. The turbine experts were able to understand and interpret the flow physics induced by these geometric peculiarities and therefore, new design principles were formulated as a result of the optimisation.

5. Discussion and Summary

In this paper, we outlined the optimisation of transonic turbine blades, starting with the description of the evolution strategy, the optimisation algorithm, which we employed. When we choose the representation, we usually have to face a trade-off between compactness and flexibility. A compact representation allows us to operate in a lower dimensional search space, however, we might not be able to describe all geometrical variations. A higher degree of flexibility gives us a higher degree of freedom for the design, however, the optimisation task will be harder. In order to overcome this problem, recently, adaptive representations have been suggested which aim at combining the best of both worlds, i.e. starting with a compact representation and increasing the flexibility only at specific parts of the geometry during the optimisation, see Olhofer et al.⁽²⁹⁾

CFD simulations are computationally expensive, therefore, we proposed a framework in which approximate models, in our case neural networks, can be efficiently combined with the Navier-Stokes solver. The framework takes the actual model fidelity into account and thus, guarantees correct convergence while minimising the number of necessary evaluations using the CFD solver.

The results of the optimisation show that the evolution strategy is able to adapt to the specific topology of the search space. The pressure loss of the optimised blade geometry is reduced by 20% compared to the original, standard blade. Furthermore, it was possible to associate properties of the optimised flow field to geometric changes of the blade. Thus, new design *principles* have been extracted besides of generating one singular optimisation result.

The high quality of the simulation method was finally

demonstrated by comparing the physical measurement in a transonic windtunnel to the numerical results both for the original and the optimised blade.

New, "intelligent" optimisation methods together with the latest developments in numerical simulations of system properties, e.g. fluid-dynamics, and processes, e.g. controllers, are likely to play an increasing role in the future of system engineering. Although much remains to be learned from the biological principles of evolution, the optimisation set-up outlined in this work already provides a powerful framework for the solution of complex problems.

Acknowledgements

The authors would like to thank E. Körner and M. Hasenjäger for stimulating discussions and their support and advice.

References

- (1) T. Arslan, E. Ozdemir, M. S. Bright, and D. H. Horrocks. Genetic synthesis techniques for low-power digital signal processing circuits. In *Proceedings of the IEE Colloquium On Digital Synthesis*, pages 7/1-7/5. IEE, London, UK, February 1996.
- (2) R. S. Zebulum, M. A. Pacheco, and M. Vellasco. Synthesis of CMOS operational amplifiers through Genetic Algorithms. In *Proceedings of the Brazilian Symposium on Integrated Circuits, SBCCI'98*, pages 125-128, Rio de Janeiro, Brazil, September 1998. IEEE.
- (3) M. S. Bright and T. Arslan. Multi-objective design strategies for high-level low-power design of DSP systems. *IEEE International Symposium on Circuits and Systems, ISCAS 99*, 1:80-83, May-June 1999.
- (4) G.M. Robinson. Load cell shape optimisation using genetic algorithms and finite elements analysis. In *Proceedings of the 14th IMEKO TC3 Conference*, pages 167-172, Warsaw, Poland, 1995.
- (5) A.J. Keane and S.M. Brown. The design of a satellite boom with enhanced vibration performance using genetic algorithm techniques. In I.C. Parmee, editor, *Proceedings of the Conference on Adaptive Computing in Engineering Design and Control 96*, pages 107-113, University of Plymouth, Plymouth, 1996.
- (6) Xin Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423-1447, 1999.
- (7) B. Sendhoff and M. Kreutz. Variable encoding of modular neural networks for time series prediction. In V.W. Porto, editor, *Congress on Evolutionary Computation CEC*, pages 259-266. IEEE Press, 1999.
- (8) H. Furuta, K. Maeda, and W. Watanabe. Application of genetic algorithm to aesthetic design of bridge structures. In *Microcomputers in Civil Engineering*, 10/6:415-421, 1995.
- (9) S. Obayashi, Y. Yamaguchi, and T. Nakamura. Multiobjective genetic algorithm for multidisciplinary design of transonic wing planform. *Journal of Aircraft*, 34(5):690-693, 1997.
- (10) J. Sobiesanski-Sobieski and R. T. Haftka. Multidisciplinary aerospace design optimization: Survey of recent development. In *34th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 1996. AIAA Paper 96-0711.
- (11) A. Vicini and D. Quagliarella. Airfoil and wing design through hybrid optimization strategies. In *16th Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics, 1998. AIAA Paper 98-2729.
- (12) D.J. Doorly. Parallel genetic algorithms for optimization in cfd. In J. Périaux and G. Winter, editors, *Genetic Algorithms in Engineering and Computer Science*. Wiley, 1995.
- (13) Brian R. Jones, William A. Crossley, and Anastasios S. Lyrintzis. Aerodynamic and aeroacoustic optimization of airfoils via a parallel genetic algorithm. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1998. AIAA Paper 98-4811.
- (14) D.B. Fogel. *Evolutionary Computation*. IEEE Press, 1995.
- (15) H.-P. Schwefel. *Evolution and Optimum Seeking*. John Wiley & sons, New York, 1995.
- (16) N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159-196, 2001.
- (17) M. Olhofer, T. Arima, T. Sonoda, and B. Sendhoff. Optimization of a stator blade used in a transonic compressor cascade with evolution strategies. In I. Parmee, editor, *Adaptive Computation in Design and Manufacture (ACDM)*, pages 45-54. Springer Verlag, 2000.
- (18) T. Arima, T. Sonoda, M. Shirotori, A. Tamura, and K. Kikuchi. A numerical investigation of transonic axial compressor rotor flow using a low-Reynolds number κ - ϵ turbulence model. *Journal of Turbomachinery*, 121:44-58, 1999. Transactions of the ASME.
- (19) K. Y. Chien. Predictions of channel and boundary-layer flows with a low reynolds number turbulence model. *AIAA Journal*, 20:33-38, 1982.
- (20) T. H. Pulliam and D. S. Chaussee. A diagonal form of an implicit approximate factorization algorithm. *Journal of Computational Physics*, 39:347-363, 1981.
- (21) S. R. Chakravarthy and S. Osher. A new class of high accuracy tvd schemes for hyperbolic conservation laws. In *AIAA 23rd Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, 1985. AIAA Paper 85-0363.
- (22) C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, UK, 1995.
- (23) Y. Jin and B. Sendhoff. Knowledge incorporation into neural networks from fuzzy rules. *Neural Processing Letters*, 10(3):231-242, 1999.
- (24) Y. Jin, M. Olhofer, and B. Sendhoff. On evolutionary optimization with approximate fitness functions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 786-792. Morgan Kaufmann, 2000.
- (25) J. Redmond and G. Parker. Actuator placement based on reachable set optimization for expected disturbance.

- Journal Optimization Theory and Applications*, 90(2):279-300, August 1996.
- (26) A. Ratle. Optimal sampling strategies for learning a fitness model. In *Proceedings of 1999 Congress on Evolutionary Computation*, volume 3, pages 2078-2085, Washington D.C., July 1999.
 - (27) Y. Jin, M. Olhofer, and B. Sendhoff. Managing approximate models in evolutionary aerodynamic design optimization. In *Congress on Evolutionary Computation (CEC)*, volume 2, pages 592-599, Seoul, Korea, May 2001.
 - (28) A. Geist, A. Beguellini, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *Parallel Virtual Machine: A Guide to Networked and Parallel Computing*. The MIT Press, 1994.
 - (29) M. Olhofer, Y. Jin, and B. Sendhoff. Adaptive encoding for aerodynamic shape optimization using evolution strategies. In *Congress on Evolutionary Computation (CEC)*, volume 2, pages 576-583, Seoul, Korea, May 2001.

■ 著 者 ■



Markus Olhofer



Toshiyuki Arima



Yaochu Jin



Toyotaka Sonoda



Bernhard Sendhoff