Adaptive encoding for aerodynamic shape optimization using Evolution Strategies

Markus Olhofer, Yaochu Jin, Bernhard Sendhoff

2001

Preprint:

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

Adaptive encoding for aerodynamic shape optimization using Evolution Strategies

Markus Olhofer
markus_olhofer@de.hrdeu.com

Yaochu Jin
yaochu_jin@de.hrdeu.com

Bernhard Sendhoff
bs@el-tec.de

Future Technology Research Division
Honda R&D Europe (Deutschland) GmbH
63073 Offenbach/Main, Germany

Abstract. The evaluation of fluid dynamic properties of various different structures is a computationally very demanding process. This is of particular importance when population based evolutionary algorithms are used for the optimization of aerodynamic structures like wings or turbine blades. Besides choosing algorithms which only need few generations or function evaluations, it is important to reduce the number of object parameters as much as possible. This is usually done by restricting the optimization to certain attributes of the design which are seen as important. By doing so, the freedom for the optimization is restricted to areas of the design space where good solutions are expected. This can be problematic especially if the properties of the design and their interactions are not known sufficiently well like for example for transonic flow conditions. In order to be able to combine the conflicting constraints of a minimal set of parameters and the maximal degree of freedom, we propose an adaptive or growing representation for spline coded structures. In this way, the optimization is started with a simple representation with a minimal description length. The number of describing parameter is adapted during the optimization using a mutation operator working on the structure of the encoding. We compare this method with four different Evolution Strategies using a spline fitting problem as a test function. Of special interest are on the one hand the total number of fitness evaluations, which determine the computational resources necessary for an optimization and on the other hand the final quality of the match measured by the distance between a target curve and the generated spline.

1 Introduction

In recent years, the number of publications describing the application of evolutionary algorithms to aerodynamic design optimisation in particular to turbine blade optimisation has significantly increased, see e.g. [1, 2, 3, 4]. This is mainly due to the fact that the computational fluid dynamics calculations need considerable computing power which only recently became available at a reasonable cost. Evolutionary Algorithms have several advantages compared to gradient based optimisation methods. The estimation of the gradient which is a time consuming and in particularly in the presence of noise difficult exercise is not needed in Evolutionary Algorithms (EA). Due to the population based approach EAs have the ability to escape from local optima, which makes them particularly suitable for design optimisation. Furthermore, they allow multi objective optimisation in a very intuitive way. On the other hand, the demand for computational resources is usually higher for EAs. It has been shown in [5, 6] that a special variant of the Evolution Strategy, the derandomized covariance matrix adaptation, can cope with small populations and still achieve convergence speeds which are as good as or even better than other EAs with larger populations. This strategy was successfully employed for the optimisation of aerodynamic structures in [7].

Another way to increase the convergence speed and therefore, to reduce the necessary computation time is to limit the dimensionality of the search space. This is usually done by using a priori knowledge (gathered from experience with similar designs and from theoretical investigations) to identify the most important design parameters and to restrict the optimisation to these parameters. This tends to lead to very compact representations with a low number of parameters and it has been shown that a high increase of performance for a wide range of designs can be achieved. However, the determination of the correct free parameters is crucial and can be problematic for design problems with unknown fluid dynamical properties where the interaction of the shapes of different areas is not fully understood like for example blade designs for transonic flow conditions.

In this domain, models known from subsonic designs are frequently used and, although good results are achieved, the search space is limited to known features from subsonic blades. The optimisation tends to result in small modifications of known “standard” designs. In order to find designs of principle novelty the degree of freedom must be increased in order to be able to represent designs which not necessarily comply with “engineering intuition”. Unfortunately, this increase in the number of free parameters brings about an increase in the dimensionality of the search space. For Evolution Strategies Hansen et al.[6] estimated the convergence speed on various test functions to scale as $O(n)$ to $O(n^2)$, if n is the dimension of the search space. Therefore, the new parameters have to be selected very carefully. An alternative would be to integrate the increase of the parameter space in the Evolutionary Algorithm as a variation of the representation. In this paper, we will present an approach on how such an integration of a *structure variation* in the Evolution Strategy can be achieved. The optimisation starts on a simple, low dimensional representation of the aerodynamic design, which leads to a fast convergence to a preliminary solution. Subsequently, the chromosome size is increased and the representation becomes more refined and the search space more complex. The change of the representation size is achieved through structure mutations which are designed in such a way as to preserve both, the principle of strong causality [8, 9] and the knowledge gathered in earlier generations.

The remainder of this paper is organised as follows. In the next section, we will introduce different encoding methods for turbine blades for evolutionary optimisation. In Section 3 and 4, the adaptive encoding with structure mutations will be introduced. Since computational fluid dynamics is very time consuming, the simulation results in Section 5 are not generated from an aerodynamic optimisation problem but from fitting a spline to a target structure using a distance measure for closed two dimensional curves. In Section 6 and 7, we will conclude the paper and present a short outlook to future research.

2 Representations of turbine blades for optimisation

In order to achieve a compact parameterisation, which corresponds to a low dimensional search space, a representation [1] of the turbine blade can be based on expert knowledge about the problem and about the possible designs of the blade. An example for a model defined for subsonic turbine blades is given in Figure 1. Important parameters for the flow condition and also for the stability of the design are included in the model. The curves are defined by 3^{rd} order Bézier curves which are fixed by angles and radii at the leading and trailing edge and by a defined radius in the middle section of the blade.

A more detailed description of the curve can be achieved by allowing a larger degree of freedom at the middle part of the blade. Such an extended model is shown in Figure 2 where the top and the bottom curve of the blade are defined by higher order polynomials. Here a more detailed structure of the blade can be represented at the cost of a higher dimensional search space. Such a representation was successfully employed for the optimisation of turbine blades with Evolution Strategies in [7].

An even more flexible model with minimal a priori knowledge besides elementary constraints like closeness, continuity and differentiabilty of the curve is shown in Figure 3. Additionally, the number of control points, which influences the smoothness of the spline and at the same time the degree of details which can be represented, has to be defined. Further a priori knowledge can be integrated to reduce the time for optimisation by defining the initial position of the control points of the spline.

[1]In this paper "representation" and "encoding" will be used synonymously as describing the mapping from the genotype to the phenotype (the 2D-blade cross section)

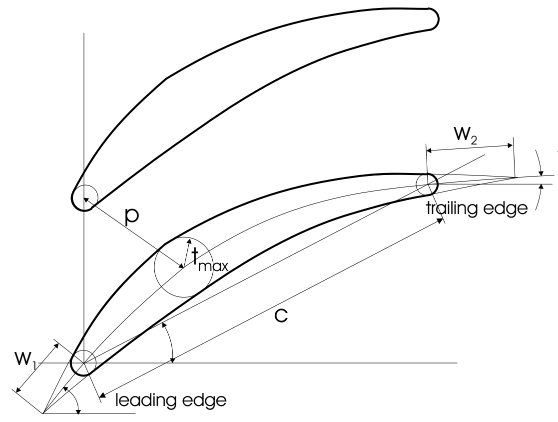

Figure 1: Encoding of a stator blade which is strongly based on a priori knowledge.

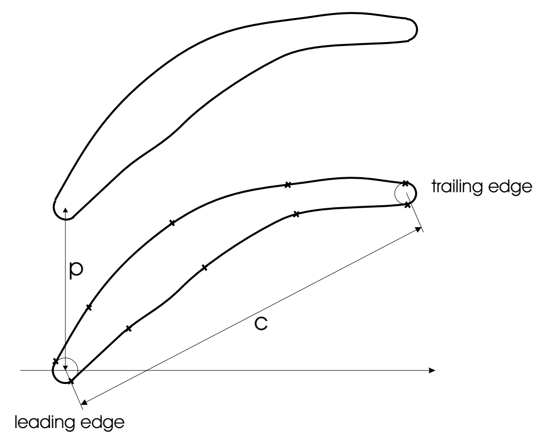

Figure 2: Extended encoding for the suction and pressure side of the airfoil.

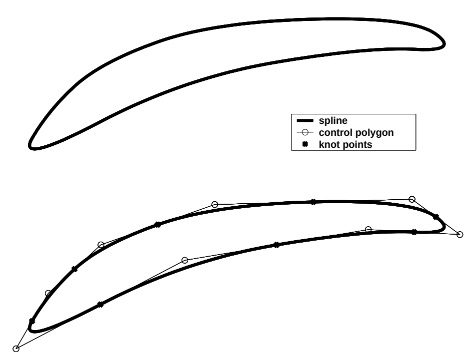

Figure 3: Spline encoding of the stator blade with increased degree of freedom at the cost of a higher dimensional search space.

3 Adaptive encoding of stator blades

In the last section, we introduced different examples[2] for the encoding of turbine blades based on various degrees of a priori knowledge and degrees of freedom. A low number of parameters can be chosen to achieve a fast convergence or a high number of parameters to achieve a high degree of freedom. Since the later will at best lead to an increase of the computational demand and in the worst case to a non-convergent search process, i.e. one that will not be able to reach an optimal solution, simply further increasing the number of parameters, i.e. the number of control points in Figure 3, cannot be the ideal way. Instead, we will propose an adaptive representation in this section which is aimed at combining the advantages of low dimensional search with those of increasing the degrees of freedom when needed. Thus, the idea behind an adaptive representation is as follows: A low parametric encoding can be used in the beginning for the conceptual design. This leads to a fast convergence to regions in the low dimensional search space where good solutions can be found. If the progress of the evolutionary process stagnates due to the limitations in the variability of the encoding, the search space and the complexity of the encoding should be increased by introducing new parameters. In [10] and [11] a similar idea has been formulated to extend the genotype-phenotype map in genetic algorithms and in [12] a process of variable encoding used in the structure optimisation of neural networks where a ontogenetic stage has been simulated. The effect of the increase in the dimensionality of the search space is schematically illustrated in Figure 4. Before the extension of the search space the optimisation takes place in a subspace represented by the two dimensional plane. By introducing a new dimension in the search space the evolution process can explore the higher dimensional search space starting from a near optimal position. The shape of the new fitness landscape depends on the parameter which is introduced. To find an extended search space with a high potential for generating solutions with a higher fitness is a search process by its own. Since it is not clear where to extend the representation, we will realize the adaptation of the encoding as an additional mutation operator. We have two kinds of parameters with two different means of adaptations. Firstly, the object parameters, the x- and y-coordinates of the spline control points like in the static model in Figure 3, whose changes have a direct influence on the shape of the turbine blade and therefore, on the fitness value of the individual. The adaptation of the object parameters has a direct influence of the selection process. Secondly, the structure parameters, which code for the number of object parameters and their position on the spline. As we will see in the following, the adaptation of the structure parameters occurs more indirectly.

The object parameters are real-valued numbers and are

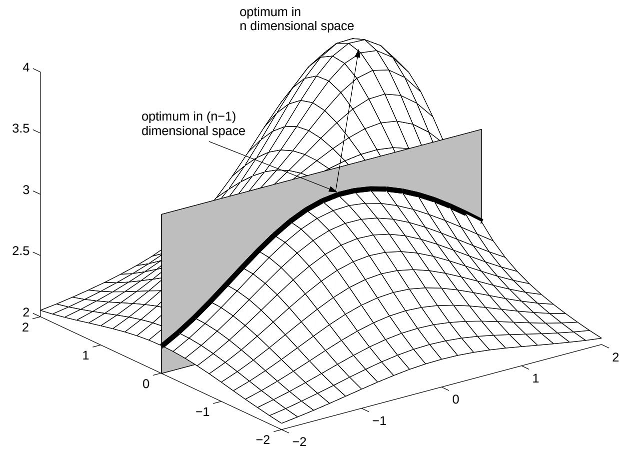

Figure 4: Illustration of the effects of introducing new parameter in the encoding on the fitness landscape.

optimised with an Evolution Strategy. In opposite to this, structure parameters are usually discrete, control points either exist or do not exist. The same applies to other structure optimisation problems like in neural networks where connections or neurons also either exist or do not exist. In these cases, where the smallest possible mutation is to remove or add a control point, the principle of strong causality, which is often regarded as an important property for the success of Evolution Strategies for continuous optimisation problems, is difficult to comply with. As a result a direction of search in the structure parameter space is difficult to identify and most mutations will in general be lethal, i.e. will lead to a design of a turbine blade with a considerably lower performance[3]. A possible way not to disrupt the mapping between the genotype (the vector of control and knot points) and the phenotype (the curve as shown in Figure 3) during a structure mutation is to make it neutral. Thus, the change in the genotype has no direct influence on the phenotype and in turn on the fitness of the individual. This seems to be reasonable because the structure mutation should be regarded as a process which aims at increasing the variability of the encoding, which further on can give rise to a better adaptation of the object parameters towards higher fitness of the individual. The usefulness of neutral mutations has also been demonstrated by [13] for the evolution of digital circuites using GAs. The effect of these kind of mutations cannot be observed from the fitness value directly after the mutation. It can only be measured on a longer time scale, when the possible increase of variability resulted in a fitness increase. A selection process based on the fitness of individuals with a mutated representation directly after the mutation itself is independent from the changes in the representation. In order to measure the effect

[2] All of these have been used for the optimisation of turbine blades with Evolution Strategies with the expected results of increasing the likeliness of new, unsuspected designs at the cost of a substantial increase in computational demand.

[3] Indeed during the complete optimisation with the computational fluid dynamics calculations such mutations are “lethal” in the sense that the flow solver for the Navier-Stokes calculations does not converge for the resulting designs.

of the modification in the encoding we use an evolution in a separated sub-population, where the individual with the mutated structure parameters is evolved. This is done for a small number of generations, until the effect of the structure modification should have been appeared. After the separation the fitness can be compared with that of other sub-populations to evaluate the effect of the modification of the encoding.

In the adaptive encoding not all features of the design have to be defined in the initial representation and the integration of a high degree of expert knowledge and with it the danger of restricting the search space to the domain of common engineering solutions, is not necessary. Of course the success of the scheme proposed in this section relies on two main assumptions. Firstly, a neutral structure mutation can be realised, for splines this is possible (at least for adding control points) as we will see in the next section. Secondly, we assume that the phenotype described with few parameters is a good starting point for the refined design after the structure mutation, i.e. the parameters which have been optimised before the structure mutation, will be a good starting point in the enlarged search space.

3.1 Adaptive encoding with Non Uniform Rational B-splines

In order to use the adaptive encoding for the blade optimisation, the spline encoding of the geometry is well suited. The chromosome of each individual is given by four vectors, the object vector $\vec{i_1}$ with the coordinates (c_x, c_y) of the control points:

$$\vec{i_1} = \left(c_{x_0}, c_{y_0}, \dots, c_{x_{n-1}}, c_{y_{n-1}}\right), \tag{1}$$

a vector containing the position of the knot points along the spline,

$$\vec{i_2} = (u_0, \dots, u_{n-1}), \tag{2}$$

which define the spawns of the spline. Furthermore, a vector consisting of the strategy parameters

$$\vec{i_3} = (\sigma_0, \dots, \sigma_{m-1}) \tag{3}$$

and a vector $\vec{i_4}$ describing the number of control points. The value of m depends on the type of Evolution Strategy which is employed, i.e. $m = 2n$ for the mutative step-size adaptation per individual.

A Non Uniform Rational B-spline (NURBS) curve is defined in parametric form by

$$\mathbf{C}(u) = \frac{\sum_{i=0}^{n-1} N_{i,p}(u) w_i \mathbf{P}_i}{\sum_{i=0}^{n-1} N_{i,p}(u) w_i} \qquad a \le u \le b, \tag{4}$$

where $\mathbf{P_i} = (c_{x_i}, c_{y_i})$ are the n control points encoded in the chromosome, w_i are coefficients weighting the influence of each control point on the corresponding spawn and $N_{i,p}(u)$ are the p^{th} degree B-spline basis functions

$$N_{i,0}(u) = \begin{cases} 1 & : \quad \textit{if} \quad u_i \le u \le u_{i+1} \\ 0 & : \quad \textit{otherwise} \end{cases}, \tag{5}$$

$$\begin{aligned} N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) \\ &+ \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u), \end{aligned} \tag{6}$$

defined on the knot vector

$$U = (u_0, ..., u_{n+p}). \tag{7}$$

In case of $w_i = 1$ for $i = 0 \dots n-1$ a NURBS curve is identical to a B-spline curve, which is used here in order to reduce the number of parameters in the optimisation.

In order to generate close curves the last $(p-1)$ spawns have to be equal to the first $(p-1)$ ones in the spline. This is realised by extending the control point vector at the end by the first p control points and the knot vector by p knot points with the same respective distances as the first p control points.

The description of a curve can be easily extended by adding a control point to $\vec{i_1}$ according to the *knot insertion algorithm*, see e.g. [14].

Therefore, it is possible, at least for the extension of the control polygon, to realize a neutral mutation on the structure variables. The removal of control points is not possible in every case while preserving the shape of the curve. However, an estimation of control point positions which generate a very similar spline curve can be used.

4 Experiments on target shape approximation

4.1 Setup of the experiments

Transonic fluid dynamics computations are too costly for performing tests with evolutionary optimization in a wide range of different parameter sets and for different methods. Therefore, a test function based on the comparison of the current shape with a 2-D target shape is used, i.e. a closed spline curve is fitted to a given target curve. The comparison is based on a modified Hausdorff distance [15] measure between two sets of points ($\mathcal{Z}_1$ and $\mathcal{Z}_2$), one given by the target curve and the other by the current closed spline. The sum of all smallest euclidian distances between the two sets of sample points is calculated and used as a distance measure between the curves:

$$\begin{aligned} f_{dist} &= \frac{1}{2} \left(\sum_{i=1}^{|\mathcal{Z}_1|} \min\{|\vec{a_i} - \vec{b}|^2; \vec{b} \in \mathcal{Z}_2\} \right. \\ &\qquad \left. + \sum_{j=1}^{|\mathcal{Z}_2|} \min\{|\vec{b_j} - \vec{a}|^2; \vec{a} \in \mathcal{Z}_1\} \right). \end{aligned} \tag{8}$$

Additionally, a penalty factor $f_{loop} = 2$ was applied to the fitness value in case the curve intersected itself. Such loops in the 2-D spline correspond to zero thickness in the turbine blade and therefore, do not represent viable solutions. In the following experiments, the set size of sample points is

$$|\mathcal{Z}_1| = |\mathcal{Z}_2| = 200.$$

The resulting fitness function is defined by

$$fit = \begin{cases} f_{dist} \cdot f_{loop} & ; \text{ spline with loops} \\ f_{dist} & ; \text{ spline without loops} \end{cases} . \quad (9)$$

The target curve and the initial setting for an encoding with $p = 8$ control points, which is used for the comparison of different Evolution Strategies, is shown in Figure 9 (a) and for $p = 3$ control points, which was used in the experiments for the dynamic encoding, in Figure 9 (b). All results, i.e. the value of the best individual in each generation, are averaged over 20 runs with different settings for the random number generator and the same initial conditions and also the same target curve.

In Figure 5, a comparison of four different strategy parameter adaptation methods in evolution strategies with a fixed encoding is shown. On the one hand, this is done to compare the strategies on this spline fitting problem, on the other hand to provide a basis for a comparison with the proposed method of an adaptive encoding. We employed the following strategies: the global mutative step size adaptation (GSA), the individual mutative step size adaptation (ISA) and two derandomized strategies: the individual derandomized step size adaptation (IDA) and the covariance matrix adaptation (CMA). All parameters were set to standard values, see [16] and [17] for the mutative strategies and [18] for the derandomized strategies. We did not employ any recombination operator, because it turned out to be too disruptive in most cases. The figures describing the test results show the averaged logarithmic fitness values vs. the number of fitness evaluations, which is for the turbine optimisation problem proportional to the overall computational demand. In all strategies, we used a $(\mu = 2, \lambda = 10)$ population. In particular, for the mutative strategies this seems to be very small. However, experiments have shown that even for these strategies small population sizes are more efficient measured in performance vs. function evaluations. This is likely to be a result of the absence of a recombination operator which would take more advantage of the information contained in the population.
In a second and third experiment the influence of the number of control points is shown. The tested numbers of control points are $n = 3, 4, 5, 10, 15$ and 20. In Figure 6, the results for the global step size adaptation are shown and in Figure 7 for the covariance matrix adaption.

Finally, experiments for the adaptive encoding were carried out and compared to the static encoding. In these experiments, the spline was initialised with three control points. The structure mutation operator was used with a fixed probability of $p_{ins} = 0.5$ in fixed periods of generations: $g = 1, 2, 4, 8, 16$ and 32, according to Figure 10. For example, for $g = 8$, in every 8^{th} generation the structure mutation algorithm was applied and an additional control point was inserted with a probability of p_{ins} to every single individual.

After the mutation of the encoding the new individual was protected in a sub-population for a period of $G = 10$ generations. In the protected sub-populations a $(1, 2)$ strategy was applied. The number of allowed sub-populations was set to $n_s = 5$ in order to assure that the number of fitness evaluations in each generation is identical to the one in other generations. The reason is the parallelisation on λ machines, which are optimally utilized if the number of fitness evaluations are a multiple of the number of machines. Due to the same reason the structure mutation operator was used only every g^{th} generation instead of mutate the structure with a correspondingly smaller value of the mutation probability p_{ins} in every generation. In the selection process after the period of protection, the μ best individuals from the best parents of all sub-populations and from the individuals in the last offspring generation are chosen.

4.2 Results

From the comparison of the different strategy adaptation methods in Figure 5 it can be seen, that the adaptation of one global step size (GSA) shows the best performance during the first few generations. Thus, for the first rough approximation of the target curve one global step size is sufficient. However, for a more refined approximation at later generations individual step sizes are needed, as can be seen from the later stagnation of the GSA method. On the other hand, the individual mutative step size adaptation shows the worst convergence speed, which is probably due to the small population size. The best results for optimisations with more than about 1000 fitness evaluations[4] are observed with the derandomized strategies. An example of a result is shown on Figure 8, where it can be seen that the final curve matches the target curve very well, with the exception of the round edges in the lower part of the curve, where the number of control points is not sufficient. The use of the history of the optimisation instead of using only information from the current generation to adapt the strategy parameter seems to be favourable after a certain number of generations. In addition to the "derandomized" adaptation this leads to a faster convergence towards the optimum (or at least a good sub-optimum). At the same time, the initial advantage of the GSA can be important for cases with long evaluation times and limited overall time for the optimisation.

In Figures 6 and 7, the results of the optimisation for different chromosome lengths are shown for global mutative step size adaptation and covariance matrix adaptation. It can be seen, that the convergence speed and also the quality of the final result heavily depend on the number of control points. In

[4] This corresponds to the order of fitness evaluations, which are feasible for the turbine blade optimisation with the computational fluid dynamics methods.

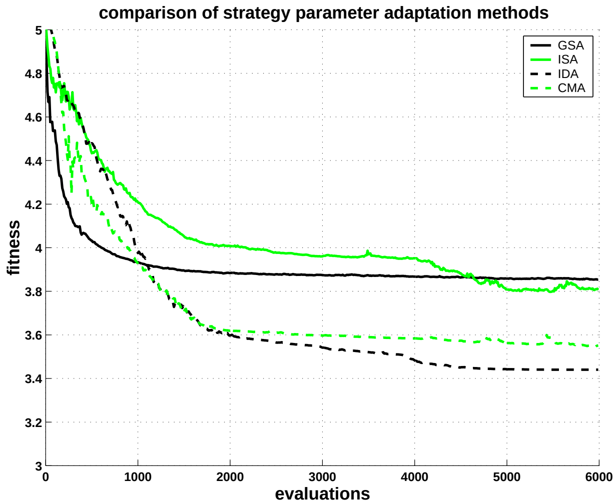

Figure 5: Comparison of different strategy parameter adaptation methods. A static encoding with 8 control points was employed to a ES(2,10)

.

case of a lower number of control points, i.e. less than $n = 8$, the convergence speed is very high, but the encoding does not provide enough flexibility to achieve a good approximation of the target curve with the exception of the results for $n = 10$ for the CMA algorithm. If the number of control points is higher, the convergence is very slow and additionally the algorithms converge to sub-optimal solutions in the high dimensional search space. This can also be seen from the "steps" in the fitness progression, which are due to the penalty term added to the fitness value if the splines contain loops, see Equation (9).

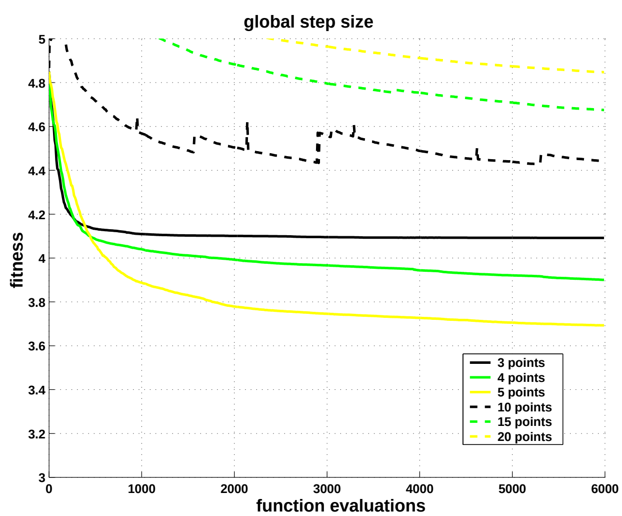

Figure 6: Results of a static encoding for different numbers of control points for the GSA(2,10).

In Figures 10 and 11, results of the proposed method of an adaptive chromosome size are shown. The structure mutation

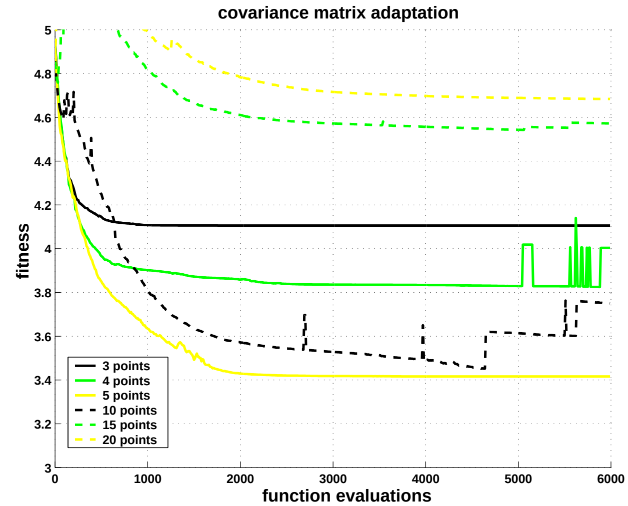

Figure 7: Results of a static encoding for different numbers of control points for the CMA(2,10).

operator was applied every g generations, where g is varied from $g = 1$ to $g = 32$ as discussed in the last section. In the first figure the fitness value and in the second figure the actual number of control points is shown.

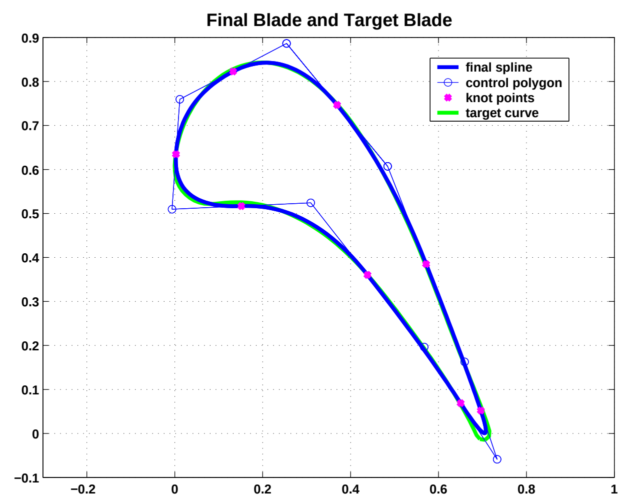

Figure 8: Example of a final spline approximation

The convergence speed for generation gaps between the structure mutations larger then $g = 2$ is comparable to the fast convergence of the GSA during the first 100 evaluations. Thereafter, an increase of the fitness can be observed which is comparable to the derandomized algorithms. The averaged final result after 6000 evaluations (600 generations) is better than the averaged result for all other methods with static encodings. This is even the case for a structure mutation interval of 32 generations, where the final number of control points is $n = 8$, which is equivalent to the experiments with the static encoding. Another interesting observation is, that there is no significant influence of the interval in which the structure mu-

(a)

(b)

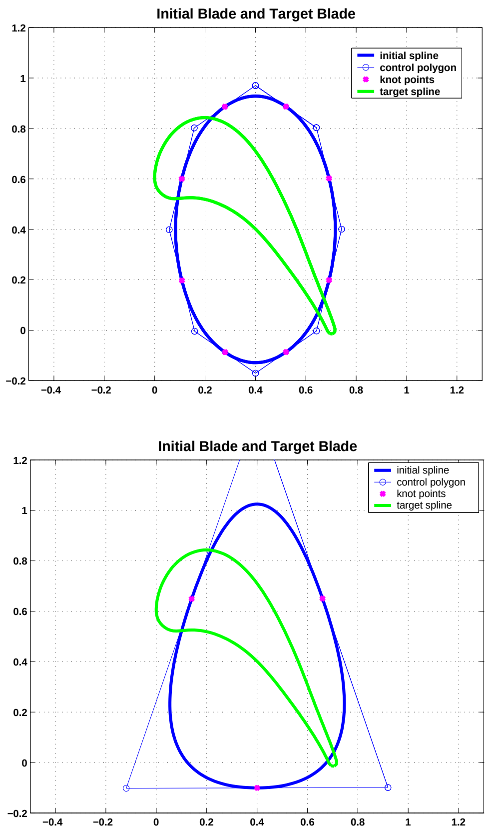

Figure 9: The target design and the initial encodings with 8 control points (a) and three control points (b).

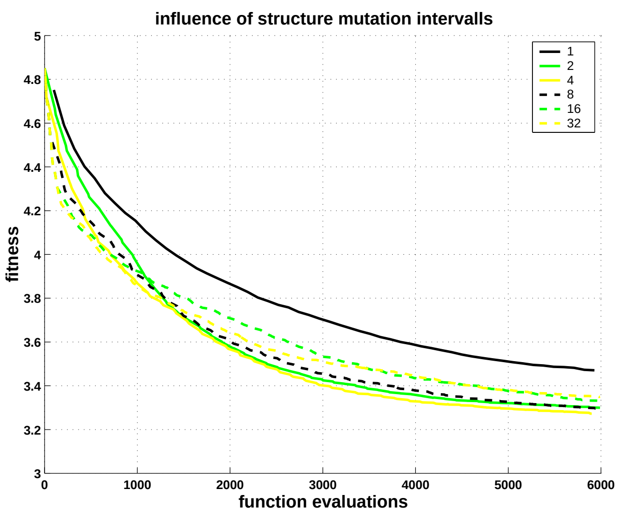

Figure 10: Fitness history for an adaptive encoding with a global mutative step size adaptation for an ES(2,10).

tation is used on the fitness value (with the exception of $g = 1$ for which the representation seems to grow too fast). At the same time the influence on the chromosome length is significant. The size of the chromosome increases, although the fitness is similar. For practical applications, a compact de-

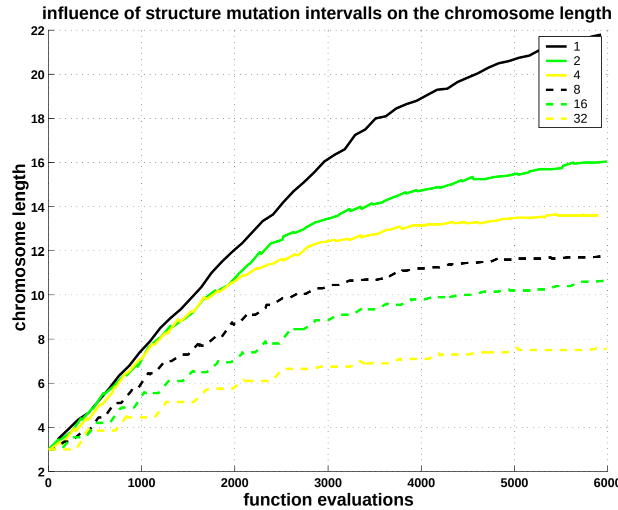

Figure 11: History of the chromosome length measured in control points for an adaptive encoding with a global mutative step size adaptation for an ES(2,10).

scription is desirable in most cases. Therefore, the interval should be sufficiently long.

Furthermore, it can be seen, that there is a strong increase in the encoding length at the beginning when the chromosome size is very small. Later on during the optimisation the increase becomes more gradual and for large generation gaps, e.g. $g = 32$, nearly stagnates. Therefore, a regularization term in the fitness function does not seem to be necessary for guiding the optimization towards solutions with a minimal encoding length.

5 Conclusion

The introduction of an adaptive encoding of the spline parameters resulted in a performance increase compared to static encoding using mutative adaptation as well as derandomized adaptation methods. This performance increase corresponds to both the convergence speed as well as the quality of the final result. Furthermore, a lower degree of a priori knowledge has to be assumed and encoded in the representation and also in the initial condition of the optimization. A priori knowledge can be integrated in the employed spline encoding in two ways. One is to determine the number of parameters of the initial spline and the other is to define the initial parameter values.

In the experiments in this paper, we concentrated on the number of parameters. The setting of the initial population was kept similar in order to reduce its influence. However, it can be shown that a representation with two or three control points can be initialized randomly, because in that case loops in the spline, which are the main source for local optima and a convergence to not valid geometries cannot be represented.

A random initialization of a representation with a higher number of parameters is likely to result in premature convergence to a local optimum even if we use the penalty term for loops in the curve. Therefore, a good initial solution has to be used together with a small step size in order to preserve that encoded knowledge.

In the static methods the distinction between conceptual design and design optimization is made [19]. *Conceptual design* corresponds to an optimization in a large parameter range with a low number of parameters in order to find a rough shape of a design in a large search space and *design optimisation* corresponds to a fine tuning of well defined and fixed parameters. If a dynamic encoding is used both kinds of optimization can be integrated. In the first stage of the optimization the conceptual design with a highly restricted search space is realized. Continuously, the conceptual search is narrowed down to a particular sub-domain, where additional degrees of freedom are introduced and a fine tuning can be achieved.

6 Outlook

In the experiments only some parameters were tested. For example the influence of the probability for a structure mutation was constant during all experiments. In addition, the time when structure mutations are possible can be determined from a random process or from heuristics based on the history of fitness values instead of being fixed externally like in this paper.

Other parameters which should be subject to further research are the number of generations, in which individuals should be protected in separated generations, after a structure optimization was made. Furthermore, it has been shown before, that the estimation of the strategy parameters in evolution strategies has a large influence on the convergence speed.

If the representation grows, the number of strategy parameters also increases (with the exception of the GSA algorithm). The intial values of the step sizes play an important role for the performance of the algorithm. An initialization of newly inserted parameters could have a similar effect on the optimisation. It could for example be based on the values of step sizes of neighbouring object variables, i.e. control points.

Acknowledgments The authors would like to thank E. Körner, T. Arima and T. Sonoda for their support and for stimulating discussions on the topic.

References

[1] A. Vicini and D. Quagliarella. Airfoil and wing design through hybrid optimization strategies. In *16th Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics, 1998. AIAA Paper 98-2729.

[2] D.J. Doorly. Parallel genetic algorithms for optimization in cfd. In J. Périaux and G. Winter, editors, *Genetic Algorithms in Engineering and Computer Science*. Wiley, 1995.

[3] D.J. Doorly and J. Peiró. Supervised parallel genetic algorithms in aerodynamic optimisation. In G.D. Smith, N.C. Steele, and R.F. Albrecht, editors, *Artificial Neural Nets and Genetic Algorithms – Proceedings of the 1997 International Conference*. Springer Verlag, 1998.

[4] S. Obayashi, Y. Yamaguchi, and T. Nakamura. Multiobjective genetic algorithm for multidisciplinary design of transonic wing planform. *Journal of Aircraft*, 34(5):690–693, 1997.

[5] A. Ostermeier. A derandomized approach to self adaptation of evolution strategies. *Evolutionary Computation*, 2(4):369–380, 1994.

[6] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 2000. To appear.

[7] M. Olhofer, T. Arima, T. Sonoda, and B. Sendhoff. Optimisation of a stator blade used in a transonic compressor cascade with evolution strategies. In I. Parmee, editor, *Adaptive Computation in Design and Manufacture (ACDM)*. Springer Verlag, 2000.

[8] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Frommann-Holzboog, 1973.

[9] B. Sendhoff, M. Kreutz, and W. von Seelen. A condition for the genotype–phenotype mapping: Causality. In Thomas Bäck, editor, *Genetic Algorithms: Proceedings of the 7th Int. Conf. (ICGA)*, pages 73–80. Morgan Kaufmann, 1997.

[10] L. Altenberg. Evolving better representations through selective genome growth. In *Proceedings of the 1st IEEE Conference on Evolutionary Computation. Part 1 (of 2)*, pages 182–187, Piscataway N.J., 1994. IEEE.

[11] L. Altenberg. Genome growth and the evolution of the genotype-phenotype map. In Wolfgang Banzhaf and Frank H. Eeckman, editors, *Evolution as a Computational Process*, pages 205–259. Springer-Verlag, Berlin, 1995.

[12] B. Sendhoff and M. Kreutz. Variable encoding of modular neural networks for time series prediction. In V.W. Porto, editor, *Congress on Evolutionary Computation CEC*, pages 259–266. IEEE Press, 1999.

[13] V. K. Vassilev and J. F. Miller. The advantages of landscape neutrality in digital circuit evolution. In J. F. Miller, A. Thompson, P. Thomson, and T. C. Fogarty, editors, *Proceedings of the 3rd International Conference on Evolvable Systems: From Biology to Hardware*, Lecture Notes in Computer Science 1801, pages 252–263. Springer-Verlag, 2000.

[14] L. Piegl and W. Tiller. *The Nurbs Book*. Springer Verlag, 1997.

[15] F. Hausdorff. *Grundzüge der Mengenlehre*. Chelsea Publishing Company, 1965.

[16] H.-P. Schwefel. *Evolution and Optimum Seeking*. John Wiley & sons, New York, 1995.

[17] Th. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.

[18] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaption. In *Proc. 1996 IEEE Int. Conf. on Evolutionary Computation*, pages 312–317. IEEE Press, 1996.

[19] P. Bentley. *Evolutionary Design by Computers*. Peter Betley, Morgan Kaufmann, 1999.