Crawling Along the Pareto Front: Tales From the Practice

Martina Hasenjäger, Bernhard Sendhoff

2005

Preprint:

This is an accepted article published in The 2005 IEEE Congress on Evolutionary Computation (IEEE CEC 2005). The final authenticated version is available online at: https://doi.org/[DOI not available]

In: *The 2005 IEEE Congress on Evolutionary Computation (IEEE CEC 2005)*, pages 174-181. IEEE Press, Piscataway, NJ, 2005.

Crawling Along the Pareto Front: Tales From the Practice

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

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

Abstract- We present a case study of applying multi-objective optimization techniques to the three dimensional design of a turbine blade in a gas turbine that is designed for use in a small business jet. We illustrate the iterative approach to the formulation of the fitness function that is characteristic for such a practical problem and show how the Pareto front accumulated in this process may serve to represent the knowledge on this problem that was collected in the complete optimization process.

1 Introduction

Multi-objective optimization has been the subject of intense research in the recent years in the field of evolutionary computation. The simplest but nevertheless very efficient way to treat problems with multiple criteria is to aggregate the objectives linearly weighted by an appropriate choice of parameters and to render the problem single-objective. As long as the choice of the parameters is well founded, this approach is indeed very sensible since it is the computationally most efficient one. At the same time, the information we gain about the problem after the optimization is limited and the result is valid strictly only for the chosen weights. More popular are evolutionary algorithms which target the whole Pareto front, i.e., the set of all solutions for which no solutions exist which are better than the Pareto solutions in all objectives. Therefore, the optimization target of the Pareto methods is a set of solutions or even a solution space instead of a single solution. Since evolutionary algorithms inherently operate on sets, i.e., the population of solutions, they seem to be particularly suitable for multi-objective optimization problems. The drawback is that most multi-objective optimization algorithms seem to require more function evaluations than efficient single-objective methods like the derandomized evolution strategies with cumulative step-size adaptation. We carefully chose the statement "seem to require" in the previous sentence, since no statistically sound analysis has been published yet to support this proposition. At the same time, it seems intuitive that the identification of a set of solutions or even a space will on average constitute a more difficult optimization problem than the search for a single solution does. Indeed this intuition is supported by our own empirical findings. The larger number of required function evaluations poses a problem when each function evaluation is computationally demanding, like for aerodynamic design optimization problems [1, 2], other fluid-dynamic problems [3] or even multi-disciplinary problems [4]. In these cases, effort has been undertaken to minimize the needed population size [2] or to employ meta-models [5, 6]. At the same time, most of these problems are multi-objective, even if the additional objective might be hidden either in an inherently aggregated problem formulation or might be represented as "soft constraints". We will come back to the formulation of soft-constraints in later sections, because it can – in some instances – provide a simple way out of the problem of computational demand of multi-objective algorithms.

Many different multi-objective evolutionary algorithms have been suggested which mainly differ with respect to the selection mechanisms they employ [7]. Since we deal with a particularly computationally demanding aerodynamic optimization problem in this paper, we have to employ algorithms which minimize the required number of individual evaluations. We will use both "standard" static aggregated methods and the dynamic weight aggregation method [8, 9] which employs an evolution strategy to represent the set of Pareto optimal solutions in an archive. We will use the visualization of the solutions close to the Pareto curve in quality space to represent the information that we gain about the optimization problem during several optimization runs. We will motivate such a patchwork-style optimization in Section 5 after we introduced the problem and presented the optimization results in Sections 2, 3 and 4.

2 Three Dimensional Turbine Blade Design Optimization

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 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 spiraling around the axis. Our goal is to optimize the design of the turbine stator blades.

2.1 The Turbine Stator Blade

Fig. 2 depicts part of a turbine stator row. The stator is of a special type in our case. It 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

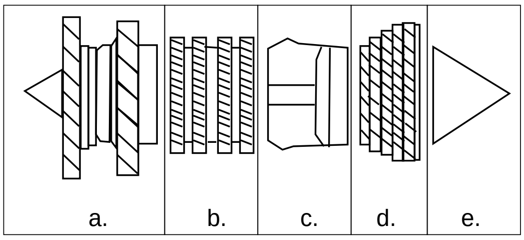


Figure 1: Schematic sketch of a gas turbine. The turbine consists of a fan (a.) that pulls air into the engine. Part of this air is compressed in the compressor (b.) and then forced into the combustion chamber (c.) where it is mixed with fuel and ignited. The resulting hot, high energy gases go into 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.

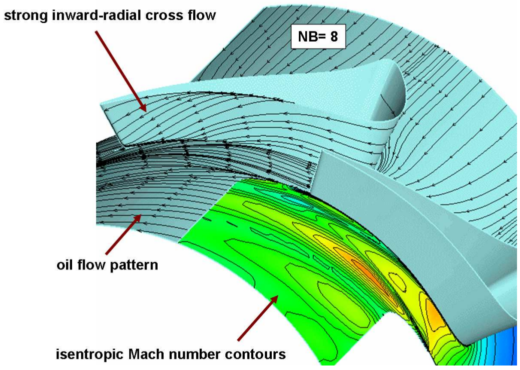


Figure 2: Ultra low aspect ratio turbine stator blades and their flow fields. The blade count is $NB = 8$.

turbine designs. Fig. 2 depicts part of a ULAR stator row. For details on the design specifications of the ULAR stator refer to [10].

Low aspect ratio turbine stator blades have rarely been adopted as components of conventional turbines because of their relatively poor performance. 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 ULAR blade flow characteristics make it unlikely that the advanced design principles developed for conventional 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 direct exploitation of design principles developed for high aspect ratio blades. Hence 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 identify new design concepts that control the three-dimensional nature of the flow.

2.2 The 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

- flexibility: the model must be flexible enough to allow for a wide variety of different designs,
- compactness: the number of parameters describing the model must be low enough to allow for reasonable convergence times of the optimization algorithm, and
- 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 [11] 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. The geometry of the baseline blade is defined by two cross sections, the tip section and the hub 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

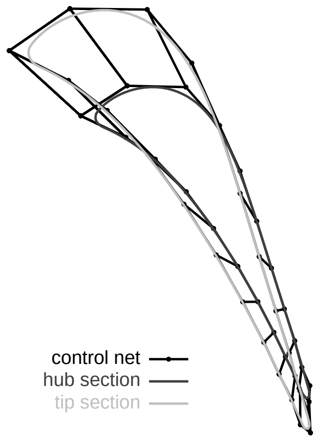


Figure 3: The blade model is created from the hub section (dark gray) and the tip section (light gray) of the baseline blade. These sections are defined by 25 control points each. The control net of the baseline surface model, that connects neighboring control points is depicted by a black line.

> 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.

2.3 Flow Analysis

For the evaluation of the fitness function, an analysis of the aerodynamic properties of the proposed design is necessary. Eventually the design will be built and tested in a wind tunnel. This procedure would be too expensive and time consuming during the design process, so the usual approach is to simulate the flow and thus to estimate the dynamic properties of the blade designs.

For these simulations we used the parallelized 3D Navier-Stokes flow solver HSTAR3D [12]. 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 about 300 generations of the evolutionary optimization, takes about 6 weeks time.

2.4 The Simulation 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.

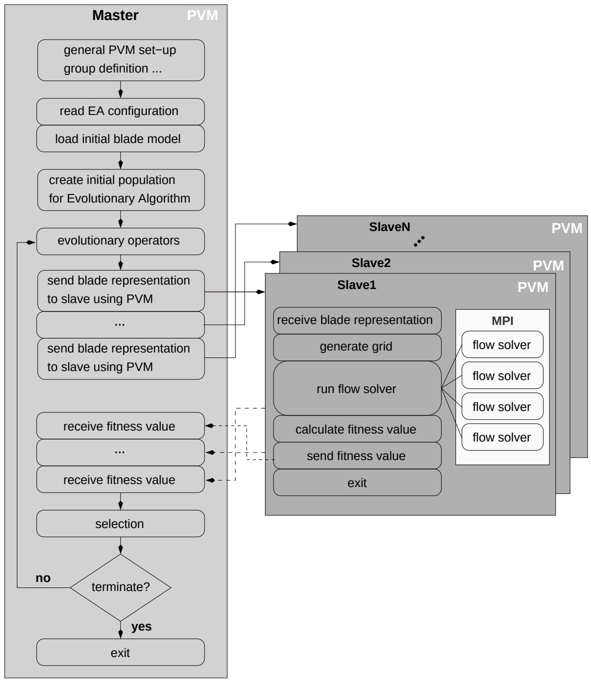


Figure 4: Simulation environment architecture. The program is parallelized at 2 levels: the first level of parallelization is a master-slave model that uses PVM to organize the distribution of single individuals to slave processes while the second level that is started by the slave processes is a node-only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

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 on 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 [13], while the master-slave model of the optimization loop was implemented using the Parallel Virtual Machine (PVM) library [14]. 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.

2.5 Using Evolutionary Algorithms

Evolutionary algorithms [15] are a class of stochastic optimization algorithms whose use in design optimization problems is well established by now [16].

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) [17]. 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.

2.6 Multi-Objective Optimization

A complex optimization problem like aerodynamic design optimization is inherently multi-objective even if often only a single objective is taken into account for optimization. One approach that is often used to avoid more complicated and less efficient multi-objective approaches is to use all but one objective as soft constraints, i.e. small deviations from the target are tolerated. Such an approach is especially appropriate where it is desirable but not mandatory to minimize the objectives used as constraints while an increase of the objective value above a certain value is not tolerable.

Another way to avoid multi-objective optimization is by linear aggregation of the objectives. The drawback of such an approach is that the weights of the aggregated objectives have to be chosen carefully – a difficult task that often needs additional experiments – and that the appropriate weighting of the constraints may change during optimization.

When using real multi-objective approaches in real world problems in which the evaluation of the fitness function is time consuming the most important criterion for the choice of the multi-objective optimization algorithm again is efficiency.

In our approach we used dynamic weight aggregation [8, 9] which searches for the Pareto front by dynamically, gradually, and periodically changing the weighting factors of a linear aggregation of the objectives during optimization. This approach can easily be combined with arbitrary evolutionary strategies so that we can retain using CMA-ES and the associated small populations.

3 Fitness Functions and Constraints

The main performance index in aerodynamic blade design is the aerodynamic loss of the blade which is measured by the mass averaged pressure loss ω. The use of this quantity as performance measure is certainly a coarse but viable simplification of the problem. In practice, a number of other quantities play an important role in the assessment of the blade quality. Unfortunately, it often is difficult to detail the crucial factors and anticipate the influence of a combination of these quantities on the optimization.

However, in the course of the blade optimization project, we identified the variation of the pitch-wise static outlet pressure $\mathrm{PST_{VAR}}$ as a second quantity that should be controlled explicitly in the optimization. From the engineering point of view it is well known that the pressure loss ω and $\mathrm{PST_{VAR}}$ are closely related but to date it is not clear whether it is possible to minimize both quantities at the same time. So the goal in our optimization project is twofold: to find an improved blade design and to collect knowledge about the relation between ω and $\mathrm{PST_{VAR}}$.

With these aims in mind we conducted a series of both single and multi-objective optimizations:

- We only minimized the pressure loss and observed $\mathrm{PST_{VAR}}$.
- We only minimized $\mathrm{PST_{VAR}}$ and observed the pressure loss.
- We used both objectives but rendered $\mathrm{PST_{VAR}}$ as a constraint thus turning the multi-objective problem into a single objective one.
- We used simple linear aggregation of the two objectives. Here initially both objectives received the same weight. This constitutes a simple, naïve approach to multi-objective optimization.
- We used the true multi-objective approach of dynamic weight aggregation [8].

In addition to these two objectives, we used a number of geometrical and manufacturing constraints that concerned the blade thickness, the outflow angle, the solidity, and the mass flow rate. These constraints were fixed during one optimization but were subject to change between different simulation runs based on the analysis of the optimization results.

The baseline blade model that we used to initialize the optimization and that is described in Sec. 2.2 lies within the feasible region of the design space. We formulated the objective function in a way that only violated constraints contributed a penalty. 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.

4 Results

In our experiments, we used a (μ, λ) CMA-ES with parameter settings as described in [17]. We used $\mu = 1$ parent individual and $\lambda = 10$ offspring individuals. The strategy parameters σ were initialized with $\sigma = 0.1$. We ran 5 single- and multi-objective variants of the optimization:

A. Minimization of the pressure loss In this optimization run we used the single objective of minimizing the pressure loss ω. This is the standard approach in aerodynamic blade optimization. In Fig. 5 the results from this run are denoted with A and are marked by squares. In Fig. 5 also the performance of the blade design that was used to initialize the optimizations is plotted as a dotted line for each of the two objectives. This means all blade designs in the upper right quadrant are inferior to the initial blade and all blade designs in the lower left quadrant are superior to the initial blade with respect to both objectives. The designs found by solely minimizing the pressure loss clearly achieve the lowest pressure loss of all designs but are also characterized by the highest static pressure variation.

B. Minimization of the static pressure variation The opposite is true when we minimize only the static pressure variation $\mathrm{PST}_{\mathrm{VAR}}$. This run is denoted by B in Fig. 5.

C. Static linear aggregation of both objectives Although in some cases a reduction of only the static pressure variation irrespective of the pressure loss is a good result, we were rather interested in a reduction of both objectives at the same time. So we used as a first naïve approach a linear aggregation of both objectives. The weights in the aggregation were chosen such that initially both objectives received the same weight. Figs. 6(a)-(d) in which the development of the populations over time is plotted show that this approach, denoted by C, works quite well in the initial phase of the optimization but then it becomes clear that the reduction of the static pressure variation receives too much weight in the optimization: $\mathrm{PST}_{\mathrm{VAR}}$ is reduced but the pressure loss increases even though this increase is not as strong as in the unconstrained case.

D. Minimization of the pressure loss with $\mathrm{PST}_{\mathrm{VAR}}$ as constraint As last single objective approach we minimized the pressure loss and used the second objective as a soft constraint. We constrained the value of the static pressure variation from above, i.e. values that exceeded the value of the initial blade incur a penalty in the fitness function. This approach is quite successful in this case: the pressure loss is decreased and the static pressure variation is controlled, see individuals denoted by D in Fig. 5. The optimization ran for 451 generations in this case.

E. Dynamic weight aggregation Finally, we used with the dynamic weight aggregation a multi-objective approach. Here we varied the weights of the two objectives according to $w_1(t) = |\sin(\frac{2\pi}{100}t)|$ and $w_2(t) = 1 - w_1(t)$, i.e. the weights are periodically varied from 0 to 1 within 50 generations. The results of this method are denoted by E in Fig. 5. We see that this approach is quite successful in minimizing both objectives in an efficient way. The results shown here are taken after 426 generations.

Looking at the overall picture in Fig. 5 and Fig. 6 it becomes nicely clear how the combination of the results from all 5 runs of the optimization helps to develop a more complete picture of the Pareto front for this problem, a picture that cannot be created in this completeness using only one objective function.

5 Patchwork Optimization

The patchwork-like approach to the multi-objective optimization problem that we outlined in the last section might be typical for practical optimization problems for several reasons. Firstly and most obviously, the problem is computationally expensive. Therefore, we have to choose the algorithms that we employ very carefully with respect to the number of function evaluations the algorithm requires until Pareto optimal or sub-optimal solutions are found. It is often not possible to run a high-fidelity but time consuming algorithm to obtain the Pareto front in one step. Instead we have to “grope” toward the Pareto front step by step learning more and more about the problem on the way. In this sense, weight aggregation both static and dynamic seems to be a feasible way to start the optimization project. Secondly, we would argue that many practical problems are inherently dynamic with respect to the quality criterion. From our experience, more often than not, the definition of the optimization problem in itself is subject to a learning process for the application engineers. Real-world application problems are usually tackled by bringing application engineers together with optimization experts. However, we should not assume that the application engineers can formulate the problem characteristics in the same concrete and transparent way as we formulate test functions. Instead the engineers learn more and more about their problem as optimization proceeds. Therefore, driven by some intermediate solutions they will alter the formulation of the problem, the definition of the constraints and add new quality measures or change existing ones. For the optimization expert this process can be difficult depending on the degree of changes. Often the endpoint of such a process is not marked by a final optimal solution of one or even several optimization runs, but by a collection of information about the problem at hand. From this *information pool* the application engineers pick innovative elements and ideas and combine them with own intuition to finally devise the blueprint of the new improved design or process. Of course such an approach to a problem requires an appropriate representation of the information gained during the different optimization runs that have been carried out during the project. In this paper, the two-dimensional quality space, particularly close to the Pareto front, turns out to be such a useful representation. However, different projects might require different representation and research is required into what these might look like.

One might argue that the “patchwork” optimization de-

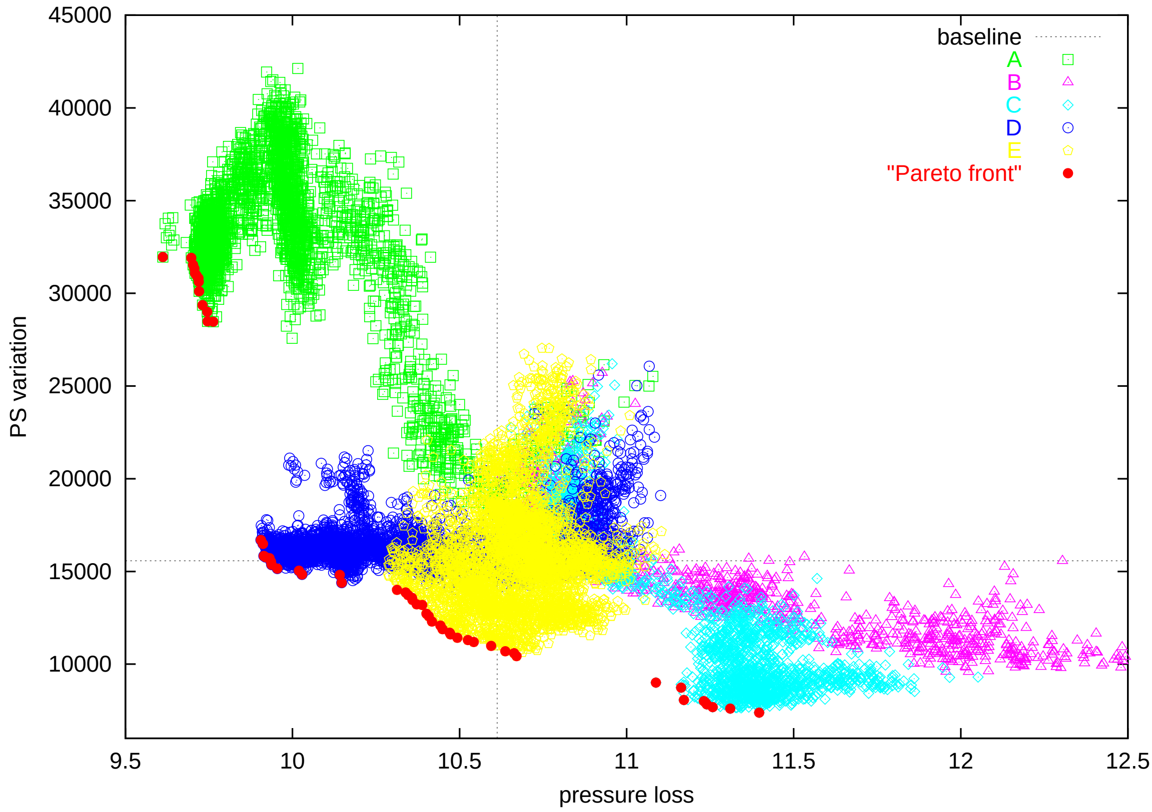


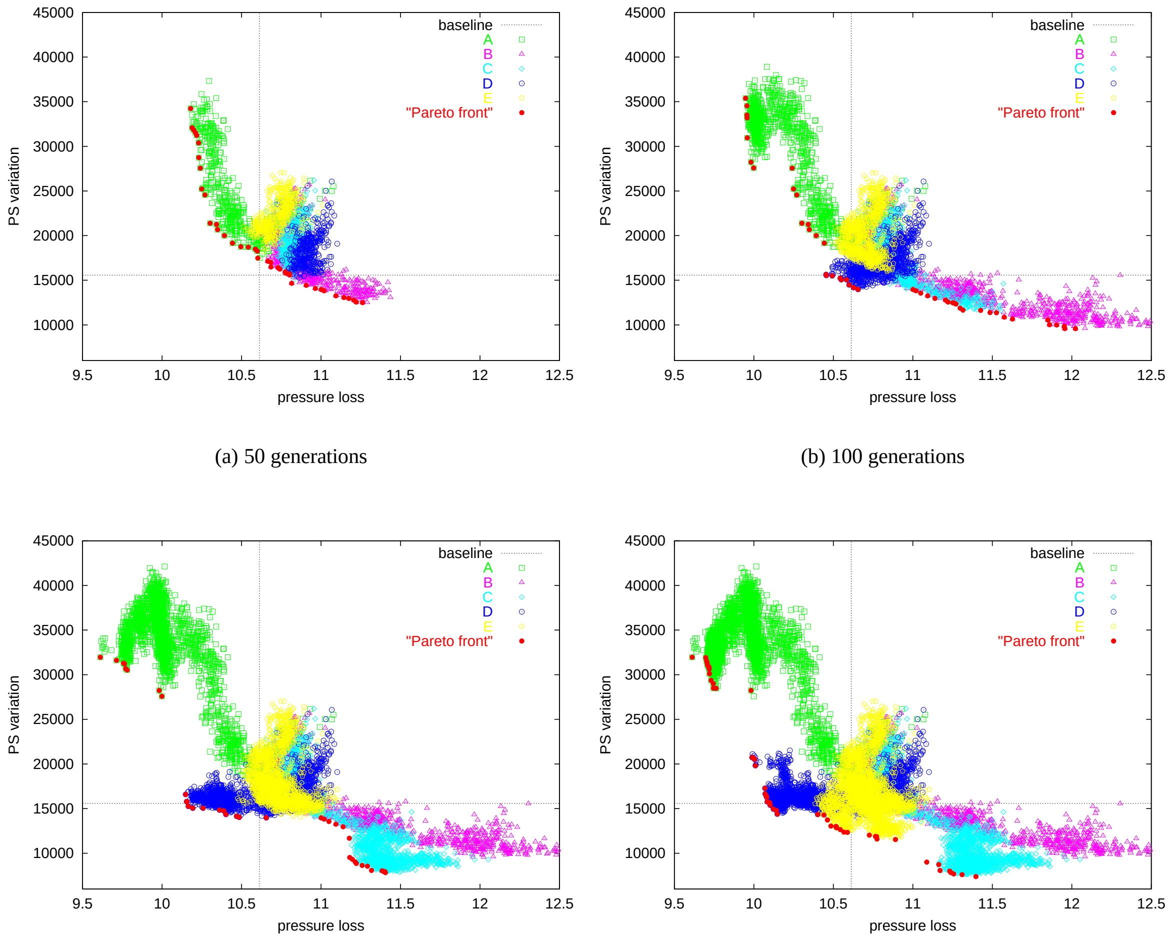
Figure 5: Accumulated Pareto front of the blade optimization problem. Here “baseline” denotes the performance of the baseline blade. A - minimization of the pressure loss, B - minimization of the static outlet pressure variation, C - static linear aggregation of both constraints, D - minimization of the pressure loss with $\mathrm{PST}_{\mathrm{VAR}}$ as constraint, E - multi-objective optimization using dynamic weight aggregation. The individuals that build the Pareto front are marked by filled circles.

scribed above (as opposed to a linear optimization chain, see Figure 7) is neither surprising nor new to optimization engineers. Nevertheless, it is worthwhile to point out that the difference between targeting one optimal design and gathering information about a certain problem is at least as big as between targeting the Pareto front or Pareto space instead of one design. Indeed, by moving our target from one single design point to the space of Pareto optimal solutions is already a big step toward gaining knowledge about the problem at hand. However, we might not want to restrict ourselves to the space of Pareto optimal solutions. One could consider a space made up of different Pareto fronts which belong to different constraints/quality values, where some might be frozen in some runs. Information on the robustness of solutions [18], which constitutes a multi-objective optimization problem in itself [19], might also be worth including in the final description of the problem. One could even consider the representation of poor quality solutions, as long as they represent some design aspect which is non-trivial and which at the same time can be identified to be avoided. Basically, the choice of the representation, in the sense of what can be represented and how easily can it be read out by the application engineer, is the only limiting factor to collecting knowledge about the problem during the optimization process. Therefore, to advance these kind of optimization projects it seems that more research on adequate information representations is required. Such representations are not restricted to quality landscapes, the meta-modeling technique [6] – although usually employed for other reasons – also provides a means to collect information about the problem.

The fast identification of optimal or sub-optimal solutions, which usually is the measure we currently use for the quality of optimization algorithms, might be superficial for this type of optimization problems, because the quality landscape is not sufficiently stable during the lifetime of the optimization project. It seems that evolutionary methods are particularly useful for patchwork optimization because their population based approach already represents the collection and storage of information during search.

6 Conclusions

Aerodynamic design optimization is a particularly suitable application area of multi-objective optimization methods. The use of numerical optimization methods in this subject area requires the explicit formulation of an objective function for the optimization. Traditionally, engineers do not use



(a) 50 generations

(b) 100 generations

(c) 200 generations

(d) 300 generations

Figure 6: The evolution of the Pareto front in the course of the optimization. The results from the different optimization runs are shown after (a) 50, (b) 100, (c) 200), and (d) 400 generations. For an explanation of the nomenclature of the figures, see Fig. 5.

such an explicit formulation of the objective when creating a design but rather depend on a rough formulation of the objective augmented by experience and expert knowledge. The paramount goal in aerodynamic design optimization in general is the reduction of the aerodynamic loss, the pressure loss. At the same time a number of other criteria, whose relevance may emerge only during the optimization process, play a role in the final quality of the design. So generally in a practical application the selection of the objectives to use is an iterative process. In this contribution, we showed that we can and must make use of the experience we gained during this iterative process by recording and utilizing the results collected during the process of optimization.

Bibliography

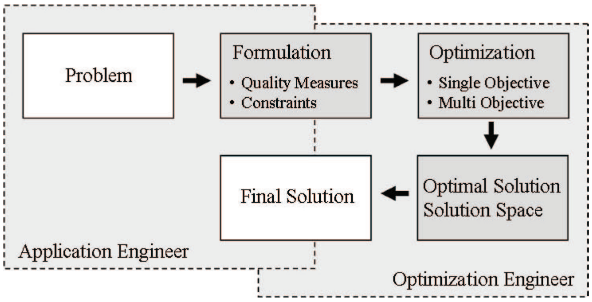
[1] S. Obayashi, Y. Yamaguchi, and T. Nakamura, “Multiobjective genetic algorithm for multidisciplinary design of transonic wing planform,” *Journal of Aircraft*, vol. 34, no. 5, pp. 690–693, 1997.

[2] B. Naujoks, W. Haase, J. Ziegenhirt, and T. Bäck, “Multi-objective airfoil design using single parent population,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1156–1163, Morgan Kaufmann, 2002.

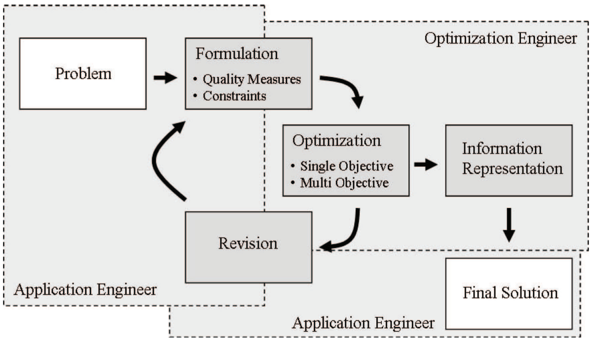
[3] T. Okabe, K. Foli, M. Olhofer, Y. Jin, and B. Sendhoff, “Comparative studies on micro heat exchanger optimization,” in *Congress on Evolutionary Computation (CEC)*, vol. 1, pp. 647–654, IEEE Press, 2003.

[4] K. Chiba, S. Obayashi, K. Nakahashi, and H. Morino, “High-fidelity multidisciplinary design optimization of wing shape for regional jet aircraft,” in *Evolutionary Multi-criterion Optimization*, pp. 621–635, Springer Verlag, 2005.

[5] Z. Z. Zhou, Y. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum, “Combining global and local surrogate models to accelerate evolutionary optimization,” *IEEE Transactions On Systems, Man and Cybernetics - Part C*, 2005.



(a) Standard work-fbw in an optimization project.



(b) Patchwork-style optimization with a circular work-fbw.

Figure 7: In practice, for each optimization project the chain of the standard work-flow sketched in (a) is executed more than once, however, usually information is not collected. Patchwork-style optimization (b) is characterized by a circular work-flow: information is collected and stored during the whole optimization project and used to improve the overall picture about the problem.

[6] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 481–494, 2002.

[7] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, 2001.

[8] Y. Jin, M. Olhofer, and B. Sendhoff, "Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?," in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO*, pp. 1042–1049, 2001.

[9] Y. Jin, T. Okabe, and B. Sendhoff, "Solving three objective optimization problems using evolutionary dynamic weighted aggregation: Results and analysis," in *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO*, Lecture Notes in Computer Science, pp. 636–637, Springer, 2003.

[10] N. Kuno and T. Sonoda, "Flow characteristics in a transonic ultra-low-aspect-ratio axial turbine vane," *Journal of Propulsion and Power*, vol. 20, no. 4, pp. 596–603, 2004.

[11] G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design*. San Diego: Academic Press, 4 ed., 1997.

[12] 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*, vol. 121, no. 1, pp. 44–58, 1999.

[13] "MPI: A message-passing interface standard, http://www-unix.mcs.anl.gov/mpi/."

[14] 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.

[15] T. Bäck, D. B. Fogel, and T. Michalewicz, eds., *Evolutionary Computation 1: Basic Algorithms and Operators*. Institute of Physics, 2000.

[16] A. Osyczka, *Evolutionary Algorithms For Single And Multicriteria Design Optmization*. Physica-Verlag, 2002.

[17] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[18] B. Sendhoff, H.-G. Beyer, and M. Olhofer, "The influence of stochastic quality functions on evolutionary search," in *Recent Advances in Simulated Evolution and Learning* (K. Tan, M. Lim, X. Yao, and L. Wang, eds.), Advances in Natural Computation, pp. 152–172, World Scientific, 2004.

[19] Y. Jin and B. Sendhoff, "Trade-off between optimality and robustness: An evolutionary multiobjective approach," in *Evolutionary Multi-Criterion Optimization* (C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, eds.), pp. 237–251, Springer Verlag, 2003.