Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Exploring the multi-modal fitness landscape of a realistic turbofan engine blade optimization

Jakub Kmec, Sebastian Schmitt

2018

Preprint:

This is an accepted article published in EngOpt 2018 - 6th International Conference on Engineering Optimisation. The final authenticated version is available online at: https://doi.org/10.1007/978-3-319-97773-7_46

Exploring the fitness landscape of a realistic turbofan rotor blade optimization

Jakub Kmec[1] and Sebastian Schmitt[2]

[1] Palacký University in Olomouc,
Olomouc, Czech Republic
jakub.kmec01@upol.cz
[2] Honda Research Institute Europe GmbH,
Offenbach, Germany
sebastian.schmitt@honda-ri.de

Abstract. Aerodynamic shape optimization has established itself as a valuable tool in the engineering design process to achieve highly efficient results. A central aspect for such approaches is the mapping from the design parameters which encode the geometry of the shape to be improved to the quality criteria which describe its performance. The choices to be made in the setup of the optimization process strongly influence this mapping and thus are expected to have a profound influence on the achievable result. In this work we explore the influence of such choices on the effects on the shape optimization of a turbofan rotor blade as it can be realized within an aircraft engine design process. The blade quality is assessed by realistic three dimensional computational fluid dynamics (CFD) simulations. We investigate the outcomes of several optimization runs which differ in various configuration options. We compare the results from the covariance matrix adaptation evolutionary strategy (CMA-ES) with the outcome of a particle swarm optimization (PSO). We also investigate the changes induced by a different initialization of the CMA-ES and by a variation of its population size. A particular focus is put on the variation of the results if we use different number of degrees of freedom for parametrization of the rotor blade geometry. For all such variations, we generally find that the achievable improvement of the blade quality is comparable for most settings and thus rather insensitive to the details of the setup. On the other hand, even supposedly minor changes in the settings, such as using a different random seed for the initialization of the optimizer algorithm, lead to very different shapes. Optimized shapes which show comparable performance usually differ quite strongly in their geometries over the complete blade. Our analyses indicate that the fitness landscape for such a realistic turbofan rotor blade optimization is highly multi-modal with many local optima, where very different shapes show similar performance.

1 Introduction

Aerodynamic shape optimization using computational fluid dynamics (CFD) is widely used for creating efficient designs in engineering applications such as the

automotive or the aerospace domain [1]. Global optimization techniques enable unbiased search for improved shapes and are capable of yielding conceptually new designs. However, they usually require a significant number of CFD simulations for assessing the quality of the proposed designs. This constitutes a substantial drawback for extensive application of global optimization techniques in actual design cycles since realistic CFD simulations are usually rather complex and have a large time and resource demand. Difficulties arise in that context due to the many options available to the designer for setting up such a numerical design optimization. These include the choice of the optimizer algorithm, specific configuration of the parameters for the optimizer algorithm, representation of the design and design changes to be used, and many more.

In practice, one usually employs a specific setup with some more or less well motivated choices. Then optimization is performed and depending on the results some refinements are done and a new optimization run is performed. This is iterated until a satisfactory result is obtained. It is not possible to assess whether the results obtained in this manner are representative. In particular, it is not clear if the results could have been much better, if a slightly different configuration was chosen. Or, if a rather different design could have been obtained which still has a similarly good fitness value. The answers to such questions are of course strongly linked to the qualitative behavior of the fitness function.

In this work, we explore the fitness landscape of a realistic turbofan simulation case. We investigate the outcome of different optimization runs in which we vary the configuration settings. We compare the achievable fitness values obtained with a covariance matrix adaptation evolutionary strategy (CMA-ES) [4] to a particle swarm optimization (PSO) [3].

Another key ingredient to shape optimization approaches is the choice how to represent a specific design and in particular how to generate sets of different designs to be evaluated during the optimization. In a typical engineering application, a valid baseline design is readily available and the design process focuses on improving that design. Deformation methods are especially suited and well-established for such situations. A crucial decision to be taken by the engineer is the choice of the deformation method and the details of its realization. In the context of evolutionary aerodynamic shape optimization, shape morphing methods are popular such as free-form deformation (FFD) and radial basis functions (RBF) based deformation methods, or shape-function methods like Hicks-Henne (HH) functions [2].

2 Optimization and simulation setup

2.1 Blade representation

During the optimization, the blade geometry has to be changed automatically based on the parameters proposed by the optimization algorithm. Thus, a blade representation has to be defined. In this work, we use a deformation approach, where we start from a given baseline blade design and encode changes to it via

the optimization parameters. Specifically, a blade is represented as a collection of sections stacked on top of each other as shown in Figure 1 (left). Each section (which can be thought of as a 2D airfoil profile) is defined by a cylindrical cut of the blade geometry where the same number of sample points is used for each section. The 3D geometry is reconstructed from those sections by linearly interpolating between the sections.

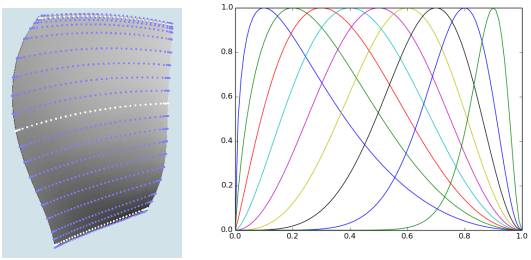


Fig. 1. Left: 3D baseline blade geometry where the points for the sections are shown as blue dots and the three independently deformable sections near the hub, the mid-span and the shroud are shown in white. Right: Hicks-Henne (HH) shape functions.

In the most general situation, all sections would be allowed to change independently to generate a modified blade design. In order to keep the optimization parameters manageable, we select only three sections, one close to the hub, one in the mid-span region, and one close to the shroud, which we deform based on the optimization parameters. The in-between sections are linearly interpolated.

For each of those independently deformed sections, we allow the following changes: (i) rotation of the section around the leading edge point, (ii) movement of the section in the axial-meridional plane, and (iii) deformation of the section profile by adding HH shape functions (see Figure 1 right), which is a well-known representation from 2D airfoil design, see e.g. [5].The explicit functional form of the HH shape functions we use is

$$b(x, x_0) = \left[\sin\left(\pi x^{\frac{\log(0.5)}{\log(x_0)}}\right)\right]^2, \tag{1}$$

where $x \in [0, 1]$ parametrizes the chord length of each section and x_0 controls the location of the maximum of each shape function. The number of shape functions per section $N_{\text{HH/section}}$will be varied in this work and for a given number of $N_{\text{HH/section}}$ we use equally spaced locations for the maxima, i.e. $x_0(i) = \frac{i}{N_{\text{HH/section}}+1}$ where $i = 1, \ldots, N_{\text{HH/section}}$.

In order to fully characterize the deformed blade geometry, we need to specify for each of the three independently deformable sections the amplitude for each shape function, the rotation angle, and the shift in axial and rotational direction. Therefore, the total number of free parameters, which is also the number of search variables for the optimization problem, is given by $N_{\text{search}} = 3(N_{\text{HH/section}} + 3)$.

2.2 Simulation setup

The turbo-fan simulation setup was inspired by the GE Honda HF120 [6] small business jet engine operated at cruise condition. We used the compressible flow solver `steadyCompressibleMRFFoam` from the OpenFOAM CFD suite (specifically, the `foam-extend-3.2` version) which was adapted to be more robust [7]. The blade rotation speed was set to a fixed value of 15600rpm and constant mass-flow and radial equilibrium boundary conditions were employed. We used the $k\omega$SST turbulence model. The meshes for all simulations consisted of about 6 million cells and were created by an in-house software based on the OpenFOAM utility `blockMesh`. The flow equations were solved on 32 CPU cores in parallel, iterated for 12000 solver iterations in order to arrive at a converged solution, and the run-time was typically in the range of 2-4 hours.

The flow-domain of the simulation setup and an example output flow field (pressure) at a circumferential section at a fixed radius is shown in Figure 2. In order to stabilize the CFD solutions we did not include a clearance between rotor blade and shroud geometry. This is somewhat unrealistic as it qualitatively changes the tip vortex caused by such a gap and thus overestimates the blade efficiency. But for the current study it seems a valid simplification which should still provide realistic results except for the blade geometry in the tip region.

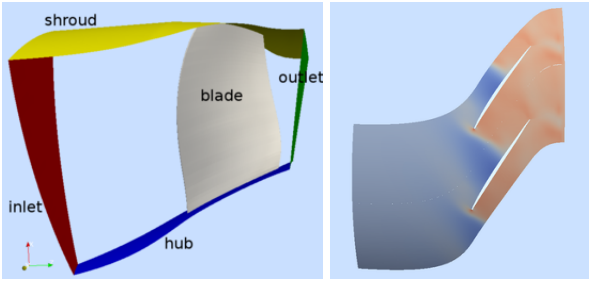


Fig. 2. Left: Schematic side view of the flow passage. Right: Exemplary pressure distribution at a cylindrical cut at 90% span height between hub and shroud. In order to more clearly see the shock forming within the passage, we plot two adjacent passages. Red colors imply high pressure, blue colors imply low pressure.

2.3 Fitness function and optimization algorithms

The optimization maximizes the isentropic efficiency of the rotor which is calculated from the OpenFOAM CFD solutions as

$$\eta = \frac{\left(\frac{P_{\mathrm{T,outlet}}}{P_{\mathrm{T,inlet}}}\right)^{\frac{\gamma-1}{\gamma}} - 1}{\frac{T_{\mathrm{T,outlet}}}{T_{\mathrm{T,inlet}}} - 1} \tag{2}$$

where P_{T} and T_{T} are the mass-flow averaged total pressure and the total temperature at the specified location and $\gamma = 1.4$ is the heat capacity ratio.

The fitness function which is minimized during the optimization is given by

$$f = 1 - \eta + P \tag{3}$$

where the efficiency η is averaged over the last 1000 flow solver iterations and P represents penalty terms which increase (i.e. worsens) the fitness in case the CFD solution does not converge or the generated blade geometry is not feasible.

We employ two different optimization algorithms, the particle-swarm optimization (PSO) [3] and the covariance-matrix adaptation strategies (CMA-ES) [4]. Unless otherwise stated, for the CMA-ES we always use a populations size of $\lambda = 12$ with $\mu = 4$ parents and an initial step size of $\sigma = 0.05$ in relative units where the maximal allowed variation is normalized to one (i.e., $\sigma = 0.05$ amounts to 5% initial variation). For the PSO we use 12 particles (which amounts to 12 evaluations per generation) and the parameters $\omega = 0.8$, $\phi_1 = 1.7$, and $\phi_2 = 1.4$. (see Eq.(6) in [3]).

3 Results

3.1 Comparison between CMA-ES and PSO optimization methods

In order to evaluate the impact of the optimization algorithm, we compared the results of the two optimization methods, specifically a $(\mu = 4, \lambda = 12)$ CMA-ES and a PSO with 12 particles. For the blade deformation, we chose $N_{\mathrm{HH/section}}$=9 which lead to a total of 36 parameters for the optimizer algorithms.

Figure 3 shows the progress of the optimization where the efficiency of each of the proposed individual geometries is shown as function of the optimization generations. The efficiency is normalized to the efficiency of the baseline geometry, i.e. $\bar{\eta} = \eta/\eta_{\mathrm{baseline}}$.

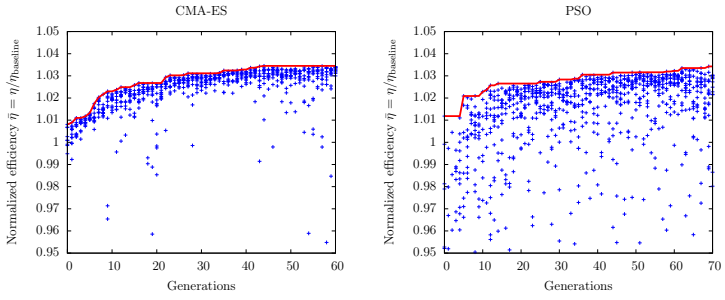


Fig. 3. Normalized efficiency of each evaluated blade geometry as function of optimization generations for CMA-ES (left) and PSO (right). The red marked solutions just indicate the best solution found so far.

As can be seen from the Figure 3, the variation within the population in each generation is much larger for the PSO than for the CMA-ES. This is because we used a rather small initial step size for the CMA-ES, but initialized the particles and velocities for the PSO randomly within the complete search domain. Thus the PSO performs a more global search, whereas the CMA-ES is more local. However, when looking at the progress of the best solution found up to a certain generation, both approaches produce very similar results. Also, the fitness values during later stages of the optimization are approaching very similar results. For example, the best normalized efficiencies were found to be $\bar{\eta}_{\mathrm{PSO}} = 1.0342$ and $\bar{\eta}_{\mathrm{CMA-ES}} = 1.0345$.

Comparing the actual final optimized geometries, we observe that the two optimization runs converged to different geometries. As can be seen in Figure 4 where the three independently deformed sections are shown, all three sections are different among the baseline, the PSO, and the CMA-ES optimized designs. This leads to substantial differences across the entire blade, which are nicely visualized in Figure 5 where the difference between the geometries is colors-coded. The PSO and CMA-ES designs show qualitatively different changes to the baseline. For example, the PSO has a substantially modified trailing edge region of the fan-rotor whereas the CMA-ES design is mostly changed in the mid-chord upper region of the blade.

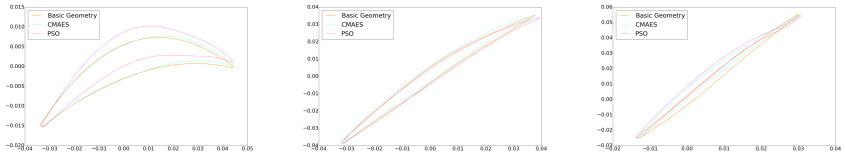


Fig. 4. Comparison of three independently deformed sections of the final optimized designs with the baseline design. The plots are for the hub (left), the mid-span (middle), and the shroud (right) sections.

3.2 Comparison for different random seed initialization

The stochastic optimization algorithms used in this work need an initialization of the pseudo-random number generator which is done by specifying a random seed. The smallest possible change between two optimization runs is thus given just by using different values for the random seed and leaving everything else unchanged.

We used three different values for the random seed to initialize the CMA-ES optimization run with an otherwise identical setup with $N_{\mathrm{HH/section}}$=7 which lead to a total of $N_{\mathrm{search}} = 30$ search parameters. The progress of the best solutions produced by the optimization runs is shown in Figure 6. It can be seen that the initial progress is quite similar but after some initial generations,

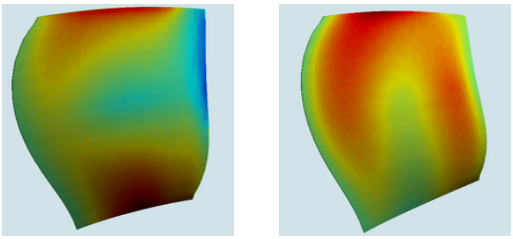


Fig. 5. The difference between the optimized geometries from PSO (left) and CMA-ES (right) to the baseline design where the baseline geometry is shown and the difference is color-coded. Green color indicates similar geometries in that region, red indicates the PSO/CMA-ES geometry is displaced more into the picture away from the viewer and blue indicates the PSO/CMA-ES geometry is displaced toward the viewer as compared to the baseline geometry.

the three runs diverge. During later stages the runs converge again to similar efficiency values, $\bar{\eta}_{\text{seed1}} = 1.0397$, $\bar{\eta}_{\text{seed2}} = 1.0391$, $\bar{\eta}_{\text{seed3}} = 1.0405$.

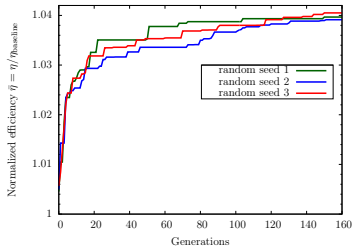


Fig. 6. Comparison of best solutions found during the optimization as function of the generation of three different CMA-ES optimization runs which just differ in the value of the random seed used to initialize the optimization algorithm.

The comparison of the actual best geometries reveals the same insights as in the previous sections as the final geometries differ strongly between the runs (see Figure 7).

3.3 Variation of the number of evaluations per generation

Another design choice for the optimizer setup is how many evaluations per generation are made. Here we compare two CMA-ES runs with $(\mu = 4, \lambda = 12)$ and $(\mu = 4, \lambda = 24)$ and otherwise identical setup. For the blade deformation, we chose two setups with $N_{\text{HH/section}}$=9 and $N_{\text{HH/section}}$=12 which lead to a total of $N_{\text{search}} = 36$ and $N_{\text{search}} = 45$ search parameters, respectively.

The expectation was that the optimization progress would be faster for higher number of evaluations per generation, if measured in terms of generations since

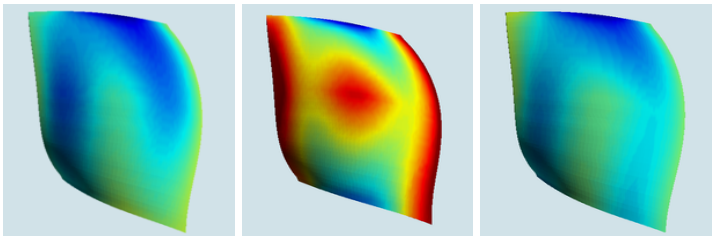


Fig. 7. Color coded differences between the three optimized geometries obtained for three different random seeds 1, 2, and 3 (from left tor right) and the baseline geometry. The color coding of the difference is the same as for Figure 5.

a larger region of the search space could be sampled in each generation and the information gathered could be used more efficiently. In practice, with a usually large resource demand for the CFD simulations, there is trade-off between the parallel resources (or licenses) utilized and the overall run-time. This leads to a limit on the number of evaluations running in parallel. Therefore, the more relevant measure for optimization performance is in terms of actual evaluations, i.e. number of CFD simulations.

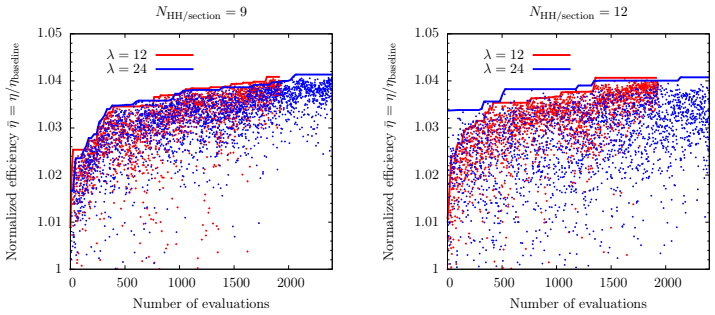


Fig. 8. The efficiency as a function of the number of evaluations for $N_{\mathrm{HH/section}}$=9 (left) and $N_{\mathrm{HH/section}}$=12 (right) for $\lambda = 12$ and $\lambda = 24$.

As can be seen in Figure 8, the optimization progress in terms of actual evaluations was very similar between the $\lambda = 12$ and $\lambda = 24$ runs for both setups. The $\lambda = 24$ runs seem to have slightly larger spread in each generation. It also seems that for larger search dimension, i.e. for $N_{\mathrm{HH/section}}$=12, a larger population size is beneficial in the initial phase of the optimization. Both tendencies are somewhat instructive, but in order to arrive at a really conclusive evidence, single optimization runs as performed in this work are not sufficient. Instead, one would need to run each optimization multiple times and compute statistics

over the results. However, this was not done due to the already high resource demand and long run-times of the CFD simulations.

Comparing the achieved efficiency and the actual optimized blade geometries, the same observations as in the previous sections can be made. Even though the fitness values are comparable, ($N_{\mathrm{HH/section}}$=9: $\bar{\eta}_{\lambda=12} = 1.0409$, $\bar{\eta}_{\lambda=24} = 1.0414$; $N_{\mathrm{HH/section}}$=12: $\bar{\eta}_{\lambda=12} = 1.0407$, $\bar{\eta}_{\lambda=24} = 1.0408$), the resulting geometries differ substantially between the runs over the complete blade for $N_{\mathrm{HH/section}}$=9 whereas the geometries are rather similar for $N_{\mathrm{HH/section}}$=12 (see Figure 9).

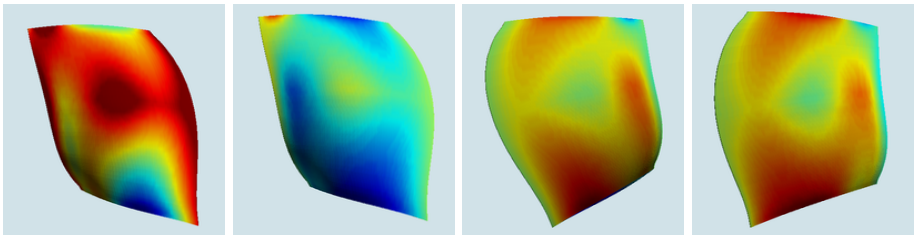


Fig. 9. Color coded differences between the baseline geometry and the optimized geometries for $N_{\mathrm{HH/section}}$=9 and $\lambda = 12$, $N_{\mathrm{HH/section}}$=9 and $\lambda = 24$, $N_{\mathrm{HH/section}}$=12 and $\lambda = 12$, and $N_{\mathrm{HH/section}}$=12 and $\lambda = 24$ (from left to right). The color coding of the difference is the same as for Figure 5.

3.4 Variation of the number of Hicks-Henne shape functions

A very important choice in the course of blade optimization is the specification of the number of HH shape functions per section. It is expected that this parameter has a strong influence on the outcome of the optimization. Choosing a rather small number leads to a very rigid blade representation where only a limited number of designs can be realized which might not include very efficient blades. On the other hand, a very large number might allow to represent a plethora of shapes in principle, including very efficient designs, but the resulting search space has a very high dimension and thus the optimization problem becomes very hard. This usually leads to a much slower progress of the optimization and threatens that a sufficiently improved design will not be found within a reasonable time and the limited resources for the search.

We explored this trade-off by running several optimization runs and varying the number of shape functions ranging from $N_{\mathrm{HH/section}}$=3 up to $N_{\mathrm{HH/section}}$=12. This amounted to search dimensions ranging from $N_{\mathrm{search}} = 18$ to $N_{\mathrm{search}} = 45$, and we used the CMA-ES with a populations size of either $\lambda = 12$ or $\lambda = 24$.

The best results as a function of the optimization generations are shown in Figure 10. First of all it can be observed that there is no clear trend for the achievable efficiency as function of $N_{\mathrm{HH/section}}$. For $\lambda = 12$ the smallest number of shape functions per section $N_{\mathrm{HH/section}}$=3 indeed gives the lowest improvement, but

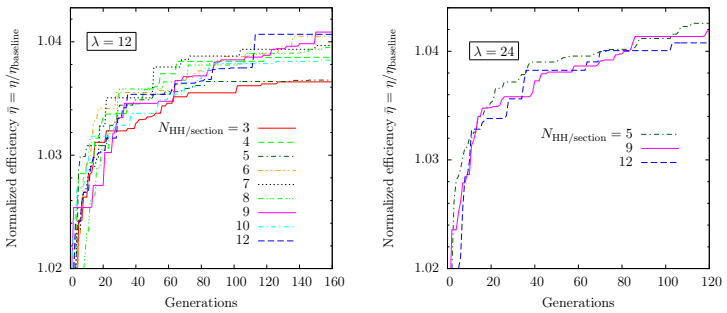


Fig. 10. The comparison of the best solutions found during the optimization for different number of HH functions for $\lambda = 12$ (left) and $\lambda = 24$ (right) evaluations per generations.

for example, for $N_{\mathrm{HH/section}}$=4 a rather large improvement is realized whereas for $N_{\mathrm{HH/section}}$=5 the improvement is as small as for $N_{\mathrm{HH/section}}$=3. In case of $\lambda = 24$, the best efficiency is actually produced with the lowest $N_{\mathrm{HH/section}}$=5.

Figure 10 suggests that the optimization runs are not yet converged after 120 generations but instead keep improving with more generations. This is actually expected for such population sizes and search dimensions. Therefore, it still could be that there is a clear relation between the achievable efficiency and $N_{\mathrm{HH/section}}$, but for all practical purposes, with limited computing resources, this is not relevant. The increased flexibility from a larger number of design parameters is compensated by the increased difficulty of the search problem.

The preliminary insight from this is that already with a rather small number of shape functions per section a considerable improvement in the efficiency can be realized. So the choice of $N_{\mathrm{HH/section}}$is not determined by the expected efficiency increase but rather based on the geometrical aspects implied by the blade representation and the available optimization run time.

We again observe that all the resulting geometries are rather different over large parts of the blade which can be seen in Figure 11. In particular, the shapes are usually much more curvy and wavy during the optimization for larger values of $N_{\mathrm{HH/section}}$, but with increasing generations the shapes seem to be smoothed out again even for larger values of $N_{\mathrm{HH/section}}$(not shown).

4 Conclusion

We studied multiple shape optimization runs for a realistic turbo-fan blade geometry setup. In order to explore the fitness landscape, we investigated many setups, varying the optimization algorithms, the random seed for the initialization, the population size and the number of Hicks-Henne shape functions per independently deformable blade section.

The number of Hicks-Henne shape functions does not seem to have a pronounced influence on the achievable efficiency improvement, at least with the

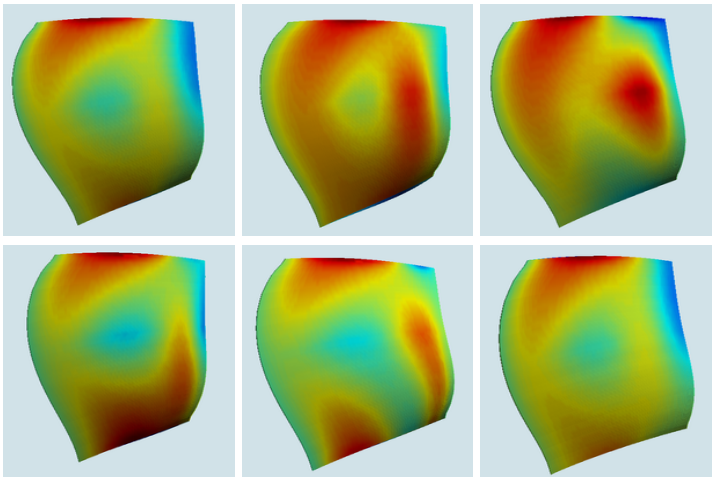


Fig. 11. The comparison of the optimized blade geometries for $N_{\mathrm{HH/section}}$=3 (top left), $N_{\mathrm{HH/section}}$=4 (top middle), $N_{\mathrm{HH/section}}$=5 (top right), $N_{\mathrm{HH/section}}$=7 (bottom left), $N_{\mathrm{HH/section}}$=10 (bottom middle), $N_{\mathrm{HH/section}}$=12 (bottom right) where the difference to the baseline design is color coded and displayed on the blade. The color-coding is the same as in Figure 5.

realistic constraint that only a small number of evaluations (i.e. optimization generations) can be done. This most likely directly reflects the trade-off between the increasing flexibility with a larger number of design parameters and the increasing difficulty of the search problem. For the practical application this implies that a small number of HH shape functions can already be sufficient and geometrical aspects for the location and number of shape functions might be equally or maybe even more important.

The central finding of this work is that for all the tested variants, comparable improvements in the rotor efficiency could be achieved but the actual optimized geometries showed substantial variation over the complete blade geometry. This leads to the conclusion that the fitness landscape of such a realistic turbo-fan optimization is highly multi-modal with many local minima. Even minor changes (e.g. the random seed for the initialization of the optimization algorithm) may lead to very different geometries with comparable efficiencies.

This has strong implications for practical blade design optimization approaches. On the one hand, the designer can easily generate a set of design variations which all have comparable efficiencies by minor changes to the setup. On the other hand, this makes surrogate-assisted optimization approaches [8] more difficult. First of all, training a surrogate model on a highly multi-modal function requires large amounts of data which are most likely not available thereby reducing the model accuracy drastically. Second, data from finished optimization runs will be dense only around one or a few local minima which makes models trained

such data not very helpful for optimizations runs which end up converging to another local minimum.

In the current work we only investigated a single operating condition (fixed mass-flow and rotational speed at cruise condition) and only the rotor passage. A more realistic scenario should include the downstream stator blade which is know to have a strong influence on the fan blade design. Also, additional operating conditions with different mass-flow and rotational speed (e.g. take-off condition) need to be incorporated. The expectation of the authors is that the fitness landscape of such a more realistic setup will still be multi-modal, probably to a lesser extent than for the current case, but the general findings of this work will still be valid. Such investigations are left for future work.

Acknowledgments

The authors thank Hisato Tanaka, Kunio Nakakita, Jiří Šimonek, Radek Máca and Tomáš Krátký for valuable discussion.

References

1. See, for eaxmple, Endicott, G., Olhofer, M., Arima, T., Sonoda, T. (2014). *A Novel Transonic Fan Swept Outlet Guide Vane Using 3D Design Optimization.* Proceedings of the ASME Turbo Expo; Skinner, S., Zare-Behtash, H. (2017). *State-of-the-art in aerodynamic shape optimisation methods.* Applied Soft Computing 62, pp 933-962.
2. Sieger, D., Menzel, S., Botsch, M., (2014) *Constrained Space Deformation for Design Optimization.* Procedia Engineering 82, pp. 114-126; Sonoda, T., Schnell, R., Arima, T., Endicott, G., Nicke, E. (2013) *A Study of a Modern Transonic Fan Rotor in a Low Reynolds Number Regime for a Small Turbofan Engine* Proceedings of ASME Turbo Expo 2013; Kaandorp, M., Menzel, S, Schmitt, S. (2017) *An Aerodynamic Perspective on Shape Deformation Methods.* AIAA AVIATION Forum (AIAA 2017-3145);
3. For a recent review see, Bonyadi, M. R., Michalewicz, Z. (2017). *Particle swarm optimization for single objective continuous space problems: a review.* Evolutionary Computation 25. pp 1-54.
4. Hansen, N. (2006). *The CMA evolution strategy: a comparing review.* Towards a new evolutionary computation. Springer, pp 1769-1776.
5. Hicks, R., Henne, P. (1978). *Wing Design by Numerical Optimization.* Journal of Aircraft 15, pp 407413; Wu, H., Yang, S., and Liu, F. (2003). *Comparison of Three Geometric Representations of Airfoils for Aerodynamic Optimization.* Proceedings of the 16th AIAA Computational Fluid Dynamics Conference: 4095
6. See, for example, Honda R&D Technical Review Vol.26 No.2 (2014).
7. Rusche, H., Schmitt, S. (2016). *Stability Improvements of Pressure-Based Compressible Solver and Validation for Industrial Turbo Machinery Applications.* 4th Annual OpenFOAM User Conference 2016.
8. See, for example, Mengistu, T., Ghaly, W. (2008) *Aerodynamic optimization of turbomachinery blades using evolutionary methods and ANN-based surrogate models.* Optimization and Engineering 9, pp 239-255.