

Three dimensional evolutionary aerodynamic design optimization with CMA-ES

Martina Hasenjäger, Bernhard Sendhoff, Toyotaka Sonoda, Toshiyuki Arima

2005

Preprint:

This is an accepted article published in {GECCO 2005}: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation. The final authenticated version is available online at: [https://doi.org/\[DOI not available\]](https://doi.org/[DOI not available])

Three Dimensional Evolutionary Aerodynamic Design Optimization with CMA-ES

Martina Hasenjäger
Honda Research Institute Europe GmbH
Carl-Legien-Str. 30
63073 Offenbach/Main, Germany
martina.hasenjaeger@honda-ri.de

Bernhard Sendhoff
Honda Research Institute Europe GmbH
Carl-Legien-Str. 30
63073 Offenbach/Main, Germany
bs@honda-ri.de

Toyotaka Sonoda
Honda R&D Ltd.
1-4-1 Chuo, Wako-shi
Saitama 351-0193, Japan
toyotaka_sonoda@n.n.rd.honda.co.jp

Toshiyuki Arima
Honda R&D Ltd.
1-4-1 Chuo, Wako-shi
Saitama 351-0193, Japan
toshiyuki_arima@n.w.rd.honda.co.jp

ABSTRACT

In this paper, we present the application of evolutionary optimization methods to a demanding, industrially relevant engineering domain, the three-dimensional optimization of gas turbine stator blades. This optimization problem is high-dimensional search and computationally very expensive. We show that, despite of its difficulty, the problem is feasible. Our approach not only successfully optimizes the aerodynamic design but also yields interesting results from an engineering point of view.

Categories and Subject Descriptors

J.2 [Physical sciences and engineering]: Aerospace; I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Design, Algorithms

Keywords

Design optimization, evolutionary strategies, covariance matrix adaptation, real world application

1. INTRODUCTION

The availability of relatively inexpensive, massive computing power in form of PC clusters made it possible to treat complex, computationally intensive application problems and to verify the everyday or real world suitability of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'05, June 25–29, 2005, Washington, DC, USA.
Copyright 2005 ACM 1-59593-010-8/05/0006 ...\$5.00.

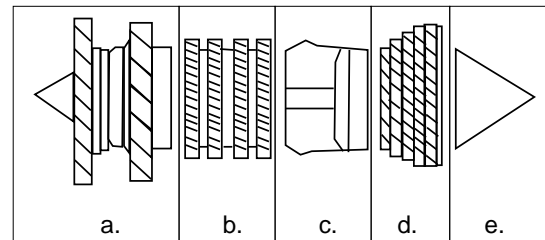


Figure 1: Schematic sketch of a gas turbine. The turbine consists of a fan (a.) that pulls air into the engine. Part of this air is compressed in the compressor (b.) and then forced into the combustion chamber (c.) where it is mixed with fuel and ignited. The resulting hot, high energy gases go into the turbine (d.) causing the turbine blades to rotate. The task of the turbine is to convert gas energy into mechanical work to drive the compressor (b.). The nozzle (e.) is the exhaust duct of the engine.

algorithms that hitherto were only tested on toy problems or in simplified application domains.

One of the problems on the verge of tractability even today is the three-dimensional optimization of aerodynamic designs. The difficulty of this optimization task stems on the one hand from the difficulty and high computational costs of 3D computational flow analysis and on the other hand from the fact that 3D flows are not fully understood yet.

In this paper, we present an approach to 3D aerodynamic design optimization using evolutionary strategies. We show that this approach is indeed feasible and not only successfully optimizes the aerodynamic design but also yields interesting results from an engineering point of view.

The aerodynamic design that we optimize is part of a gas turbine that is used in small business jets. The main parts of a gas turbine are depicted schematically in Fig. 1. In the current research we focus on the turbine which is composed of several rows of airfoil cascades. Some of these rows, the rotors, are connected to the central shaft of the engine

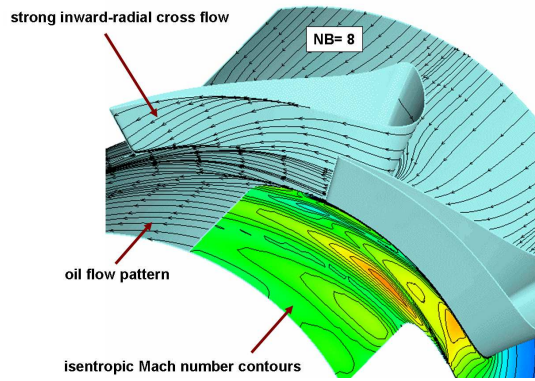


Figure 2: Ultra low aspect ratio turbine stator blades and their flow fields. The blade count is $NB = 8$.

and rotate at high speed thus driving the engine’s fan and compressor and converting gas energy to mechanical energy. The other rows, the stators, are fixed and serve to keep the flow from spiralling around the axis. Our goal is to optimize the design of the turbine stator blades which are of a special type in our case. The stator is a so-called ultra-low-aspect-ratio (ULAR) stator. This means that a stator row is made up of only 8 stator blades. This is a very small number compared to 20 - 60 blades that are used in more conventional turbine designs. Fig. 2 depicts part of a ULAR stator row. The ULAR blades offer considerable benefits as compared to more conventional blades. For example, the blade resonance in the following rotor cascade is reduced. In general, however, ULAR stator blades are only rarely used in gas turbines because of their relatively poor stage efficiency that is much lower than the efficiency of the more conventional high aspect ratio turbines. The flow field of the ULAR stator that is depicted by arrows in Fig. 2 shows the reasons for the poorer performance. There is a very strong inward-radial cross flow on the blade suction side. Also the flow field near the hub-end-wall is very complicated due to the interaction of the secondary flow with the transonic main flow.

As a result, the aerodynamic loss near the hub region is considerably increased as compared to the conventional high-aspect-ratio blade. It is this higher loss near the hub region that prevents a further improvement of the stage efficiency. The ULAR blade flow characteristics make it unlikely that the advanced design principles developed for high aspect ratio blades will help to improve the efficiency of ULAR stator blades: The flow phenomena that can be observed are too different to enable exploitation of design principles for high aspect ratio blades directly.

Therefore, ULAR stator blades are excellent candidates for numerical optimization. The goal of optimization from the engineering point of view is not only to increase blade efficiency, i.e., ultimately to reduce the engine’s fuel consumption, but also to find new design concepts for suppressing the secondary flow loss near the hub-end-wall region and hence to enable control of the three-dimensional nature of the flow.

From the optimization point of view 3D aerodynamic design optimization constitutes an interesting problem for several reasons. We have to deal with a high-dimensional search

space while evaluation of the objective function is so costly that an appropriate acquisition of data from the search space is hardly feasible. This means that we have to use techniques that reduce the dimensionality of the search space as much as possible and we have to use algorithms that are able to cope with extremely sparse data.

Also the formulation of the optimization objective is not as obvious as it may seem. Traditionally, the minimization of the aerodynamic loss, i.e., the average pressure loss, is chosen as optimization target. However, we must not forget that this is an extreme simplification of the problem. In fact, the pressure loss is only one among many quantities that an engineer takes into account when assessing the blade performance. To compound matters, three-dimensional aerodynamic flow is not fully understood yet, so that it is difficult to foresee the effects that may result from the choice of a particular combination of optimization objectives.

In this paper, we consider two formulations of the objective function. As a first guess we use the single objective of minimization of the mass-averaged total pressure loss. While this yields satisfactory results, we found that it is desirable to include a second objective, the minimization of the variation of the circumferential static pressure distribution. This renders the problem multi-objective. As a first approach we address the multi-objective optimization problem in the simplest way, namely by linear aggregation of the two objectives.

Three-dimensional design optimization is not only an interesting problem from the engineering and optimization point of view but also from a technical point of view the design of a stable and fault tolerant computing environment that includes a whole bunch of computers and a variety of software components is a challenge.

In this paper, we will discuss 3D aerodynamic design optimization from all three above mentioned perspectives. In Sec. 2 we present the ultra-low-aspect-ratio transonic stator blades that will be optimized. In Sec. 3, we discuss the components of the optimization problem, i.e., the choice of blade model, the formulation of the fitness function, and the architecture of the simulation environment. The optimization results are presented and discussed in Sec. 4. Finally, our conclusions are given in Sec. 5.

2. ULTRA-LOW-AR TRANSONIC STATOR BLADES

Low aspect ratio turbine stator blades have rarely been adopted as components of conventional turbines because of their relatively poor performance which is caused primarily by increases of secondary flow losses due to the low aspect ratio. However, there are considerable benefits when adopting low aspect ratio blades. For example, for a low number of stator blades, rotor blade resonance, and hence material fatigue, is considerably reduced.

The stator that we optimize consists of only eight blades and the aspect ratio AR , i.e., the ratio of span height to axial chord-length, is $AR = 0.3$. The stator blades are circumferentially leaned by 14 degrees in order to suppress the development of secondary flow near the hub-end-wall. Fig. 2 shows a three dimensional outline of the stator blades and their flow fields. For details on the design specifications of the ULAR stator refer to [9].

Due to manufacturing reasons, the blade geometry is defined

by two sections, the hub section and the tip section. The remaining blade geometry is defined by linear interpolation between these two sections.

In the following, we will refer to this blade as the *baseline blade*.

3. DESIGN OPTIMIZATION WITH EVOLUTIONARY ALGORITHMS

Evolutionary algorithms [2] are a class of stochastic optimization algorithms whose use in design optimization problems is well established by now [12].

In our approach to 3D turbine blade optimization we use a special variant of evolutionary algorithms namely an evolution strategy (ES) with covariance matrix adaptation (CMA) [8]. The basic idea of CMA-ES is to make maximum use of the information contained in the search history for self-adaptation of the search direction that is defined in terms of the covariance matrix of a normal distribution from which individuals are drawn. Thereby the population size is decoupled from the dimension of the search space.

Especially the latter feature is indispensable in 3D blade optimization which is characterized by a fundamental conflict: on the one hand the design space is very high-dimensional. As a consequence a large number of different designs has to be evaluated during optimization. On the other hand each evaluation of the blade performance is a computationally extremely demanding task so that only a limited number of evaluations can be afforded.

3.1 Blade Model

A crucial point in design optimization is the parametric model of the geometry that will be optimized since this determines the design space.

There are a number of requirements on the design of a proper blade model. Among these are

1. flexibility: the model must be flexible enough to allow for a wide variety of different designs,
2. compactness: the number of parameters describing the model must be low enough to allow for reasonable convergence times of the optimization algorithm, and
3. locality: variations of a single model parameter should result in only local variations of the model and should not affect the global model shape.

A good choice to fulfill these requirements is to use non-uniform rational B-spline (NURBS) surfaces [5] to represent the blade. A B-spline surface is a tensor product of two B-spline curves and hence is defined by two parameters, a set of control points and two knot vectors, one for each parameter. Usually not all of these parameters are subject to optimization. Often the variables in the design optimization problem are given by the coordinates of the control points. However, the suitable number of control points must be chosen with care: the use of too few control points may unnecessarily restrict the design space and exclude potentially interesting designs while the use of too many control points complicates the optimization problem and additionally may have unwanted side-effects like the creation of cusps or even self-intersections of the resulting surfaces.

As mentioned in Sec. 2, the geometry of the baseline blade is defined by two cross sections, the tip section and the hub

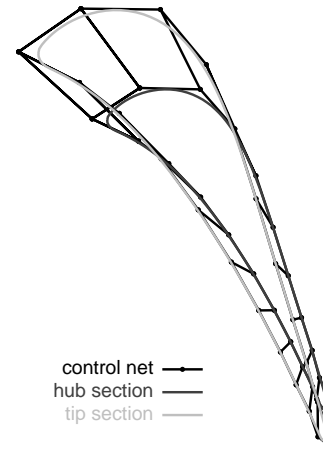


Figure 3: The blade model is created from the hub section (dark gray) and the tip section (light gray) of the baseline blade. These sections are defined by 25 control points each. The control net of the baseline surface model, that connects neighboring control points is depicted by a black line.

section. Our blade model consists of a B-spline surface defined by a periodically closed cubic B-spline in one parameter direction and a second order open B-spline in the other direction. The hub section and the tip section of the blade are each modeled using 25 control points so that all in all 50 control points are used. The control net of the blade model and the section geometries are shown in Fig. 3.

Using the coordinates of the 50 three-dimensional control points directly as design variables would result in a 150-dimensional search space. Fortunately, we can exploit two facts to reduce the search space dimension to only 88:

1. We note that we use closed periodic splines in the first parameter direction of the blade surface model to achieve a closed and seamless shape that has no beginning or end points. This implies that the first d and the last d control points of each blade section coincide. Here d denotes the degree of the splines which is $d = 3$ in our case of cubic splines. This means that each of the two blade sections is defined by only $25 - 3 = 22$ independent control points. The periodic control points need not be taken into account as design variables so that in total we only have to consider 44 control points.
2. The hub section as well as the tip section of the blade are defined to lie on cylindrical surfaces. This means the z -coordinates of the control points are implicitly fixed by the blade geometry. Hence we only need to consider the x - and y -coordinates of the non-periodic control points as design variables and so we are left with only $2 \times 44 = 88$ design parameters.

The knot vectors are not subject to optimization.

3.2 Fitness Functions

The performance measure of a specific blade design was given by a weighted sum of the two objectives of the total mass averaged pressure loss ω and the maximum variation of the pitch-wise static outlet pressure PST_{VAR} . Minimization

of these aggregated objectives was subject to a number of constraints as detailed below:

$$f = w_1 t_1 + w_2 t_2 + \sum_{i=3}^6 w_i t_i^2 \rightarrow \min \quad (1)$$

with

- t_1 pressure loss ω
- t_2 variation of the static outlet pressure PST_{VAR}
- t_3 $\max(0, |\beta_{2,design} - \beta_2| - \delta\beta_2)$
- t_4 $\max(0, \Theta_{min,design} - \Theta_{min})$
- t_5 $\max(0, \Theta_{TE,min,design} - \Theta_{TE,min})$
- t_6 $\max(0, s_{max} - s_{max,design})$.

That means, the outflow angle β_2 , which is determined as a result of the flow analysis was constrained to lie in a range of $\delta\beta_2$ around the design value $\beta_{2,design}$. The other constraints concern the blade geometry and basically can be considered as manufacturing constraints. These are the minimum blade thickness Θ_{min} that was constrained to be bigger than the design value $\Theta_{min,design}$, the minimum trailing edge thickness Θ_{min} that had to be at least $\Theta_{TE,min}$, and the blade solidity s_{max} that was constrained from above by $s_{max,design}$.

We considered two variants of the fitness function Eq. 1: $w_2 = 0$, i.e., only the pressure loss ω was minimized, and $w_1 t_1 \approx w_2 t_2$, i.e., the pressure loss ω and an additional criterion, the pitch-wise variation of the static outlet pressure PST_{VAR} were jointly minimized where PST_{VAR} is defined as

$$PST_{VAR} = \max_{i=1 \dots K} \left(\left| \max_{j=1 \dots L} \{PST(i,j)\} - \min_{j=1 \dots L} \{PST(i,j)\} \right| \right). \quad (2)$$

Here i is the span-wise and j the pitch-wise index in the computational grid that was used in the flow analysis. Initially both objectives received the same weight. This constitutes a simple, naïve approach to multi-objective optimization.

The baseline blade model lies within the feasible region of the design space. According to Eq. 1 only violated constraints contribute to the objective function. The weights on the constraints were chosen such that the contribution of a violated constraint by far outweighs the contribution of the objectives in order to quickly drive the search back into the feasible region.

3.3 Grid Generation and Flow Solver

For the simulation of the fluid dynamic properties of the blade designs we used the parallelized 3D Navier-Stokes flow solver HSTAR3D [1] with Wilcox's $k-\omega$ two equations model [14]. The computational grid for the solution of the Navier Stokes equation consisted of $175 \times 52 \times 64 = 582,400$ cells. For each evaluated blade design a new grid was generated. This is a relatively inexpensive operation that takes on average about 40 seconds on an AMD Opteron 2 GHz double processor.

The flow analysis, however, is an extremely time consuming task that takes between 2 hours and 3.5 hours on an AMD Opteron 2 GHz double processor depending on the number of flow solver iterations needed for convergence. So the calculation of the results presented here, i.e., about 300 generations of the evolutionary optimization, took about 6 weeks time.

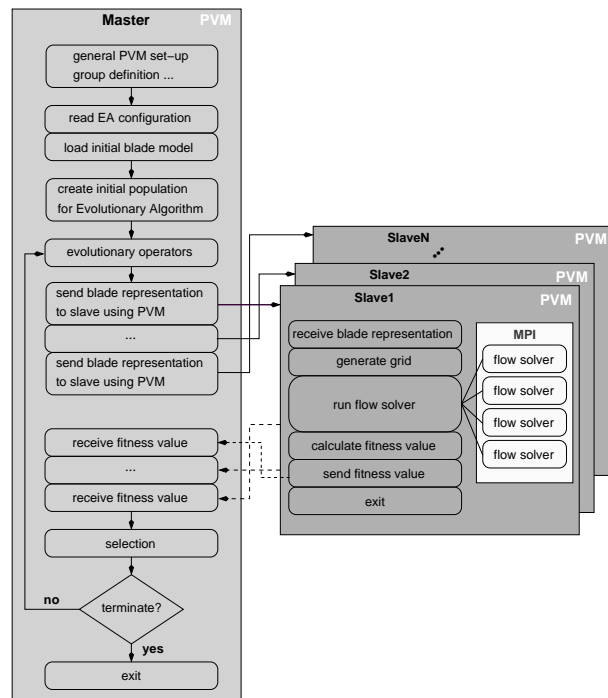


Figure 4: Simulation environment architecture. The program is parallelized at 2 levels: the first level of parallelization is a master-slave model that uses PVM to organize the distribution of single individuals to slave processes while the second level that is started by the slave processes is a node-only model for parallelizing the flow solver calculations using MPI.

3.4 Optimization Environment

To conduct the blade optimization, we designed and implemented a simulation environment that is highly configurable and at the same time hides much of the complexity of running a large scale simulation from the user.

Using evolutionary optimization is ideally suited to parallelization. In our case the fitness evaluation is the most time consuming task we have to solve. Thus we decided to evaluate the individuals' fitness in parallel using a master-slave model where the master is responsible for the organization and execution of the evolutionary cycle except for the evaluation of the fitness function. For each offspring individual, the master spawns a slave process to take care of this. The slave processes generate the computational grids, run the flow solvers, calculate the fitness values from the CFD results and return them to the master.

Note that there is a second level of parallelization in the slave processes: the flow solver itself is also parallelized into 4 processes. This means the evaluation of λ offspring individuals requires to manage 4λ processes! The architecture of the simulation environment is shown in Fig. 4. The flow solver is parallelized using MPI [10], while the master-slave model of the optimization loop was implemented using the Parallel Virtual Machine (PVM) library [6].

Large scale applications that involve extremely long run times like the one discussed here raise interesting questions concerning hard- and software stability and fault tolerance.

Ideally the failure of one or more of the involved hosts should be intercepted either by migrating the jobs from the affected machine to another machine or by rescheduling and/or restarting the jobs on the next available machine. Another option which is especially applicable in population based methods like evolutionary algorithms would be to tacitly ignore the host failure and simply proceed with a smaller population in the affected generation.

In any case some kind of check-pointing is highly advisable that regularly saves the internal state of the simulation and in this way allows for a restart of the calculation after a crash without noticeable loss of results or the need for heavy recalculations.

4. RESULTS AND DISCUSSION

For optimization of the baseline stator blade, we used the blade model as described in Sec. 3.1. We used two variants of the fitness function Eq. 1 as explained in Sec. 3.2. As optimization algorithm we used a (1,10)-CMA-ES according to [7] with the standard parameter settings. The population size was determined partly by experience gained with similar problems and partly out of practical considerations: the population size was limited by the available hardware, cf. Sec. 3.4. The strategy parameters were initialized with $\sigma_{\text{init}} = 0.1$. This value gave good results in preliminary tests with a wider range of possible initialization values. We did not use recombination. The simulations were run on a cluster of AMD Opteron 2GHz double processors.

In the discussion of the results, we will first give an account of the optimized blade from an engineering point of view and then discuss the results from the point of view of optimization.

The blade geometry resulting from minimizing only the pressure loss, i.e., using Eq. 1 with $w_2 = 0$, is shown in Fig. 5. The hub and tip sections of the optimized blade are shown in Fig. 5 (a) and (b) together with the baseline sections. In the hub section of the optimized blade, the curvature of the frontal part on the suction surface is considerably reduced and the trailing edge part is bent in the upstream direction, see Fig. 6 for a zoom into the trailing edge region of the blade.

A closer analysis of the flow shows that the loading pattern at the hub is significantly shifted toward downstream. This tendency is also observed for the tip section. Also there is a large difference between the baseline blade and the optimized blade in the boundary layer development on the blade suction side. It seems that due to a pitch-wise weaker driving force the strong inward radial cross flow on the suction surface observed in the baseline blade is weakened and/or shifted more downstream in the optimized blade.

Fig. 7 shows the development of the fitness function in this simulation setting. The fitness had to be minimized and we obtain a decrease in fitness of roughly 10% which is quite reasonable for this problem and for the restricted number of generations that the algorithm could run. Note that the optimization has not converged yet but that this is rather a snapshot taken after a relatively short time in terms of the optimization if we consider the problem size. However, the absolute calculation time for these results was quite long – about 6 weeks on 40 processors.

Fig. 9 gives the evolution of the strategy parameters of the CMA-ES, i.e., the global strategy parameter in Fig. 9 (a) and the condition number of the covariance matrix in

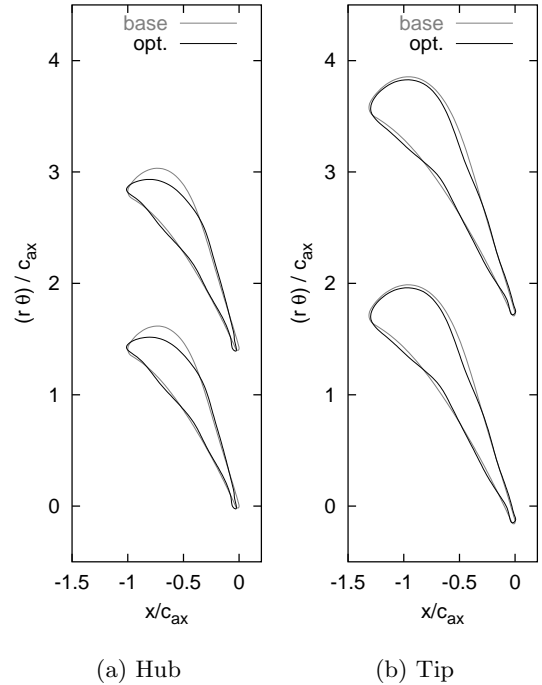


Figure 5: Comparison of blade geometries for (a) the hub section and (b) the tip section of baseline blade (gray) and optimized blade (black), resp.

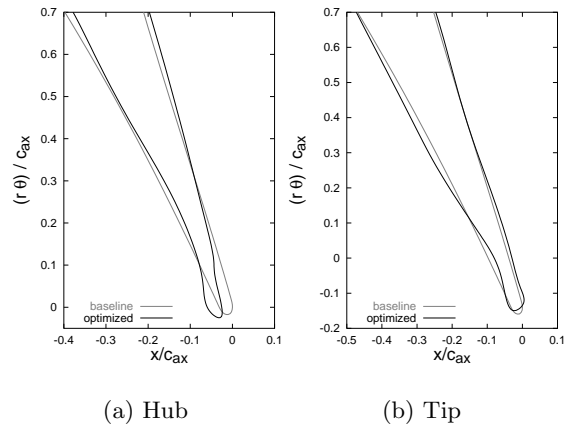


Figure 6: Comparison of the TE parts of the blade geometries for (a) the hub section and (b) the tip section of baseline blade (gray) and optimized blade (black), resp.

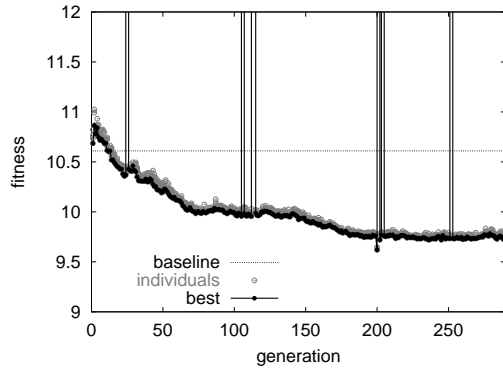


Figure 7: Fitness as a function of the number of generations. The dotted black line gives the fitness of the baseline blade. The fitness values of the single individuals are marked by open circles where the best individuals of each generation (black circles) are connected by a black line. The vertical lines in the plot denote cases in which even the best individual of the population does not comply with the constraints and hence receives a high penalty.

Fig. 9 (b). The results indicate that the self-adaptation of the parameters works properly and that the process indeed detects a search direction.

Finally, we show the optimization objective and the constraints for this simulation in Fig. 8. The pressure loss in Fig. 8 (a) is essentially equivalent to the fitness in Fig. 7. The vertical lines in Fig. 7 indicate that in some cases even the fitness of the best individual of the population lies outside of the range of the plot. This means that the complete population moved out of the feasible region of the optimization problem. A comparison with Fig. 8 (e)-(f) shows that these outliers coincide with the violation of some constraint. In principle, this is critical because the population may be lead astray and not be driven back into the feasible region thus causing a failure of the optimization process. But here these outliers are smoothly absorbed by the optimizer and do not disturb the process. This demonstrates the stability of the optimization method which is especially important in such a real world application that is susceptible to numerous sources of disturbance. For example, it may also happen that the computational grid needed by the flow solver cannot be generated for all individuals or that the flow solver fails for some reason so that not all individuals can be attributed with a fitness value.

Condensing the blade performance index into one single objective value of average pressure loss is a practical approach to render the problem feasible for optimization. However, it turns out that this is not sufficient from an engineering point of view.

A closer analysis of the fluid dynamic properties of the optimized blade shows that the pitch-wise variation of the static pressure of the optimized blade is considerably increased as compared to the baseline blade, cf. Fig. 8(b). However, strong pitch-wise variations of the static outlet pressure are not desirable even for non-resonance rotor blade designs as in our case. They might lead to unsteady losses induced by the interaction between stator and rotor. Therefore, we started a second optimization where we combined the min-

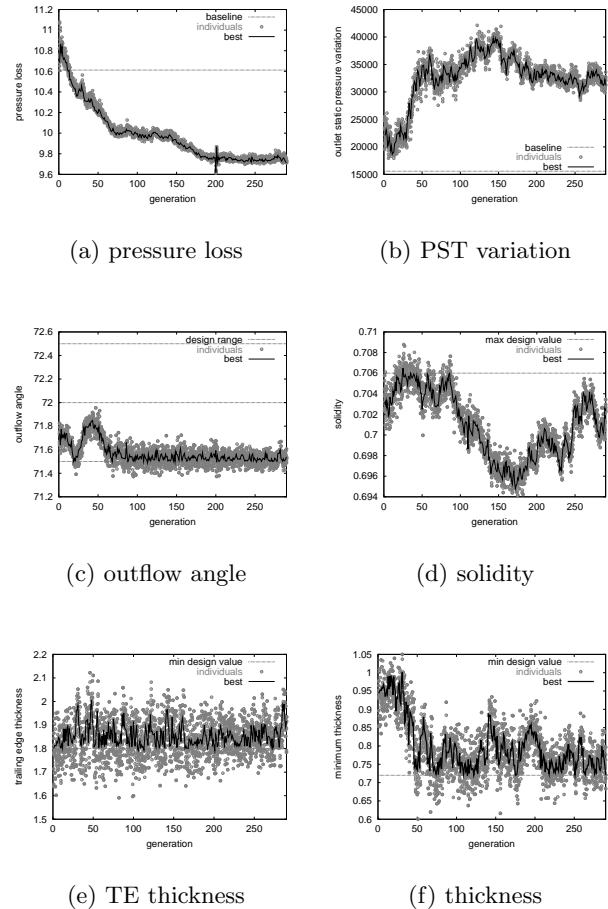
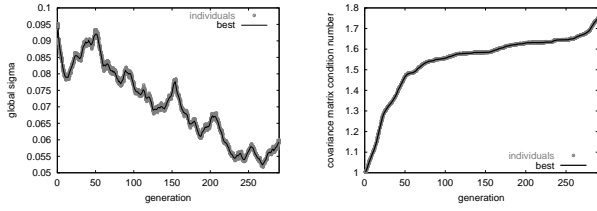


Figure 8: Objectives and constraints during optimization: (a) the pressure loss that was subject to minimization, (b) PST variation that was not used as an objective in this simulation, and the constraints (c) the outflow angle, (d) the solidity, (e) the trailing edge thickness, (f) the minimum blade thickness. The single individuals are marked by open circles, the fitness-best individuals of each generation are connected. For the constraints the design range is indicated by a dotted line.



(a) global strategy parameter

(b) covariance matrix condition number

Figure 9: Strategy parameters during optimization: (a) the global strategy parameter and (b) the condition number of the CMA covariance matrix. The single individuals in the population are marked by open circles, the fitness-best individuals of each generation are connected.

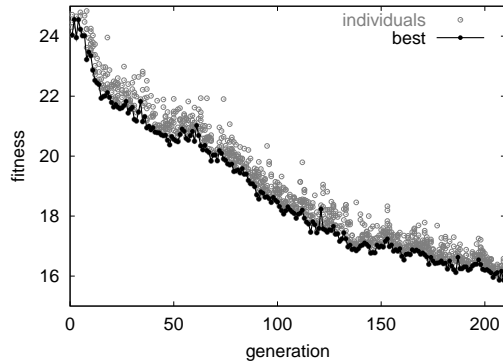
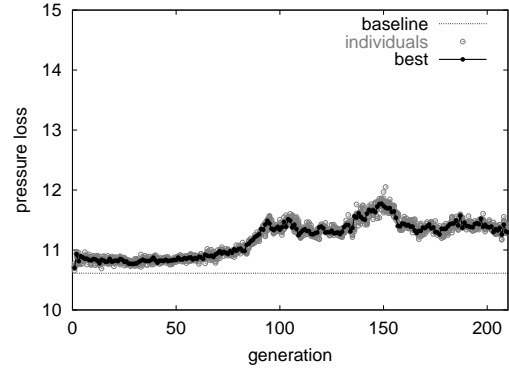


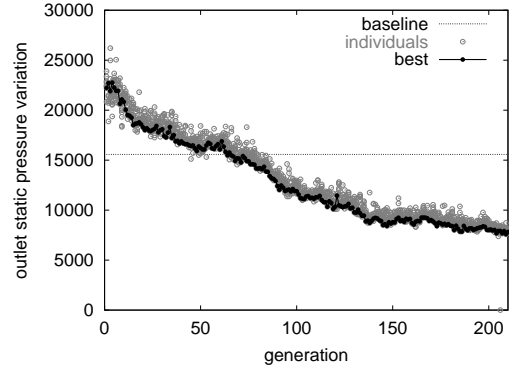
Figure 10: Fitness as a function of the number of generations. As fitness function we used an aggregation of the two criteria of pressure loss minimization and PST variation minimization. The fitness values of the single individuals are marked by open circles where the best individuals of each generation are connected by a line.

imization of the pressure loss with the minimization of the static pressure variation. Here we used Eq. 1 with $w_2 \neq 0$.

Fig. 10 shows the evolution of the fitness in this case. We achieved a fitness reduction of about 30%. A closer look at the two constituents of the fitness function in Fig. 11 reveals that this reduction is mainly due to a reduction of the variation in the static outlet pressure, cf. Fig. 11(b). The pressure loss, however, is slightly increased, cf. Fig. 11(a). This is not a poor result from the engineering point of view. Nevertheless, a simultaneous minimization of both the pressure loss and the static outlet pressure still remains to be achieved.



(a) pressure loss



(b) PST variation

Figure 11: Quantities contribution to the fitness: (a) the pressure loss, and (b) the PST variation. Both were subject to minimization. The single individuals of the offspring population are marked by open circles, the fitness-best individuals of each generation (filled circles) are connected by a line.

5. CONCLUSION

In this paper, we applied an evolutionary algorithm, namely CMA-ES, to the solution of a large scale, complex, real world optimization problem.

We described the simulation environment that we developed to solve aerodynamic optimization problems and demonstrated the feasibility of our approach.

From an aerodynamic point of view one goal of using design optimization instead of resorting to traditional engineering knowledge for aerodynamic blade design is to identify new design concepts from the optimization results. This goal was achieved in this study and we can draw the following conclusions:

- (i) The optimization resulted in a blade with a noticeable bent part near the trailing edge. This extremely aft-loaded blade is effective for the reduction of secondary losses.
- (ii) The flow mechanism for reducing the secondary flow loss seems to be the reduction of the pitch-wise driving force resulting in a reduced migration of low momentum fluid near the tip and/or hub casing around the mid-axial chord position.
- (iii) This leads to a considerably delayed development of the boundary layer on the suction surface.

From the optimization point of view the combination of the different criteria and constraints constitute an interesting problem. In particular the relation between the pressure loss and the variation of the static outlet pressure deserves further attention.

The trade-off relation between both objectives has been previously pointed out in the literature [13]. In this paper, we employed the simplest way to combine the two objectives, namely linear aggregation. Its advantage is that the problem remains single objective and more efficient algorithms can be used than are available for multi-objective optimization. The drawback is – besides the linearity of the combination – the ad-hoc choice of the relative weights. Indeed as our results show, the static pressure variation has received too much attention during the optimization. In other words, the selection pressure toward smaller pressure loss values was not sufficient. There are two principle ways to proceed. First, we can avoid the combination of both objectives and use multi-objective optimization methods [4, 3] to determine the Pareto front. Second, we can render the second criterion a constraint. Thus, we set the variation of the static outlet pressure of the baseline blade as a soft constraint, so that slight overshooting is penalized only slightly, and solve the remaining single objective optimization problem of minimizing only the pressure loss. We intend to follow both pathways in future optimizations.

Another topic for future research is the representation of the blade model. In this study, we used a B-spline surface representation. However, the number of control points necessary to achieve sufficient flexibility with this kind of model may become unfeasibly large since this number directly determines the dimension of the search space and hence the convergence time of the optimization algorithm. This trade-off between the flexibility of the model and the search space dimension cannot be resolved easily if the representation is static. Instead, a dynamic and adaptive representation is required as proposed in [11].

6. REFERENCES

- [1] 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 $k - \epsilon$ turbulence model. *ASME Journal of Turbomachinery*, 121:44–58, 1999.
- [2] T. Bäck, D. B. Fogel, and T. Michalewicz, editors. *Evolutionary Computation 1: Basic Algorithms and Operators*. Institute of Physics, 2000.
- [3] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer, 2002.
- [4] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 2002.
- [5] G. Farin. *Curves and Surfaces for Computer-Aided Geometric Design*. Academic Press, San Diego, 4 edition, 1997.
- [6] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Mancheck, and V. Sunderam. *PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Network Parallel Computing*. MIT Press, 1994.
- [7] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pages 312–317, 1996.
- [8] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- [9] N. Kuno and T. Sonoda. Flow characteristics in a transonic ultra-low-aspect-ratio axial turbine vane. *Journal of Propulsion and Power*, 20(4):596–603, 2004.
- [10] MPI: A message-passing interface standard, <http://www-unix.mcs.anl.gov/mpi/>.
- [11] M. Olhofer, Y. Jin, and B. Sendhoff. Adaptive encoding for aerodynamic shape optimization using evolution strategies. In *Congress on Evolutionary Computation (CEC)*, pages 576–583, Seoul, Korea, 2001. IEEE Press.
- [12] A. Osyczka. *Evolutionary Algorithms For Single And Multicriteria Design Optimization*. Physica-Verlag, 2002.
- [13] M. L. Shelton, B. A. Gregory, S. H. Lamson, H. L. Moses, R. L. Doughty, and T. Kiss. Optimization of a transonic turbine airfoil using artificial intelligence, CFD and cascade testing. ASME Paper 93-GT-161, 1993.
- [14] D. C. Wilcox. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA Journal*, 26:1299–1310, 1988.