Target shape design optimization with evolutionary computation

Wei-Wen Chang, Chan-Jin Chung, Bernhard Sendhoff

2003

In "IEEE Congress on Evolutionary Computation" pages 1864-1870 , 2003.

Target Shape Design Optimization with Evolutionary Computation

Wei-Wen Chang
Detroit Edison Credit Union
Detroit, Michigan, USA
changrex@hotmail.com

Chan-Jin Chung
Department of Math &
Computer Science
Lawrence Technological University
Southfield, Michigan 48075 USA
chung@ltu.edu

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

***Abstract* - Target shape design optimization problem[1] is to approximate an unknown shape, when a black-box function provides the fitness of the shape. The framework to solve this problem can be applied to the finding of optimized aerodynamic structures such as airplane wings or gas turbine blades.**

In order to approximate the two- and three-dimensional closed curve shapes we use the idea of incremental abstraction, that is to find the best *n*-point polygon in the beginning and double the number *n* in the next stage until *n* reaches the specified limit. This idea is embodied into Evolution Strategies with a modified 1/5 rule using multiple layers. Furthermore, for the three-dimensional hidden object, the problem was decomposed into several 2D optimization problems.

The results show that the suggested method is quite effective to solve the given problems even if the evaluation function has stochastic noise elements. The general strategy can also be applied to other practical applications using evolutionary algorithms.

1 Introduction

Evolutionary algorithms are known to be suitable for optimization problems that are non-differentiable and multi-modal, when an near optimal solution is acceptable (Fogel 1999). For this reason, evolutionary algorithms have been applied successfully to a variety of complex real-world problems, such as structural design, scheduling, and aerodynamic design optimization. However, in many real-world applications, using population based evolutionary algorithms encounters a challenge involving the basic operation – "fitness evaluation" because of several reasons: (1) explicit fitness function is not available due to the lack of data or problem is ill-posed, (2) fitness function is noisy, (3) fitness function varies with time, and/or (4) fitness function is computationally very expensive. In particular, to cope with the last item, i.e., computationally expensive fitness functions, meta-models or approximative fitness functions have been used, see e.g. (Jin et al. 2000, 2002). At the same time, extensive tests of new evolutionary operators or representations cannot be performed for the real-problem due to the high computational cost. Therefore, miniature model problems have to be constructed which should resemble the original optimization problem as close as possible while considerably reducing the computation time.

In the area of design optimization, e.g. turbine blade design (Olhofer et al. 2000) or wing design (Obayashi et al. 1997), the evolutionary process searches in the space of two- or three dimensional shapes represented by a number of coordinate points. Therefore, a suitable miniature model problem is to define a target shape or target structure and to use an appropriate distance measure between two shapes represented by two sets of points as the fitness of the design. Thus, during the search process the fitness has to be minimized. Although the actual quality measure is different from the real-world problem, the phenotype, i.e., the shape, is the same and different variants of evolutionary algorithms can be compared to a certain extent.

Two such target shapes (a two dimensional and a three dimensional one) together with the distance measure were defined by the third author and constituted the design optimization competition of the Congress on Evolutionary Computation (CEC) 2002 and the World Congress on Computational Intelligence (WCCI) 2003, which the first two authors won.

This paper outlines the competition problems and describes the winning evolutionary algorithms. In Section 2, we will define the 2D target shape and the fitness function. The evolutionary algorithm which we employed and the results of the optimization will be outlined in Section 3. In Section 4 the 3D problem is discussed and the approach to solve is introduced. Analysis of the results and future works are explained in Section 5 as a conclusion.

2 Two Dimensional Target Shape Design Optimization Problem

For the two dimensional design optimization competition, an unknown closed curve in two dimensions which was

[1] The 3rd author designed the problems which were part of the competitions during the CEC2001 and WCCI 2003. The 1st and 2nd author won the competition.

represented by two sets of points had to be approximated as closely as possible. In addition, the following information was given:

> "The curve describes a 2D shape like a cross section of a wing or the outline of a car. The 2D structure has no holes, e.g., no donut, is smooth, closed and does not intersect itself. Thus, it can be represented by NURBS (Non-Uniform Rational B-Splines) which are closed and periodic."

Although, the closeness of the fit was estimated by measuring the distances between sets of points, the aim of the competition was the 2D structure, which is best represented by a spline. The reason behind this was that at any stage of the optimization the shape should be reasonably smooth or it should at least be possible to smooth it without any problems. The reason is that for many real-world design optimization tasks, the shape can only be evaluated with computation modelling techniques like computational fluid dynamics if the structure is reasonably smooth.

2.1 Constraints

The following constraints were set for the two dimensional design optimization problem:

- The fitness value is provided by a black-box function, which is contained in an object file. In addition to the object file an executable is available which reads a data set and returns the fitness value.
- Since the fitness function is stochastic (see Section 2.2), the fitness value which is submitted should be the average of at least 10 evaluations of the final individual.
- The length of the optimized curve has to be within 30% of the length of the original NURBS curve, which is 3.55. The length of the curve can be approximated by the length of the polygon through the data points.
- The maximum number of evaluations, i.e., generations times population size, which are allowed to be used is 100000.

2.2 The design of the evaluation function

The stochastic evaluation function receives a vector of 500 points in 2D space. Then the function compares the target set to the set contained in the specified file and returns a fitness value based on a kind of averaged symmetric Hausdorff distance:

$$\frac{1}{8N}\left(\sum_{x\in C}\min_{y\in A}\{|y-x|\}+\sum_{x\in C'}\min_{y\in A}\{|y-x|\}\right)+$$
$$\frac{1}{8N}\left(\sum_{x\in C}\min_{y\in B}\{|y-x|\}+\sum_{x\in C'}\min_{y\in B}\{|y-x|\}\right),$$

where N is the number of points, A and B are two slightly differently sampled sets of points from the target shape, i.e., a different offset was used when moving along the spline for sampling A and B. C is the set of points defined by each individual and the points of the set C' are stochastic interpolations between two neighbouring points of C as outlined in Figure 1.

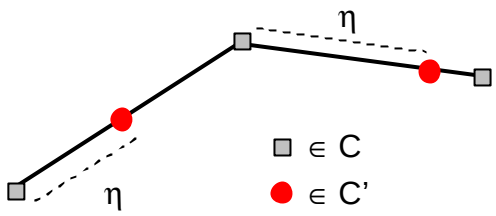


Figure 1: The points of the set C are given by each individual, the ones for C' are the result of an interpolation between two neighbouring points, the distance h **is chosen at random.**

Therefore, the distance is calculated between four different sets. The two data sets C and C' are compared with the two target sets A and B for each individual. In this way, we try to achieve a distance between the two curves and not just between the two sets of points. Of course, this means that the fitness evaluation is stochastic, although the fluctuations for a typical curve are fairly small. At the same time, a stochastic aspect of the fitness function is often been found in real-world problems and therefore, adding a stochastic element to the target shape optimization brings us closer to the real design optimization problem.

3 Evolutionary Strategies for the 2D Target Shape Design Optimization

The employed evolutionary method to solve the problem is based on a $(m + l)$ Evolution Strategy with a modified 1/5 rule. The following components of the Evolution Strategy need to be determined:

- The individual (solution) representation;
- improved variation (mutation and crossover) operators (tuned to the design optimization problem); and new ways to adjust the adaptive parameters and to influence/guide the variation of individuals using those parameters;
- The population size.

The first challenge we have is to determine the representation for the problem; it needs to be complex enough to describe all the possible solutions, but it needs to be simple enough to abstract the high dimensionality of the problem. A dynamic control point structure is being used, the details of which are described in Section 3.1.

Based on our representation, we introduce variation operators that maintain the highest degree of freedom of

changes, but also keep the solution within the problem constraints. We introduce a two-layered evolution strategy with a global level step-size for the whole population (world) and local level step-sizes for each individual shape. They control how much an individual can be changed. A genetic crossover is also employed, which makes individuals share their "lifetime leaning" experience. More details about variation operators and the evolutionary algorithm are described in Section 3.2.

3.1 Representation of the 2D geometry

To describe a 2D geometry or shape in great detail may require a large number of variables, which increases the dimension of the search space. High dimensionality has serious impact on the performance of evolutionary computation, since the number of iterations is rapidly increased and often the number of local optima also increases (in design optimization one problem with local optima occurs when the shape intersects itself and it is intuitive that the likelihood of such intersections will increase with the variability of the shape and thus in general with the dimensionality of the search space) . On the other hand, if we use only a limited number of variables to represent the 2D object, we might not be able to represent the optimal solution which in turn could never be found.

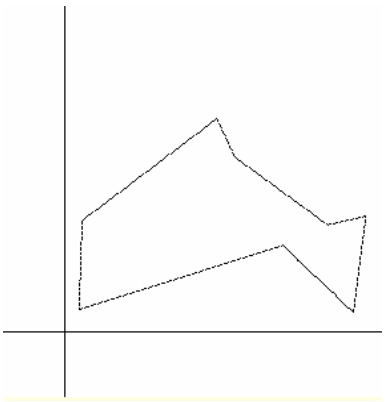


Figure 2:Individual with eight control points.

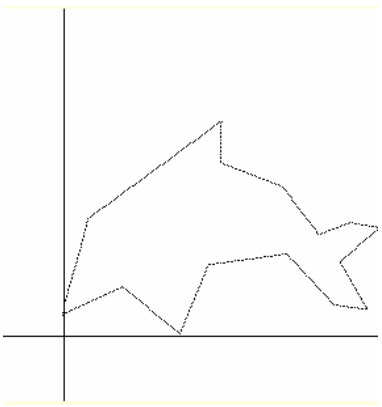


Figure 3: Individual with 16 control points

In order to have both, flexibility and compactness in our individual representation, we introduced a variable number of control points. The concept is to increase the number of control points with increasing generations. In the beginning of evolution, each individual will have a very small number of control points to represent a 2D shape. In our experiments, initially each individual contains four control points. When the population reaches its limit and does not produce any improved individuals anymore, the degree of freedom of all individual shapes is increased, thus control points are added.

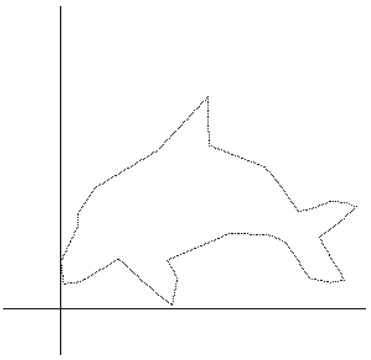


Figure 4: Individual with 32 control points

Then we doubled the control points until the number reached a predefined value. An alternative would be an adaptive approach where the update is integrated into the evolutionary process (Olhofer et al. 2001). Using a variable or dynamic number of control points the evolutionary process could achieve the best result while using a minimal number of fitness evaluations, even if the evaluation function had some stochastic elements. Figures 2-4 show the best shapes during early generations of the CEC 2001 2D problem with 8, 16, and 32 control points.

3.2 Variation operators and the evolutionary algorithm

We can learn many different aspects of generating offspring solutions from parent solutions from nature or societies. In our design optimization experiments, we use two forms of reproduction: sexual and asexual. Using the crossover operator two individuals exchange their control points to form a new individual. Asexual reproduction is essentially first cloning one individual and then mutating the control points of the clone.

There are different ways to mutate the shape; we can change all the points at once for each shape, or only change a certain number of points. However, the challenge is how to maintain individuals under the basic constraints such as closeness and continuity. If points are changed at random with full freedom, it is easy to generate a shape that is not closed. In order to solve this problem, the concept of a center control point was introduced. To find the center control point, we first randomly select ten sets of lines (randomly select two points from initial shape), then find the average mid point of these ten lines. To simplefy our approach, only the distance between points and the center control points is changed, when we mutate a point. In this way, closeness of the shape can be easily

maintained. Control points have a predefined order that will not change, and before every fitness evaluation, points will be filled (interpolated) in the gap between two neighboring control points to complete the 2D shape as described in Figure 5.

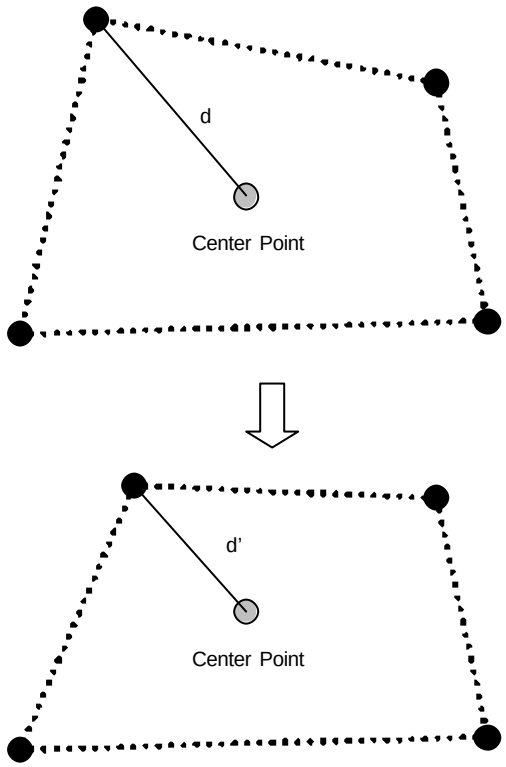


Figure 5: The center control point, four control points and interpolated points for evaluation; shown is an example of the mutation operation. Note, that the center point position is varying.

However, limiting the mutation of a point by only changing the distance to the center control point will decrease the degree of freedom of changing the individual. To overcome this issue, we change each center control point for the shape according to the step sizes learned, before we start to mutate each point. As it is done in standard ES, mutation is realized by adding a normally distributed random number. The update of the different strategy parameters (see below), the variances of the Gaussian distribution, is carried out by using Rechenberg's 1/5 rule, see e.g. (Schwefel 1995): *From time to time duringthe evolution process check the ratio of the number successes to the total number of trials (variations). If the ratio is greater than 1/5 increase the variance; if it is less than 1/5, decrease the variance.*

A generic crossover is also used in the 2D design problem. Since each control point has a fixed predefined order, we can simply exchange control points at identical positions and check if the 2D individual has improved after the change. The new individual only survives if the crossover operation was successful, i.e., it has a higher fitness value than either of its parents. An example of this process is shown in Figure 6, where combining control points (1), (2), (3) and (4') generated an offspring better than parent *p1*.

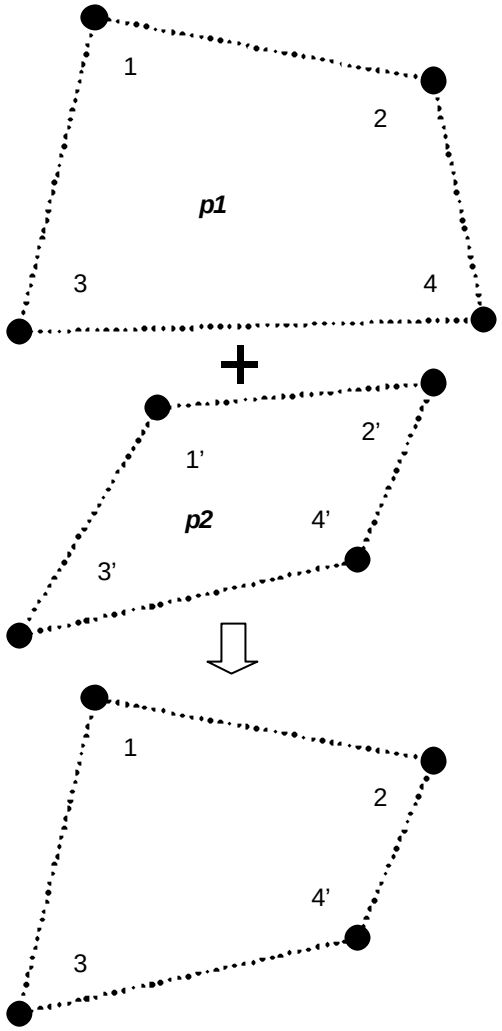


Figure 6: Crossover of two shapes. Explanation given in the text.

Due to the structure of our 2D representation explained in the previous paragraphs, we consider a two-layered evolution strategy with two levels of step sizes. 2D shape individuals defined by control points are evolved by using a $(m+l)$-ES, where $m=l=8$ was used in the experiments. A local step size for each 2D shape is maintained and updated after the mutation of each control point. A global step size is maintained at the world (population) level and updated at the end of each generation. This structure simulates our real world with multiple societies. The 2D objects are like societies, and each control point is viewed as a parameter to characterize a certain culture. A global knowledge (global step size) is shared and updated by all the societies, and a local knowledge is shared and updated in a society to modify its parameters (control points). The variation operator uses these two step-sizes to determine the size of the changes. When the societies are formed in the beginning, only a few parameters (control points) are important. The other parameters are interpolated between

dominant parameters (control points) during the evolution process. More detailed social parameters will be generated (using the dynamic representation) and will have the ability to form the concrete society in later generations. This idea is depicted in Figure 7 and similar studies can be found in (Chung and Reynolds 1998, 2000). Overall view of the evolutionary algorithms in pseudo code is given below:

```
Initialize individuals*population size
Repeat until termination condition is satisfied
   For each individual i_ n
    new_center_control = center_control +
                         Gaussian(0, f(stepsizeG))
      For each control points
         length(new_control, new_center_control)  =
         length(old _control, new_center_control) +
           Gaussian(0, f(stepsizeL, stepsizeG))
         If better
                old_control = new_control
                stepsizeL = stepsize/.82
         Else
                stepsizeL = stepsize*.82
         End-If

      End-for
   End-for

   Crossover some individuals using step sizes

   If  best individual is better than the best
       individual of last generation then 4%
            stepsizeG = stepsize/.82
   Else-if this generation reach its limit
      Update size of the control points
            stepsizeG=0.82
```

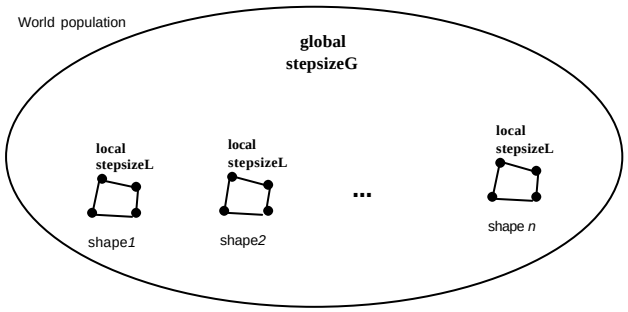


Figure 7: World population with multiple societies to be evolved.

3.3 Results and Analysis of 2D design optimization for CEC2001 Competition

By using the algorithm oultined above, we were able to find a near perfect shape of a dolphin shown in Figure 8. The number of evaluations used was 100,000 and the best score was 0.003. This shows that our algorithm is quite effective and practical to solve 2D design optimization problems.

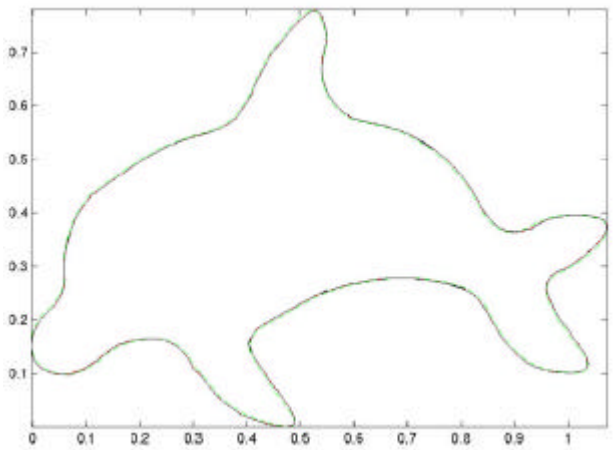


Figure 8: The hidden shape found by the outlined evolution strategy for the CEC 2001 2D design optimization. competition.

4 3D Target Shape Optimization

4.1 Problem Definition

In the 3D design optimization task, an unknown three-dimensional surface had to be approximated as closely as possible. In addition, the following information was given:

> "The surface is closed in both parameter directions and describes a 3D structure like a bottle or a teapot. The target surface is smooth, has no holes, i.e., it is not like a donut, and does not intersect itself. Thus, it can be represented by a closed and periodic Non-Uniform Rational B-Spline (NURBS) surface. "

The closeness of the fit to the target surface is basically estimated in the same way as in the 2D case. Again four data sets are used. The two data sets which were sampled from the original structure are compared to the data set of each individual and to the data set which was generated from each individual using stochastic interpolation between neighboring points. As in the 2D case, the aim was to find the 3D target structure and not just a set of points. Again the background is that in many real world applications certain constraints like smoothness have to be fulfilled for evaluation using computational modeling techniques. Since the fitness function was stochastic again, the submitted fitness value had to be the average of at least 10 evaluations of the final individual. No maximum number of evaluations was imposed. However, in case two submissions were of nearly identical quality but one used significantly fewer evaluations, the more resource sparing submission would have won.

4.2 Multi-staged Evolution Strategies to solve 3D Design Optimization Problems

The main idea to solve the 3D design optimization problem is to break down the problem into several 2D optimization problems, by introducing multiple stages. In the first stage, we will focus on finding the best 3D rectangle (hexahedron) to approximate the unknown object. We only need to find 8 points for the shape in this stage.

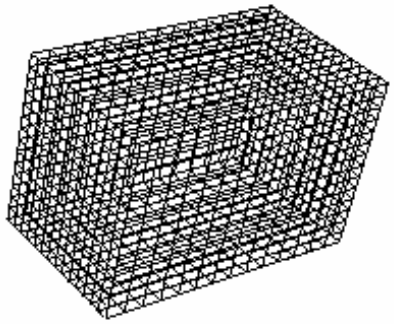
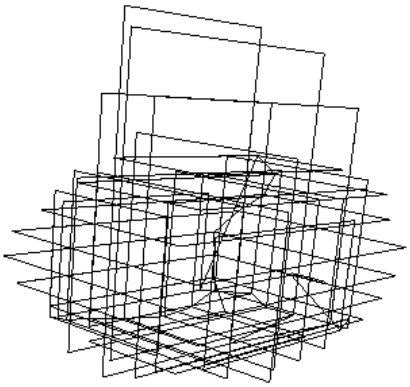
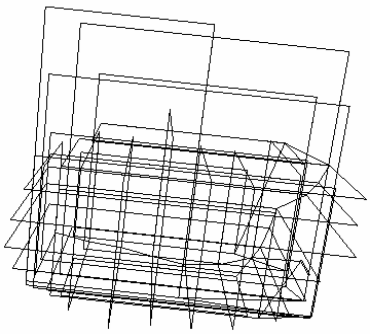


Figure 9: 3D structure after stage 1. Explanation given in the text.

In the second stage, we break the hexahedron down to forty-five 2D rectangles using a (1+1)-ES to find them. The procedure is very similar to the first stage for finding the best hexahedron. Here in this stage, each 2D rectangle requires only 4 control points. In order to find a specific 2D rectangle, the remaining 44 rectangles are fixed and a (1+1)-ES is used to find best 2D rectangle.



(a)



(b)

Figure 10: 3D structures after stage 2. Explanation given in the text.

The final stage, which is the most important and time-consuming part, is to focus on finding all forty-five 2D shapes. In order to find each 2D shape, we used the same method used in the 2D design optimization algorithm with a two layered structure and a dynamic representation discussed in previous sections. All three stages are visualized in Figures 9, 10 (a) -(b) and 11 (a) -(b)

4.3 Results and Analysis of 3D design optimization for WCCI 2002 competition

By using the above algorithm, we were able to find a near perfect shape of a duck shown below in Figure 12. The number of evaluations used was 1,728,000 and the fitness value averaged over 100 evaluations was 0.06175. Although the final 3D structure is not as smooth as might be needed for some real-world design optimization problems, it demonstrates quite clearly that our algorithm can effectively be used to tackle 3D design optimization problems. In particular the hierarchical way to decompose the 3D structure into substructures which can be tackled in a similar way as the 2D design optimization problem, seems to be promising strategy to cope with the increased dimensionality of the search space of 3D design optimization, in particular in the context of conceptual design optimization.

(a)

(b)

Figure 11: 3D structures after stage 3. Explanation given in the text.

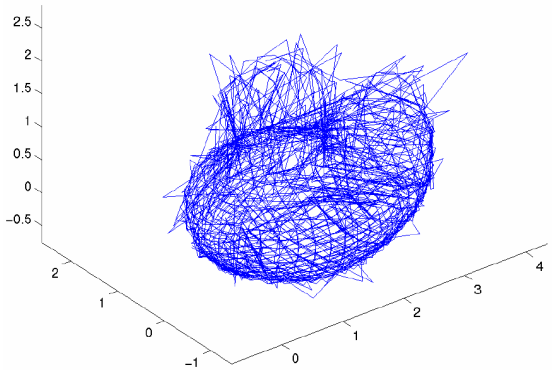


Figure 12: 3D shape found in the WCCI 2002 competition. The duck looks a bit tattered but it can be recognized.

5 Conclusions

In this paper, we presented the two design optimization competitions of the CEC 2001 and WCCI 2002 together with an outline of the winning algorithm. Although target shape optimization is only a miniature model problem in the field of design opimization, we believe it gives us the possibility to explore and empirically test new methods in particular with respect to the representation and the variation operators. The winning $(m+l)$-ES with a dynamic representation or population size if we regard each control point as an individuals, realizes a dynamic balance between compactness and variability of the representation of 2D shapes. A similar hierarchical approach was also chosen to tackle the 3D target shape design optimization problem. Here we first decomposed the structure into subunits which we could then shape using the 2D approach.

Based on these empirical studies, we can conclude that the suggested method based on multi-staged incremental approaches is very promising and practical for design optimization problems. Two level step size adaptation with local and global mutation variances was shown to be very useful for this type of problems. Future work includes introducing different selection mechanisms.

Acknowledgments

The CEC 2001 competition was sponsored by Honda R&D Europe GmbH and the WCCI 2002 3D competition was sponsored by the Evolutionary Programming Society. Bernhard Sendhoff would like to thank Martina Hasenjäger and Markus Olhofer for their support during the preparation of both competitions.

Bibliography

Chung, C.-J. and Reynolds, R.G. (2000) “Knowledge-Based Self-Adaptation in Evolutionary Search”, International Journal of Pattern Recognition and Artificial Intelligence Vol.14, No.1, page 19-33.

Chung, C.-J. and Reynolds, R.-G. (1998) "CAEP: An Evolution-Based Tool for Real-valued Function Optimization Using Cultural Algorithms," International Journal on Artificial Intelligence Tools, Vol.7, No.3, pages 239-292.

Fogel, D.B. (1999) “Evolutionary Computation: Toward a New Philosophy of Machine Intelligence”, IEEE Press, 2nd edition

Jin, Y., Olhofer, M. and Sendhoff, B. (2002) “A Framework for Evolutionary Optimization with Approximate Fitness Functions”. IEEE Transactions on Evolutionary Computation, Vol.6, No.5, pages 481-494.

Jin, Y., Olhofer, M. and Sendhoff, B. (2000) “On evolutionary optimization with approximate fitness functions”. Proceedings of the Genetic and Evolutionary Computation Conference, pages 786-792. Morgan Kaufmann.

Obayashi, S., Yamaguchi, Y. and Nakamura, T. (1997), “Multiobjective Genetic Algorithm for Multidisciplinary Design of Transonic Wing Planform”, Journal of Aircraft, Vol.34, No.5, pages 690-693.

Olhofer, M., Arima, T., Sonoda, T. and Sendhoff, B. (2000) “Optimisation of a stator blade used in a transonic compressor cascade with evolution strategies”, In I.C. Parmee, editor, Adaptive Computing in Design and Manufacture (ACDM), pages 45-54. Springer Verlag.

Olhofer, M. Jin, Y. and Sendhoff, B. (2001), “Adaptive encoding for aerodynamic shape optimization using Evolution Strategies”, Congress on Evolutionary Computation (CEC), Seoul, Korea, Vol.2, pages 576-583.

Schwefel, H.-P. (1995) “Evolution and Optimum Seeking”, Wiley Press.