MULTI-POINT AIRFOIL OPTIMIZATION USING EVOLUTION STRATEGIES

Boris Naujoks*, Lars Willmes*, Werner Haase†, Thomas Bäck*, and Martin Schütz*

*Center for Applied Systems Analysis (CASA), Informatik Centrum Dortmund (ICD)
Joseph-von-Frauenhofer Straße 20, 44227 Dortmund, Germany
Email: {baeck,naujoks,schuetz,lwillmes}@icd.de

†DaimlerChrysler Aerospace AG Dept. MT63 / Build. 70.N,
P.O. Box 80 11 60, 81663 Munich, Germany
Email: werner.haase@m.dasa.de

Key words: Airfoil design, Optimization, Evolutionary Algorithms, Multi-Criteria Decision Making

Abstract. The considered multi-point (multi-objective) optimization problem is characterized by a multi-modal, nonlinear topology and a highly sophisticated evaluation of the objective function, thus requiring an efficient, direct global optimization algorithm.

Evolution strategies have shown their capabilities for solving complex optimization problems with continuous variables in a variety of applications. This paper reports on the application of evolution strategies to an airfoil optimization problem. The objective function which holds for this test case is described by the difference in pressure described for the different design points of an airfoil compared to predefined airfoil shapes and the corresponding pressure distribution. The design conditions for the two-point optimization problem involve a typical subsonic high-lift and a typical transonic low-drag airfoil.

In particular, results for different algorithmic variants are presented. Some emphasis is put on the reduction of the number of function evaluations required for the algorithm. The results will include an assessment according to the influence of different geometry-parameterization strategies and the dependence of CFD mesh fineness. The CFD analysis utilized a full Reynolds-averaged Navier-Stokes approach in order to achieve an accurate prediction of the different flow types involved.
1 INTRODUCTION

The field of computational fluid dynamics (CFD) provides a wide range of hard optimization tasks. Several optimization techniques have been tried to achieve reasonable results here. The application of evolutionary algorithms promises improved results in comparison to the ones from standard techniques, e.g. hill climbers and gradient based methods.

Evolution strategies are one of the main paradigms in the field of evolutionary computation (EC), focusing on algorithms for adaptation and optimization which are gleaned from the model of organic evolution. These strategies are compared to specialized genetic algorithms and upgraded with some relatively new ideas from adaptation control. All this is explained in the first section in detail.

In the second section the application from the CFD field is presented. The airfoil design problem is not only a hard optimization task in the common sense but also a two criteria problem, which means that there are two ways to receive a fitness value for a given set of object variables. Therefore the results in the third section are divided in results for the multi-point design problem and the single-point design problem. The single-point design problem is just one of the two cases from the multi-point design problem but enables the optimizer to better fit his algorithm to the given problem.

2 EVOLUTIONARY ALGORITHMS

2.1 Evolution Strategies

Like all variants of evolutionary algorithms, the history of evolution strategies dates back into the early sixties (see e.g. [19]). The very first experiments at that time were based on a mutation-selection scheme using one parent and one offspring only, with applications to experimental optimization problems.

For the modern variants of evolution strategies, the concepts of a population of search points modified by normally distributed mutations and various recombination operators together with a selection operator are exploited for continuous parameter optimization problems of the form $f : M \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$. The most important distinguishing feature of evolution strategies consists in their capability to evolve the important strategy parameters – in particular the mutation variances $\sigma_i^2$ – online during the optimization, i.e., to self-adapt these parameters. Here, we give only a very brief outline of these algorithms and refer the reader to more detailed information available in the literature (see [1, 20] for thorough overviews of evolution strategies or [2] for a detailed coverage of all aspects of evolutionary computation).

The so-called multimembered (i.e., population-based) variants ($\mu$+$\lambda$)-evolution strategy and ($\mu$,$\lambda$)-evolution strategy incorporate both the idea of a population (and therefore recombination) as well as the idea of self-adaptation of strategy parameters. The notation ($\mu$+$\lambda$)-evolution strategy indicates $\mu$ parents which create $\lambda > \mu$ offspring individuals by means of recombination and mutation. The $\mu$ best individuals out of parents and offspring
are selected to form the next population. For a \((\mu, \lambda)\)-evolution strategy with \(\lambda > \mu\) the \(\mu\) best individuals are selected from the \(\lambda\) offspring only. Although at first glance the possibility of a \((\mu, \lambda)\)-strategy to forget the currently best solution seems silly, it enables the algorithm to escape from local optima, to follow a moving optimum, to deal with noisy objective functions, and to self-adapt strategy parameters effectively (see below).

An individual is now represented as a vector \(a = (x_1, \ldots, x_n, \sigma_1, \ldots, \sigma_n) \in \mathbb{R}^n\), consisting of \(n\) object variables and their corresponding \(n\) standard deviations for individual mutations. For mutation, each \(x_i\) is mutated by adding an individual, \((0, \sigma_i)\)-normally distributed random number. The \(\sigma_i\) themselves are also subject to mutation and recombination, and a complete mutation step \(m(x_1, \ldots, x_n, \sigma_1, \ldots, \sigma_n) = (x'_1, \ldots, x'_n, \sigma'_1, \ldots, \sigma'_n)\) is formalized as follows:

\[
\begin{align*}
\sigma'_i &= \sigma_i \cdot \exp(\tau'N_i(0, 1) + \tau N(0, 1)) \\
x'_i &= x_i + \sigma'_i N_i(0, 1)
\end{align*}
\] (1) (2)

Mutation is performed on the \(\sigma_i\) by multiplication with two log-normally distributed factors, one individual factor, sampled anew for each \(\sigma_i\) (controlled by the parameter \(\tau'\)), and one common factor (controlled by \(\tau\)), sampled once per individual. This way, a scaling of mutations along the coordinate axes can be learned by the algorithm itself, without an exogenous control of the \(\sigma_i\).

Learning of strategy parameters beyond the scaling along coordinate axes can be achieved by allowing the hyperellipsoids of equal probability density for placing an offspring to rotate in \(n\)-dimensional space, i.e. to adjust the orientation according to the local ascent direction. This can be achieved by introducing correlated mutations via incorporating the covariances of the standard deviations into the individuals. Then, up to \(n(n - 1)/2\) additional strategy parameters are added to the genotype, and mutation and recombination are extended w.r.t. the covariances (see e.g. [3]).

Recombination in evolution strategies with \(\mu > 1\) is always done on the whole parent population. The standard mechanisms are discrete and intermediate recombination. In discrete recombination the components of two parents are selected at random from either the first or the second parent to form an offspring individual, while in intermediate recombination the components of an offspring have values somewhere between the corresponding components of the parents. Both operators can also be extended to their global form, where one parent is selected and fixed, and for each component of the offspring a second parent is chosen anew from the population to determine the components’ values.

Due to space limitations, we can only outline the basic ideas of the standard variants of evolution strategies; for more detailed information the reader is referred to the literature. For the experiments reported in this paper, a special variant of the \((1, \lambda)\)-strategy (one parent generates \(\lambda\) offspring individuals by mutation only) has been used, which is described in detail in section 2.3.1.
2.2 Genetic Algorithms

Genetic Algorithms were mainly developed by John Holland at Ann Arbor, Michigan, during the 60ies and early 70ies. The basic theoretical investigations are given by Holland [10], and an implementation and application to parameter optimization was done in the same year by De Jong [11].

Genetic algorithms typically work on binary strings of a fixed length \( l \), i.e. \( I = \{0, 1\}^l \), with an objective function \( f : I \rightarrow \mathbb{R} \). Although this way genetic algorithms seem to be restricted to optimization of pseudoboolean functions, there is a simple way to apply them also to continuous parameter optimization problems of the form \( f' : \times_{i=1}^n[u_i, v_i] \rightarrow \mathbb{R} \). This is possible by using decoding functions \( \Gamma_{l_x}^i : \{0, 1\}^{l_x} \rightarrow [u_i, v_i] \), which map segments of length \( l_x \) to the intervals \([u_i, v_i]\). Typically, such a function is of the form

\[
\Gamma_{l_x}^i(\alpha_1, \ldots, \alpha_{l_x}) = u_i + \left( \frac{v_i - u_i}{2^{l_x} - 1} \right) \left( \sum_{j=1}^{l_x} \alpha_j 2^{j-1} \right)
\]

and the total length of the individual is \( l = n \cdot l_x \), where \( l_x \) determines the accuracy of the decoded value and may be different for the segments of an individual.

When all individuals of a population have been evaluated, the selection operator selects a new population by taking \( \lambda \) probabilistically sampled copies from the old population. For the so-called proportional selection operator, the sampling probabilities \( p_s \) of the individuals are given by their relative fitness, i.e.

\[
p_s(a_i) = \frac{f(a_i)}{\lambda \sum_{j=1}^n f(a_j)}.
\]

This mechanism will obviously fail in case of negative fitness values or minimization tasks. Due to these reasons, proportional selection is typically used in combination with fitness scaling techniques, which map objective function values \( f'(a_i) \) to fitness values \( f(a_i) \). Today, tournament selection is often preferred over proportional selection in many practical applications. In tournament selection with tournament size \( q \) (typically, \( q = 2 \) is used), \( q \) individuals are randomly sampled from the population and the best of the \( q \) survives – after a \( \lambda \)-fold repetition of this process (where \( \lambda \) denotes the population size), the selection process is completed.

For crossover, which exchanges information between individuals, a large number of operators is described in the literature (e.g. see [8]). Here, we will only describe the one point crossover operator, which works on two individuals \( a_1 = (\alpha_1 \ldots \alpha_l) \) and \( a_2 = (\beta_1 \ldots \beta_l) \) by randomly choosing a crossover position \( k \in \{1, \ldots, l - 1\} \) and exchanging information behind that position, resulting in two new individuals \( a'_1 = (\alpha_1 \ldots \alpha_k \beta_{k+1} \ldots \beta_l) \) and \( a'_2 = (\beta_1 \ldots \beta_k \alpha_{k+1} \ldots \alpha_l) \). This operator is applied with a fixed, exogenously given probability \( p_c \) (\( \approx 0.6 - 0.9 \)).
Finally, with a probability \( p_m \approx 0.001 \) the mutation operator is applied to the bits of the individuals. If a bit position is selected for mutation, its value is simply inverted. This is usually seen as a ‘background operator’ which allows to recover from converged bits, i.e. bit positions having the same value throughout the population.

Again, we can only give some very brief overview of genetic algorithms in this paper. For more detailed information, the reader is referred to [8, 12, 13] or the in-depth treatment provided in the Handbook of Evolutionary Computation [2].

Later in this paper a software tool called FRONTIER is mentioned (see 3.2) which contains a GA based optimization procedure. The method is said to be adapted to mathematical testcases especially from the field of computational fluid dynamics (CFD). Therefore it utilizes special variation and selection techniques as well as special heuristics, but the kind of operators and heuristics is not known. On the other hand some parameters of the GA, e.g. the population size, remain scalable. All GA based results presented in this paper are obtained using this FRONTIER tool.

2.3 Non-standard genetic operators

2.3.1 Derandomized mutation step sizes

In contrast to the standard technique discussed so far, the derandomized mutational step size control proposed in [14] accumulates information about the selected individual’s mutation vector \( \vec{z} \) over the course of evolution by adding up the successful mutations. The authors claim that the method enables a reliable adaptation of individual step sizes (i.e., \( n \) different standard deviations \( \sigma_i \)) even in small populations, namely, in \((1,\lambda)\)-strategies with \( \lambda = 10 \) in the experiments reported. The proposed method utilizes a vector \( \vec{z}^g \) of accumulated mutations as well as individual step sizes \( \sigma_i \) and a global step size \( \sigma \) according to [14]:

\[
\begin{align*}
\vec{z}^g &= (1 - c)\vec{z}^{g-1} + c\vec{z}^* \quad , \quad \vec{z}^0 = \vec{0} \\
\sigma' &= \sigma \cdot \left( \exp \left( \frac{\|\vec{z}^g\|}{\sqrt{n} \sqrt{\frac{c}{2-c}}} - 1 + \frac{1}{5n} \right) \right)^\beta \\
\sigma'_i &= \sigma_i \cdot \left( \frac{|z^g_i|}{\sqrt{\frac{c}{2-c}}} + 0.35 \right)^{\beta'} \\
x'_i &= x^*_i + \sigma' \cdot \sigma'_i \cdot N_i(0,1)
\end{align*}
\]

Essentially, equation (5) captures the history of successful mutations by a weighted sum of the mutations selected in preceding generations (i.e., \( \vec{z}^{g-1} \)) and the mutation vector \( \vec{z}^* \) of the selected parent individual (notice that the method applies to \((1,\lambda)\)-strategies, i.e., \( \vec{z}^* \) is the mutation vector of the single best offspring individual produced in generation
The vector \( \vec{z}^g \) is then used to update both a global step size \( \sigma \) and individual step sizes \( \sigma_i \) according to equations (6) and (7).

Equation (8) then denotes the generation of offspring individuals from the single parent (with components \( x^*_i \)) in a way similar to equation (1), but now using \( \sigma' \) and \( \sigma'_i \). Concerning the choice of the new learning rates \( c, \beta \), and \( \beta' \), both theoretical and empirical arguments are given in [14] for the settings \( c = 1/\sqrt{n}, \beta = 1/\sqrt{n}, \beta' = 1/n \).

3 MULTI-POINT AIRFOIL DESIGN OPTIMIZATION

3.1 Description of test case

The two-point airfoil design problem - originally proposed by T. Labrujere from NLR [16] - is defined by the minimization of an objective function being the difference between the computed pressure coefficient at two different design points and pre-defined target pressures.

The objective function reads:

\[
F(\alpha_1,\alpha_2, x(s), y(s)) = \sum_{n=1}^{2} \left[ W_n \int_{0}^{1} (C^n_p(s) - C^n_{p,target}(s))^2 ds \right] (9)
\]

with \( s \) being the airfoil arc-length measured around the airfoil and \( W_n \) the weighting factors. \( C^n_p \) is the pressure coefficient distribution of the current and \( C^n_{p,target} \) the pressure coefficient distribution of the target airfoil, respectively.

Table 1 provides a summary of aerodynamic flow data chosen for the multi-point design.

<table>
<thead>
<tr>
<th>Property</th>
<th>Case</th>
<th>high lift</th>
<th>low drag</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M_\infty )</td>
<td>-</td>
<td>0.20</td>
<td>0.77</td>
</tr>
<tr>
<td>( Re_c )</td>
<td>-</td>
<td>5 \cdot 10^6</td>
<td>10^6</td>
</tr>
<tr>
<td>( X_{transition} )</td>
<td>c</td>
<td>3% / 3%</td>
<td>3% / 3%</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>0</td>
<td>10.8</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 1: Summarized design conditions (c=chord length)

It becomes obvious from Table 1 that this test case is complicated with respect to the objective to receive a profile shape which holds for both transonic and subsonic (high angle of attack) flow conditions. The airfoil incidences are fixed in this investigation in order to support a more comprehensive validation, although it is possible to use \( \alpha_1 \) and \( \alpha_2 \) as additional design parameters. When taking the incidences as additional design parameters the desired optimum is altered because the targets (with non-fixed \( \alpha_n \)) are changed.
In order to check carefully the accuracy of the optimization process in general, one or two initial test cases can be run prior to the multi-point case, e.g. the pressure re-design case of either the subsonic or transonic flow.

In the European Commission’s CFD project, ECARP [16], a hill-climber was used to compute the described multi-point test case with weighting factors of \( W_1 = W_2 = 0.5 \) on the basis of a viscous-inviscid-interaction approach. When a multi-objective evolutionary algorithm is applied, two objective functions can be used, thus splitting equation (9) in order to obtain a complete Pareto curve for the subsonic and transonic objective function and not a single-point optimum.

Surface data for both airfoils, the transonic low-drag and subsonic high-lift airfoil, are prescribed. All data sets needed for validation purposes can be provided by W.Haase from DaimlerChrysler Aerospace\(^1\) or directly accessed from the INGENET Web site\(^2\). These data sets include pressure data that have been obtained from a fine (512 × 128) underlying mesh, thus providing a better surface accuracy due to the use of a component-Spline interpolation. These data are accompanied by target pressure data computed using a Navier-Stokes approach.

In general, all relevant data presented in this paper, accompanied by additional results on finer meshes and different parameterization, can be taken from the above mentioned INGENET Web site.

### 3.2 Test-case environment

It was mentioned already that the multi-point design case uses a sophisticated CFD method as an analysis tool, in particular a 2D Navier-Stokes approach. No particular attention is payed to the problem of flow-physics modeling. The turbulence models applied for the present cases are the Johnson-King model for subsonic flow (a model with clear capabilities to predict pressure induced separation) and the Johnson Coakley model for transonic flow, respectively. These models have been chosen because of their predictive accuracies to accurately describe all non-equilibrium flow physics phenomena in the desired flow regimes, compare [9].

One of the major results of the European project FRONTIER [21] is an optimization software which can be applied to multi-disciplinary, multi-objective optimization. It employs a genetic algorithm, a hill climber and a so-called MCDM (Multi- Criteria-Decision-Making) tool for treating multi-objective optimization cases. Some remarks on the variant of the GA contained in the software tool are made above (see 2.2). A graphical user interface enables an easy access to the optimizer in general and allows even for novice users to handle complex optimizations problems. All results obtained for the multi-point airfoil optimization are utilizing this FRONTIER [21] software technology as well as the results with the GA based methods for the single-point case. All these results

---

\(^1\)werner.haase@m.dasa.de  
\(^2\)http://www.inria.fr/sinus/ingenet/
have been received within 16 generations and a population size of 64 individuals. (It should be mentioned here that the FRONTIER software (Release 1.0), being the result of the mentioned FRONTIER project [21], is now available as a commercial product; for further information please contact W. Haase at DaimlerChrysler Aerospace3)

The parameterization [22] used for the airfoil investigations is based on Bezier splines for the upper and lower airfoil surface, respectively. In total, 6 or 12 Bezier “weighting points” have been chosen as design parameters, their influence on the optimization results for the airfoil pressure-reconstruction case are going to be discussed below. A parameterization obviously leading to good airfoil representations as it was used by C. Poloni [17] has not been adopted for this work, but will be likely considered for future comparisons. Instead it was suggested to use the parameterization that was used in the ECARP [16] project in order to allow for better comparison with the gradient based method.

All computations are carried out using a structured grid with $128 \times 32$ mesh volumes which - very likely - denotes the coarsest grid (from an engineering point of view) that can be taken for airfoil flow investigations. Further results are currently carried out with $256 \times 64$ mesh volumes which will provide first answers on mesh dependence aspects and how this relates to the optimization process.

In order to gain insight in the complete parametrisation and optimization strategy that is chosen, it turns out to be valuable to first carry out a pressure re-design case which gives rise to test the accuracy of both the parameterization and the analysis tool (the Navier-Stokes method in our case). On a single-objective case, parametrization tests were run by [22], using a NACA0012 airfoil as a starting airfoil and a NACA4412 as the target airfoil. Very good pressure re-construction have been obtained. This was the reason to retain this parametrization for all cases in the present paper.

Later results using evolution strategies did not use any starting airfoil in contrast to the GA based methods in the FRONTIER tool. I.e. randomly chosen starting points have been used, however often resulting in random airfoil shapes (see Fig. 7). This treatment is of course a challenge for the mesh generation tool which should have good smoothing capabilities to overcome the problem of negative volumes.

4 RESULTS

4.1 Multi-objective case

As mentioned above, the multi-point airfoil optimization denotes a rather complex test case as it does not aim at one specific point-design but - in the present case - is aiming at a compromise between two completely different flow situations. Hence it cannot be expected to achieve a “real” minimum of the objective function, i.e. the objective function cannot be driven down to very small values.

On the basis of this multi-point test case, it becomes obvious that evolution strategies provide efficient means to receive a complete set of results (Pareto frontier) for airfoil

3werner.haase@m.dasa.de
shapes ranging from purely transonic airfoil shapes up to purely high-lift airfoils. An experienced engineer may then choose one solution which is near to the transonic flow - as this can be the most important flight regime in a certain flight envelope - or try to get airfoil shapes which in addition provide improved performance in the high-lift domain. Particular constraints (e.g. moment forces) can be treated additionally, however, have not been treated in the context of this paper.

Fig. 1 contains all shape information necessary to understand the complete trial. At first, the two target airfoils are given, exhibiting the different shapes needed for a good performance in transonic and subsonic flow. As mentioned before, the use of only three Bezier points on the lower and upper surface might not be the ultimate choice but it provides ”non-oscillating” shapes. However, it turned out that the leading edge part of the airfoil is only weakly represented.

![Figure 1: Airfoil shapes for multi-point airfoil optimization](image)

Comparing the starting airfoil shape and the resulting shape provided by the optimization, see Fig. 1 it becomes obvious that the starting airfoil shape is quite different from this ”best solution”. Therefore, it is expected that the GA provided a global maximum. On the other hand, a fine-tuning with reduced design parameter ranges in order to reach the absolute extremum might be worthwhile to treat.

Fig. 1 exhibits an ”optimal” shape which combines the geometric (and physical) properties of both the subsonic and the transonic target airfoils.
Figure 2: Design space for multi-point airfoil optimization with 12 Bezier points on the left and 6 Bezier points on the right.
The computed (complete) Pareto curves for 6 and 12 Bezier parameters can be taken from Fig. 2, the arrows are pointing at the "best solution", for the 6-Bezier case this design point was provided in Fig. 1. It becomes clear that the design space for the 12 Bezier design parameters is much wider compared to the 6-Bezier case. Without any doubts this is the drawback of the latter case, not providing the complete range, i.e. the complete Pareto frontier needed for engineering trials. However, the so-called "best" value which is in that very case reasonable compromise between the transonic and the subsonic airfoil is similar in both cases. Clearly, no design will be detectable beyond the Pareto curve in this test case.

4.2 Single-objective case

4.2.1 Results from GA based methods

It was mentioned already that the chosen parametrization, although providing good results for similar airfoil shapes, led to problems in the leading edge part of the airfoil. In future work, it is intended to use more than 12 Bezier design parameters and/or to allow those points located in areas of high geometrical curvature to move also in x- or chordwise direction. This was also indicated by Giotis and Giannakoglou in the context of the INGENET thematic network initiative [7].

For the pressure re-design case, differences on the obtainable geometries and pressures by using 6 and 12 (design) points can be taken from Fig. 3. While the right-hand-side diagrams in Fig. 3 exhibit the "best-solution" pressure distributions compared to the target, it can be easily recognized that there is room for "some" improvement concerning parametrization. In particular the leading edge region is weakly represented and needs enhanced parametrization. Besides these problems with the leading-edge parametrization, it can be seen that the shock location is well resolved by the optimization process, although the shape is different. This can be explained by the objective function itself. In order to obtain "small" values for the objective function, it is a must to keep the shock in the right location, because this is now the driving issue due to the fact that the leading edge part partly failed.

4.2.2 Standard evolution strategies

The focus is now laid on algorithmic variants of evolutionary algorithms. Due to the complexity and the large fitness function evaluation times of the problem it is therefore reduced to the single-objective case. This case enables us to have faster results for different variants on evolutionary algorithms while otherwise being able to compare the results achieved with the ones from the GA variants.

The same parameterization with 12 Bezier points like for the multi-objective case are taken. The flow conditions chosen are: \( Ma = 0.77 \), \( Re = 10 \text{ million} \), angle of attack, \( \alpha = 1^\circ \). Transition from laminar to turbulent flow is set to 3% chord length on lower and
Figure 3: Pressure distributions (right), airfoil shapes, and Bezier parameters (left) for transonic airfoil re-design case
upper surface.

In contrast to the runs with the GA variants, all runs did not start with starting airfoils. The airfoils in the first generation are purely based on stochastic choices of the Bezier point components. A typical airfoil shape and pressure distribution of such a “random” airfoil is shown in Fig. 7. It is obvious that starting with this kind of solutions is a harder task than starting with given airfoil designs because of the worse fitness function values. On the other hand this procedure might give the evolutionary algorithm the opportunity to discover parts of the search space which could not be reached starting with given designs.

Starting with a standard implementation of an evolution strategy different parameter settings have been studied. Different numbers of parents and offspring have been tested with standard evolution strategies using comma and plus selection schemes. Varying the proportion of parents to offspring directly influences the selection pressure of the algorithm and the goal of search. Decreasing the proportion leads to a smoother selection pressure and thus to more exploring the search space. On the other hand a larger proportion leads to a larger selection pressure and more exploiting the search space.

Furthermore 1 and $n$ step sizes for the mutation have been tried to have one global step size or one step size in each direction of the search. Different kinds of recombination have been performed, especially intermediate recombination on the strategy parameters in combination with intermediate or discrete recombination on the object variables.

Some of the results achieved are shown in table 2. Here the $(5+20)$-ES is the one with the best results. This strategy is chosen to investigate the influence of the recombination type on the results.

Fig. 4 and Fig. 5 present 4 different runs of a $(5+20)$-ES with $n$ step sizes. In Fig. 4 the left picture runs with intermediate recombination on both, the objective variables and the strategy parameters, are shown. In Fig. 5 recombination was applied to the object variables in a discrete way and to the set of strategy parameters in an intermediate way.

The pictures show, that the runs with intermediate recombination all converge in an area with similar objective function values. The objective function values of the runs using discrete recombination on the object variables are in a larger range. This range is biased towards better values and the best run with this kind of strategy performs better than the best run with the GA variant on this single-objective case.

4.2.3 Derandomized mutation step sizes

Using derandomized mutation step sizes the results could be further improved. Ten runs with this strategy are shown in Fig. 6. Each was obtained within 1000 function evaluations and starting from randomly chosen initializations.

This best ES variant outperformed the best values known so far in 90% of all tests.
<table>
<thead>
<tr>
<th>μ</th>
<th>strat</th>
<th>λ</th>
<th>Step sizes</th>
<th>max Gen.</th>
<th>max Eval.</th>
<th>best Val.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>,</td>
<td>7</td>
<td>1</td>
<td>118</td>
<td>827</td>
<td>5.14E-02</td>
</tr>
<tr>
<td>5</td>
<td>,</td>
<td>20</td>
<td>1</td>
<td>50</td>
<td>1005</td>
<td>1.08E-02</td>
</tr>
<tr>
<td>5</td>
<td>,</td>
<td>20</td>
<td>n</td>
<td>50</td>
<td>1005</td>
<td>1.07E-02</td>
</tr>
<tr>
<td>1</td>
<td>+</td>
<td>20</td>
<td>n</td>
<td>143</td>
<td>1002</td>
<td>1.04E-02</td>
</tr>
<tr>
<td>11</td>
<td>+</td>
<td>20</td>
<td>n</td>
<td>50</td>
<td>1011</td>
<td>9.95E-03</td>
</tr>
<tr>
<td>4</td>
<td>+</td>
<td>16</td>
<td>n</td>
<td>63</td>
<td>1012</td>
<td>9.15E-03</td>
</tr>
<tr>
<td>13</td>
<td>+</td>
<td>20</td>
<td>n</td>
<td>50</td>
<td>1013</td>
<td>8.74E-03</td>
</tr>
<tr>
<td>5</td>
<td>+</td>
<td>20</td>
<td>1</td>
<td>50</td>
<td>1005</td>
<td>8.89E-03</td>
</tr>
<tr>
<td>15</td>
<td>+</td>
<td>20</td>
<td>n</td>
<td>50</td>
<td>1015</td>
<td>8.49E-03</td>
</tr>
<tr>
<td>7</td>
<td>+</td>
<td>20</td>
<td>n</td>
<td>50</td>
<td>1007</td>
<td>8.45E-03</td>
</tr>
<tr>
<td>3</td>
<td>+</td>
<td>20</td>
<td>n</td>
<td>50</td>
<td>1003</td>
<td>8.35E-03</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>8</td>
<td>n</td>
<td>125</td>
<td>1002</td>
<td>8.20E-03</td>
</tr>
<tr>
<td>9</td>
<td>+</td>
<td>20</td>
<td>n</td>
<td>50</td>
<td>1009</td>
<td>7.26E-03</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>14</td>
<td>n</td>
<td>72</td>
<td>1010</td>
<td>7.21E-03</td>
</tr>
<tr>
<td>3</td>
<td>+</td>
<td>12</td>
<td>n</td>
<td>84</td>
<td>1011</td>
<td>6.30E-03</td>
</tr>
<tr>
<td>5</td>
<td>+</td>
<td>20</td>
<td>n</td>
<td>50</td>
<td>1005</td>
<td>5.84E-03</td>
</tr>
<tr>
<td>25</td>
<td>,</td>
<td>144</td>
<td>n</td>
<td>39</td>
<td>5641</td>
<td>5.64E-03</td>
</tr>
</tbody>
</table>

Table 2: Results achieved using standard evolution strategies

Figure 4: Intermediate recombination on object variables and step sizes
Figure 5: Discrete recombination on object variables and intermediate recombination on step sizes

Figure 6: The derandomized step size control strategy without any recombination
The airfoil shape and pressure distribution of the best result achieved with the derandomized step size control mutation is presented in Fig. 9. In the left part of the figure the airfoil shape is shown and in the right part the pressure can be seen. Fig. 8 shows the airfoil shape (left) and the pressure distribution (right) of the best results from the specialized GA for this single-objective case. As mentioned before Fig. 7 shows some initial conditions generated randomly within the evolution strategies.

A major difference in the airfoil shape between the results from the GA variants and from evolution strategies is hard to recognize. The experts in the field recognize the advantage of the ES variant in the lower surface. This advantage is much clearer visible in the pressure distribution. Here the pressure generated with the ES variant is much more in coincidence with the given target than the one from the GA variant.

Another major difference is the ability to start the optimization process with randomly chosen individuals, i.e. pressure distributions and airfoil shapes. This fact will become very important when leaving the re-design testcase and looking for really new airfoil designs. Here starting with already predefined airfoils will narrow search to a specific search space area and maybe exclude the best solutions.
Figure 8: Airfoil shape (left) and pressure distribution (right) of the best solution found with the GA variant

Figure 9: Airfoil shape (left) and pressure distribution (right) found with the ES variant
5 SUMMARY AND OUTLOOK

The current paper describes the successful application of evolutionary algorithms to an industrial test case. This test case is part of the work carried out in the INGENET project. New and improved results are currently carried out and will be placed on the mentioned data base.

Although it is sometimes mentioned that evolutionary algorithms do not really hold for industrial applications, but provide very reasonable and accurate results for ”non-complex” test cases, the present paper very clearly demonstrates the capabilities of these methods in an illustrative way. Further cooperative actions - academic-industrial - will be carried out in order to learn via cross-fertilization and to get close(r) to industry requirements.

For the current case of multi-point airfoil design, it was demonstrated that evolutionary algorithms are clearly ahead of the use of a single-objective hill climber, even in a single-objective case.

In future investigations on airfoil optimization, mesh dependence and flow-physics-modeling studies as well as different parameterization techniques should be treated in order to gain an enhanced knowledge about such complex optimization processes. The following tasks remain to be performed in the near future for the current test cases using evolution strategies:

• the two dimensional multi-objective case:

    As mentioned before this is to be performed having found good strategies for the single-objective case. As shown before on mathematical test cases (see [4]) evolution strategies are well suited for multi criteria optimization and there seem to be several ways for improving the results in the current airfoil design application.

• involve x-coordinates in design optimization:

    Therefore a change in the parameterization approach seems to be adequate. Involving especially the first two x-coordinates in the objective function would give the opportunity to increase the accuracy of the leading-edge-area pressure distribution. Furthermore, the improved parameterization could lead to faster convergence to non-oscillating geometries even in the 12 design variables case.

The FRONTIER technology will be one of Dasa-M’s technologies for future optimization investigations. A fruitful cooperation with fundamental research will improve this tool and broaden industry’s optimization capabilities. As mentioned above, the new software is available (since April 2000), contact W. Haase at DaimlerChrysler Aerospace4 in case of questions.

4 werner.haase@m.dasa.de
References


