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

Target shape design optimization by evolving B-splines with cooperative coevolution

Zhenyu Yang, Bernhard Sendhoff, Ke Tang, Xin Yao

2016

Preprint:

This is an accepted article published in Applied Soft Computing. The final authenticated version is available online at:
https://doi.org/10.1016/j.asoc.2016.07.027

*Manuscript
Click here to view linked References

Target Shape Design Optimization by Evolving B-splines with Cooperative Coevolution

Zhenyu Yang, B. Sendhoff, Ke Tang, and Xin Yao, *Fellow, IEEE*

This work was supported in part by the National Natural Science Foundation of China under Grants 61175065, 61329302 and 61305084, the Program for New Century Excellent Talents in University under Grant NCET-12-0512, and the Natural Science Foundation of Anhui Province under Grant 1108085J16.

Zhenyu Yang is with the College of Information System and Management, National University of Defense Technology, Changsha 410073, China (email: zhyuyang@ieee.org). B. Sendhoff is with the Honda Research Institute Europe GmbH (email: bernhard.sendhoff@honda-ri.de). Ke Tang and Xin Yao are with the USTC-Birmingham Joint Research Institute in Intelligent Computation and Its Applications (UBRI), School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China. Xin Yao is also with the Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, U.K. (emails: ketang@ustc.edu.cn, x.yao@cs.bham.ac.uk).

Abstract

With high reputation in handling non-linear and multi-model problems with little prior knowledge, evolutionary algorithms (EAs) have successfully been applied to design optimization problems as robust optimizers. Since real-world design optimization is often computationally expensive, target shape design optimization problems (TSDOPs) have been frequently used as efficient miniature model to check algorithmic performance for general shape design. There are at least three important issues in developing EAs for TSDOPs, i.e. design representation, fitness evaluation and evolution paradigm. Existing work has mainly focused on the first two issues, in which (1) an adaptive encoding scheme with B-spline has been proposed as a representation, and (2) a symmetric Hausdorff distance based metric has been used as a fitness function. But for the third issue, off-the-shelf EAs were used directly to evolve B-spline control points and/or knot vector. In this paper, we first demonstrate why it is unreasonable to evolve the control points and knot vector simultaneously. And then a new coevolutionary paradigm is proposed to evolve the control points and knot vector of B-spline separately in a cooperative manner. In the new paradigm, an initial population is generated for both the control points, and the knot vector. The two populations are evolved mostly separately in a round-robin fashion, with only cooperation at the fitness evaluation phase. The new paradigm has at least two significant advantages over conventional EAs. Firstly, it provides a platform to evolve both the control points and knot vector reasonably. Secondly, it reduces the difficulty of TSDOPs by decomposing the objective vector into two smaller subcomponents (i.e., control points and knot vector). To evaluate the efficacy of the proposed coevolutionary paradigm, an algorithm named CMA-ES-CC was formulated. Experimental studies were conducted based on two target shapes. The comparison with six other EAs suggests that the proposed cooperative coevolution paradigm is very effective for TSDOPs.

Index Terms

Target Shape Design Optimization, Cooperative Coevolution, B-spline, Adaptive Encoding, CMA-ES

I. Introduction

DESIGN optimization has been an important area of research for many years [1]. In recent years, as robust optimizers with excellent ability in handling non-linear and multi-model objective functions, evolutionary algorithms (EAs) have successfully been applied to structural design optimization problems [2], [3], [4], especially in the field of aerodynamics, e.g. [5], [6], [7]. However, for real-world applications, especially in the area of aerodynamic design optimization, evaluation of a given design solution often requires high computational efforts such as the evaluation with Computational Fluid Dynamics (CFD) tools. The search for the most appropriate algorithm including the design representation and its parameterization based on computationally expensive CFD simulations would be very time consuming and for complex three-dimensional shapes currently impossible [1]. Therefore, miniature models or highly simplified CFD models have been suggested to ease the computational load in the early conceptual algorithmic phase. **Target Shape Design Optimization Problems** (TSDOPs) [8], [9] have been put forward as miniature models in particular to test the usage of specific design representations. Although it is known that neither miniature models like TSDOP nor simplified CFD models correlate with the detailed CFD analysis, individual algorithmic issues can be studied and results successfully transferred to real-world optimization problems. Given a **target shape**, the TSDOP is defined as minimizing the distance between the target shape and the designed shape with respect to an appropriate distance measure. We focus on TSDOP in this paper, since the interplay between representation and algorithm is the main contribution of this work.

[1]According to the experience in Honda Research Institute Europe GmbH, the calculation time for a detailed CFD analysis of a complex 3D structure takes roughly one day parallelized on several compute nodes.

Evolutionary computation for TSDOPs was first investigated in [8], and then further studied in [10], [11] and [12]. Generally speaking, they mainly focus on the following three issues which are very crucial in applying EAs to TSDOPs.

1) Representation: Representing the shape is the initial step and it determines the landscape of the search space. A number of characteristics, such as completeness, causality and compactness, have been suggested in [5] as the required properties of a suitable representation. A commonly used representation is the parameterized B-spline curves or surfaces [8], [12]. Given a B-spline as the geometry representation, the objective parameters of TSDOP consist of the coordinates of the control points and the corresponding knot vector(s).
2) Fitness evaluation: For TSDOPs, since shapes are often represented by a set of sampled points, the designed shape can be evaluated by calculating the distance over two sets of points sampled respectively from the target shape and the designed shape with respect to an appropriate distance metric, e.g. the commonly used averaged symmetric Hausdorff distance [8]. An improved fitness metric has also been proposed in [12] to alleviate the problem of wrongly assigning good quality shapes to bad.
3) Evolution: This step determines the driving force of evolution. With a B-spline as the geometry representation, existing efforts for this issue can be classified into two main categories: (i) Represent the designed shape with a predefined number of control points, and then optimizes the coordinates of control points and the knot vector with a certain EA; (ii) Represent the designed shape with **adaptive encoding** [8], which starts from a minimal representation length, and then optimize the coordinates of control points with a certain EA, and meanwhile increase the number of control points during evolution with a random **knot insertion** [13] algorithm. The first kind of evolution paradigm has at least two obvious drawbacks. Firstly, for an unknown target shape, it is often very difficult to determine the optimal number of control points. That is why the adaptive encoding scheme has been proposed. Secondly, it is unreasonable to optimize the control points and knot vector simultaneously using those EAs in which crossover is the necessary or essential operator. The reason is that knot vector and control points of B-spline are highly correlated. It makes no sense to perform crossover between two control-point sets whose corresponding knot vectors are different [2]. As for the other paradigm using adaptive encoding, although it has been proven better than the first one, the main difficulty is that the variation of the knot vector can only be obtained through irreversible knot insertion operations. Random knot insertion method is often adopted since it is very difficult to determine the optimal insertion position during evolution. This will at best limit flexibility of the designed shape and in the worst case take a risk to introduce very bad knot position.

For the issue of representation, we can still use a B-spline with adaptive encoding, since it has been verified as an effective choice [8]–[12]. For fitness evaluation, the only thing we can do is to provide a similarity metric, and eventually we have to abandon it because we need to use CFD tools for real-world applications. So among the three issues, we are mostly interested in the third one, i.e. designing appropriate evolution paradigm for TSDOPs.

In order to develop a more effective evolution paradigm for TSDOPs, we propose a new coevolutionary paradigm in this paper to evolve the control points and knot vector of a B-spline separately in a cooperative manner. This kind of evolution model is called Cooperative Coevolution (CC) [14], [15], which was originally proposed to solve large and complex problems. In the new paradigm, adaptive encoding is still utilized as the basic representation. The objective variables will be initially divided into two parts. The first part includes the coordinates of all control points, while the second part consists of knot

[2]The reason will be demonstrated in Section II-C

points. An initial population is generated for each of the two parts at the beginning of evolution. And then the two populations are evolved separately in a round-robin fashion, which means the current population will evolve for a predefined number of generations before activating the other one. For fitness evaluation, complete representation will be assembled with individuals from the active population and a representative from the unactivated population. The best individual so far is selected as the representative. CMA-ES [16] is adopted to evolve each of the two populations due to its excellent convergence feature with very limited number of fitness evaluations (FEs). The new algorithm is denoted as CMA-ES-CC. The advantages of CMA-ES-CC is apparent. It not only provides a platform to evolve both control points and knot vector(s), but also reduces the difficulty of TSDOPs by decomposing the objective vector into two smaller subcomponents (i.e., control points and knot vector). To evaluate the efficacy of CMA-ES-CC, experiments have been conducted based on two two-dimensional (2D) target shapes. The comparison with conventional EAs suggests that the proposed cooperative coevolution paradigm is very effective and efficient for TSDOPs.

The rest of this paper is organized as follows: Section II introduces the detailed formulation of TSDOP, which includes representation, fitness evaluation and related work. The drawbacks of existing evolution paradigms are also discussed in this part; Section III presents the proposed cooperative coevolution paradigm for TSDOPs; Section IV provides the experimental studies based on two target shapes; Finally, Section V concludes this paper, and discusses a few directions for future work.

II. Target Shape Design Optimization Problem

As stated in Section I, in this paper target shape design optimization problem (TSDOP) will be used as the miniature model of design optimization. The solution of TSDOP is to operate the designed shape to fit the given target shape. The basic components of TSDOP, which includes representation scheme and fitness evaluation method, will be described in this section. Related work on evolutionary approaches for tackling TSDOPs will also be reviewed briefly.

A. Representation

There are several shape parametrization techniques for shape representation and manipulation [17]. Due to the compact representation and a number of advantages, Non-Uniform Rational B-Splines (NURBS) are one of the most popular geometry representations for shape manipulation and optimization [12]. According to [13], a p-degree NURBS curve is defined in parametric form by the following equation

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

where $\{\mathbf{P}_i = (x_i, y_i), 0 \leq i < n\}$ are the n control points, the $\{w_i, 0 \leq i < n\}$ are coefficients weighting the influence of each control point on the corresponding spawn, and $N_{i,p}(u)$ are the p-degree B-spline basis functions defined as follows

$$N_{i,0}(u) = \begin{cases} 1, & \textit{if } u_i \leq u < u_{i+1} \\ 0, & \textit{otherwise} \end{cases}, \tag{2}$$

$$\begin{aligned} N_{i,p}(u) = & \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) \\ & + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \end{aligned} \tag{3}$$

defined on the knot vector

$$\mathbf{U} = \{u_0, \cdots, u_{n+p}\} \tag{4}$$

The control points determine the shape of the NURBS curve. Typically, each point of the curve is computed by taking a weighted sum of a number of control points. The weight of each control point changes according to the B-spline basis functions. A knot vector is an ascending sequence of parameter values that determines where and how the control points affect the NURBS curve. It divides the parametric space into a number of intervals, which are usually referred as the **knot spans**. For a p-degree curve, the weight of any control point is nonzero in only $p+1$ knot spans. Each time the parameter value enters a new knot span, a new control point becomes active, while one of the old control point will be discarded. This property, known as **local support**, is highly desirable, because it allows the changing of one part of a curve while keeping other parts constant [18].

In case of $w_i = 1$ in Eq. (1) for $i = 0, \cdots, n-1$, the NURBS is identical to a B-spline curve, which is often used in practice to reduce the number of objective parameters [8]. With a predefined degree p, each designed curve can be encoded by the coordinates of the control points and its knot vector. Therefore, the representation of an individual for a designed curve could be:

$$\begin{aligned} \mathbf{I_1} &= (x_0, y_0, \cdots, x_{n-1}, y_{n-1}) \\ \mathbf{I_2} &= (u_0, u_1, \cdots, u_{n+p}) \end{aligned} \tag{5}$$

Such a representation, in which the number of parameters keeps constant during evolution, is called **static representation**. In practice, the optimal number of parameters of the static representation is often very difficult to determine because of two reasons. On the one hand, a low number of parameters can be chosen to pursue a fast convergence, but it may also cause the optimization algorithms premature with an unsatisfying solution. One the other hand, although a high number of parameters is beneficial to achieve a high degree of freedom, it will at best lead to an increase of the computational demand and in the worst case to a non-convergent search process [8].

To tackle the above dilemma, an **adaptive encoding** scheme, which is aimed at combining the advantages of low-dimensional search with those of increasing the degrees of freedom when needed, has been suggested in [8] to automatically increase the number of control points during evolution. The basic idea behind the adaptive encoding 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 are more likely to 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. The change of representation is called **structure mutation**, and for B-splines it can be performed easily by adding control points without changing the current designed shape according to the **knot insertion** algorithm [13]. The adaptive encoding aimed at providing a balance between the following considerations:

1) The fast convergence to good regions in low-dimensional search space;
2) The sufficient degrees of freedom in high-dimensional case.

The superiority of adaptive encoding over static representation has been shown in [8].

In this paper, B-spline and adaptive encoding will be used as the representation. Since we intend to evolve the knot vector

as well as the coordinates of control points, the encoding for the knot vector is changed slightly as follows

$$\mathbf{I_2} = (v_0, v_1, \cdots, v_{n+p-1}) \tag{6}$$
$$v_i = u_{i+1} - u_i, \forall i \in \{0, \cdots, n+p-1\}$$

This is based on the consideration that the knot vector is an ordered list, and it should still be ordered after evolutionary operations. However, it might be very difficult to maintain such an ordered list if we operate the knot vector directly. Although, it is possible to repair the disordered list by sorting, the nature of the operated knot vector may be changed. Instead, if we manipulate the distance between any two adjacent knot points, the knot vector will be always ordered automatically.

B. Fitness Evaluation

Given TSDOP as the miniature model for shape design, the common way to measure the distance between the designed shape and the target shape is based on a modified Hausdorff distance, which calculates the distance values of two sets of points sampled from the two shape respectively. Formally it calculates the distance based on a kind of averaged symmetric Hausdorff distance [8]:

$$\begin{aligned} f_{dist} =& \frac{1}{2}(\sum_{i=1}^{|z_1|} \min_{b \in z_2} |a_i - b|^2 + \sum_{j=1}^{|z_2|} \min_{a \in z_1} |b_j - a|^2) \\ =& \frac{1}{2}(A + B) \end{aligned} \tag{7}$$

where z_1 is the set of sampled points from the target curve, and z_2 is the set from the designed curve. $|z_1|$ and $|z_2|$ are the cardinalities of the sets. The first term is to collect the distance values for the points in the target curve to their nearest points in the designed curve; the second term is that from the points in the designed curve to the nearest points in the target curve.

In [12], it has been found that in some cases the averaged symmetric Hausdorff distance tends to fail to assign the corresponding good evaluation value to a good designed shape. This is because the averaged result of A and B may hide the respective variation of them. To alleviate such a problem, a replacement, which chooses the worse evaluation value between the terms A and B as the fitness evaluation rather than the averaged one, has been suggested in [12]. It has been given as follows:

$$\begin{aligned} f_{dist} =& \max(\sum_{i=1}^{|z_1|} \min_{b \in z_2} |a_i - b|^2, \sum_{j=1}^{|z_2|} \min_{a \in z_1} |b_j - a|^2) \\ =& \max(A, B) \end{aligned} \tag{8}$$

It has been demonstrated that the modified fitness evaluation not only holds the basic causality that well fitted designed shape receives good evaluation value, but also can improved the evaluation mechanism on some cases that the original method always tends to fail [12]. The improved fitness evaluation method will be used in the present paper. In practice, loops in designed curves are undesired, because they prevent curves from being reasonably smooth. To avoid such loops, a penalty term is often added to the fitness value of designed curve.

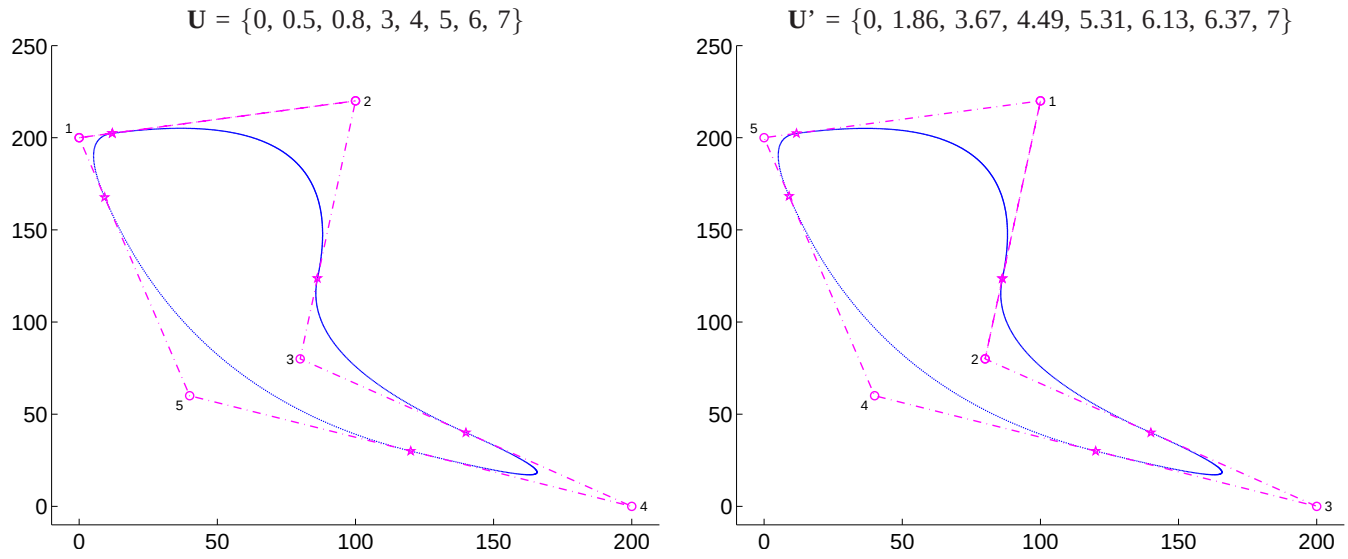

Fig. 1. Two identical B-spline shapes generated with the same control-point sets, but different knot vectors. Note that the sequences of their control points are not the same. Crossover between the two shapes will not be able to match the control points of the two sequences correctly. The coordinates of the 5 control points are {(0, 200), (100, 220), (80, 80), (200, 0), (40, 60)}.

C. Related Work and Discussion Toward Evolution Paradigm

As an effective miniature model, TSDOP has become a popular platform to check the performance of evolutionary computation techniques for structural design optimization problems. In [8], the TSDOP problem was first introduced and the adaptive encoding approach, which has been presented in Section II-A, was proposed by means of a growing representation for B-spline coded structures. The design process was started with a minimal representation length, and then the length was adapted by structural mutations. Meanwhile, the fitness evaluation method was presented for the designed shapes. Several evolution strategies (ESs) were employed to perform the optimization task. It has been found that an algorithm using adaptive encoding significantly outperforms others that use a fixed number of control points. Later in [10], TSDOP was further investigated with explicit and intensive experimental results. It introduced the idea of incremental abstraction, which 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. ES with a modified 1/5 rule [19] was used to implement this idea, and the formulated algorithm won the design optimization competition of the 2001 IEEE Congress on Evolutionary Computation (CEC2001). In [11], the applicability of polygon shape morphing methods was studied in the standard ESs for the design optimization with TSDOP as benchmark problems. Most recently, an improved fitness evaluation method (see Section II-B) was suggested in [12], in order to more reasonably check the quality of designed shapes. A novel mechanism was also proposed to perform the searching smoothly by adapting their search strategy parameters. Results reported in the paper represent the latest progress of evolutionary approaches for TSDOPs.

In all the above work, the adopted evolution paradigms can be classified into two main categories:

1) Represent the designed shape with a predefined number of control points, and then evolve both the control points and the corresponding knot vector using a certain EA;

2) Represent the designed shape with adaptive encoding [8], and then evolve the control points using a certain EA, and meanwhile expand the representation gradually by increasing new control points with the random knot insertion algorithm.

The first kind of paradigm has at least two drawbacks. Firstly, for an unknown target shape, it would be very difficult to determine the optimal number of control points. That is why the adaptive encoding scheme is proposed. Secondly, it is unreasonable to evolve the control points and knot vector simultaneously using those EAs in which crossover is the necessary or essential operator. For two (or more) individuals of a population, before crossover we always assume they are in **consistent representation**, which means the objective parameters of them are one-to-one corresponded. For example, it is reasonable to execute crossover between $(x_1, y_1, x_2, y_2, x_3, y_3)$ and $(x'_1, y'_1, x'_2, y'_2, x'_3, y'_3)$. But it makes no sense to do it between $(x_1, y_1, x_2, y_2, x_3, y_3)$ and $(x'_2, y'_2, x'_3, y'_3, x'_1, y'_1)$, because each pair of values in the corresponding positions are not samples from the same parameter. Obviously, the effect of such kind of crossover is unpredictable. So it is merited to guarantee the individuals of a population are always in the consistent representation if crossover operations are applied. However, this is not true if the control points and knot vector are evolved simultaneously. For two B-spline curves, the one-to-one correspondence between their control point sets may be destroyed if they have different knot vector. A simple example was given in Fig. 1, in which the two B-spline curves are identical in phenotype. But they have different knot vectors and the same control-point sets in different sequences, which means their representations are not consistent. In such a case, crossover between the two shapes will not be able to match their control points in the two sequences correctly. To be specific, the first control point of the left curve will wrongly crossover with the first control point of the right curve, while it should do that with the fifth control point of the right curve. For such cases, one may suggest to execute an alignment before any crossover operation. However, such alignment is possible only when the represented shapes by the parents are already reasonably similar, which implies the optimizer has entered into the last evolution stage for convergence. In other important evolution stages for exploration, we are not able to identify the correspondence of control points for alignments. As for the other evolution paradigm using adaptive encoding, although it has been proven better than the first one, the main difficulty is that the variation of the knot vector can only be obtained through irreversible knot insertion operations. Random knot insertion method is often the only choice since it is very difficult to determine the optimal insertion position [8]. This will at best limit flexibility of designed shape and in the worst case take a risk to introduce very bad knot positions. Moreover, the consistent representation will also be destroyed if the random knot insertion is individual-based, because different individuals may have different insertion positions and thus result in different knot vectors.

In summary, existing evolution paradigm for TSDOP ignored the concept of consistent representation. They either optimize the control points and knot vector simultaneously, or optimize only the control points but still perturb the related knot vector individually using knot insertion operations. In fact the control points and the corresponding knot vector should not be changed at the same time, because that will always change the consistency of representations among all the individuals. This is supported by the fact that in previous work it was found that the crossover (including recombination) operators are not helpful, and they often turned out to be too disruptive in most cases [8]. Hence these operators were usually not recommended. However, for many state-of-the-art genetic and evolutionary algorithms (GAs and EAs) such as evolutionary strategies with covariance matrix adaptation (CMA-ES) [16], real-coded genetic algorithm (RCGA) [20], differential evolution (DE) [21] and so on, crossover operator often contributes greatly to their overall performance. So it is important to investigate why

crossover does not work for TSDOP, and how to manipulate them to make them be beneficial for shape design. Even if crossover is not adopted, the concept of consistent representation might also be useful for those EAs whose mutation performed based on the distribution of each objective parameter in current population. So the desirable evolution paradigm for TSDOP must include a mechanism to maintain such consistent representation explicitly or implicitly.

III. Cooperative Coevolution for TSDOP

A. Cooperative Coevolution

Cooperative Coevolution (CC) [14] was originally proposed as a general framework for applying EAs to large and complex problems using a divide-and-conquer strategy. In CC, the objective problem (such as an objective vector) is decomposed into smaller subproblems and each of them is assigned to a subpopulation. These subpopulations evolve mostly separately with cooperation happening only during the fitness evaluation phase. The three major steps of evolutionary optimization under the CC framework can be summarized as follows [22]:

1) **Problem Decomposition:** Divide the large or complex objective problem into smaller subcomponents.
2) **Subcomponent Optimization:** Evolve each subcomponent separately using a certain EA.
3) **Cooperative Combination:** Combine the solutions of all subcomponents to form the final solution.

Thanks to the CC framework, the applicability of conventional EAs has been advanced greatly for both numerical optimization problems and real-world applications. Potter applied CC to concept learning and neural network construction [14]. García-Pedrajas *et al.* proposed COVNET, which is a new CC model for evolving artificial neural networks [23]. In the domain of learning multiagent behaviors, CC represents a natural approach to applying traditional evolutionary computation [24], [25]. As for numerical optimization, grouping decomposition based CCEAs have been proposed in [22] to tackle large-scale problems up to 1000 dimensions.

B. Evolving B-spline with Cooperative Coevolution

The basic idea of Cooperative Coevolution (CC) for optimization is to divide the objective variables into a number of smaller subcomponents, and evolve each of them separately in a cyclical fashion. The feature makes it especially suitable for TSDOPs with B-spline as the geometry representation. Firstly, a natural decomposition exists in the B-spline representation, which means the coordinates of control points could form a subcomponent, and the knot vector could be regarded as another subcomponent. These two parts have quite different effects on the shape represented by B-spline. Moreover, as analyzed in Section II-C, they should not be optimized simultaneously using any EAs with crossover operators. CC provides a platform to evolve each of them separately while keeping the other constant. Secondly, with CC the knot vector has a chance to be evolved, rather than only be perturbed by knot insertion. This increases the probability of seeking for a suitable knot vector. Finally, CC for TSDOPs has a directly benefit in reducing the dimensionality of objective problem. Problem dimensionality is critical because the applicable number of fitness evaluations (FEs) for real-world design optimization is often very limited. EAs with a high-dimensional representation might fail to find a promising design within the given time budget. In CC for TSDOP, the representation is divided into two parts, and thus the problem dimensionality is reduced significantly. It might help EAs explore the search space more easily.

In addition to the natural decomposition method, another important issue in applying CC to TSDOP is to select an effective and efficient EA. Because CC is only a general framework, it cannot solve any problem without adopting a practical EA as

Algorithm 1: CMA-ES-CC

Input : μ, λ, gap, $nfes$, cp_g, kv_g
Output: $best_shape$

*initialize **init_shape** using B-spline*;
*formulate **kv_pop** using **knot vector** of **init_shape***;
*formulate **cp_pop** using **control points** of **init_shape***;
$best_shape = init_shape$;
$cp_rep = cp_pop.random()$;
$kv_rep = kv_pop.random()$;
$iter = 0$, $count = 0$, $flag = true$;
while *($nfes > 0$)* **do**
  if *($count >= gap$)* **then**
    *call **knot insertion** using **cp_pop** and **kv_rep***;
    *update **kv_pop** based on the extended **kv_rep***;
    *evaluate **cp_pop** using **kv_rep** as **knot vector***;
    $nfes = nfes - \mu$, $count = \mu$;
    $cp_rep = cp_pop.best()$;
  end
  if *(flag == true)* **then**
    *Evolve **cp_pop** once using **CMA-ES(μ, λ)***, with **kv_rep** as **knot vector** for evaluation ;
    $nfes = nfes - \lambda$;
    $count = count + \lambda$;
    $cp_rep = cp_pop.best()$;
    if *(++iter == cp_g)* **then**
      $iter = 0$, $flag = false$;
    end
  else
    *Evolve **kv_pop** once using **CMA-ES(μ, λ)***, with **cp_rep** as **control points** for evaluation;
    $nfes = nfes - \lambda$;
    $count = count + \lambda$;
    $kv_rep = kv_pop.best()$;
    if *(++iter == kv_g)* **then**
      $iter = 0$, $flag = true$;
    end
  end
  *formulate a **shape** based on **cp_rep** and **kv_rep***;
  if *(shape **better than** best_shape)* **then**
    $best_shape = shape$;
  end
end
return $best_shape$;

Notations:

μ, λ	– parameters of CMA-ES;
gap	– FEs gap for knot insertion;
$nfes$	– maximum number of fitness evaluations;
$iter$	– generation indicator;
kv_pop	– population to evolve knot vector;
cp_pop	– population to evolve control points;
kv_rep	– representative of kv_pop;
cp_rep	– representative of cp_pop;
kv_g	– generations to evolve kv_pop in a cycle;
cp_g	– generations to evolve cp_pop in a cycle;
$flag$	– switch the evolution for cp_pop or kv_pop.

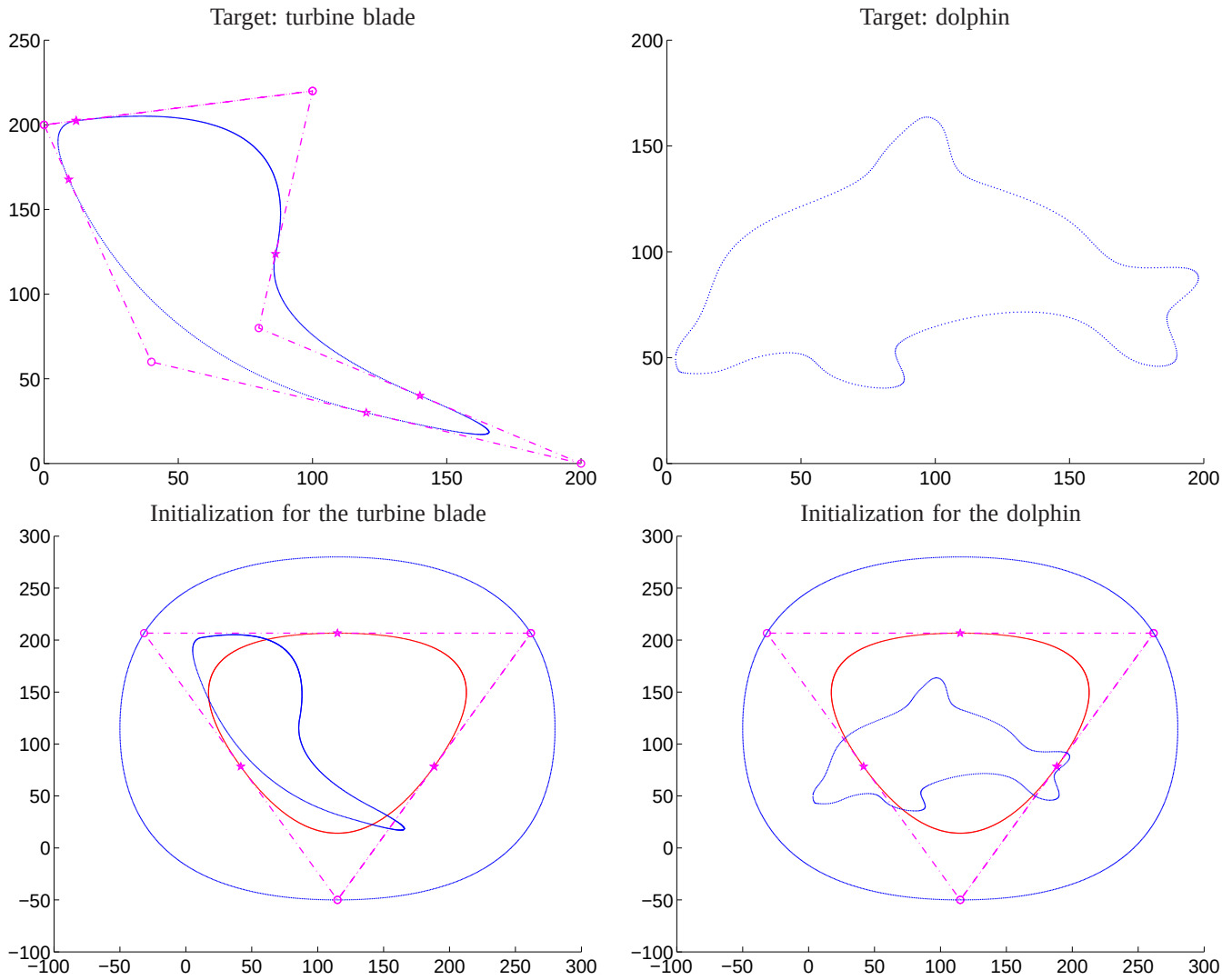

Fig. 2. The target shapes used in experimental studies, and the demonstration for initialization with 3 control points (the red circles).

basic optimizer. In this paper, we prefer CMA-ES [16] due to its good convergence features with even an extremely small population size (e.g. 2). With these basic components, the main steps of CC for TSDOPs can be summarized as follows:

1) Generate an initial shape, $shape$, using B-spline;
2) Generate populations, cp_pop and kv_pop, using the **control points** and **knot vector** of $shape$, respectively;
3) Evolve cp_pop for a predefined number of generations using CMA-ES(μ, λ);
4) Evolve kv_pop for a predefined number of generations using CMA-ES(μ, λ);
5) During the process of Steps (3) and (4), knot insertion will be performed in a fixed FEs (fitness evaluations) interval;
6) Stop if the halting criterion is satisfied; otherwise go to Step (3) for the next cycle.

The algorithm is denoted as CMA-ES-CC in this paper. The detailed pseudocode of the algorithm is provided as Algorithm 1. Notations are listed in the bottom of the code. There are only two newly introduced algorithm parameters, cp_g and kv_g, which are used to control how many generations will be assigned in each cycle to evolve the control points

and the knot vector, respectively.

C. Discussion on the Time Complexity of CMA-ES-CC

Time complexity is very important to evaluate the practicability of an algorithm. In the area of evolutionary computation, there are usually two ways to measure the time complexity of EAs. The first one is to give theoretical results, which is possible for only a few simplified EAs on some well-formulated problems [26]. The other one is to calculate the number of fitness evaluations used by EAs to solve the problem. It is based on the assumption that fitness evaluation is the most time expensive step when dealing with complex problems, and the computation time of other parts in EAs can be eliminated comparing with the time consumed by fitness evaluation. Since such an assumption is reasonable, it has been widely used in analyzing the time complexity of EAs on real-world applications.

As indicated in Section I, the fitness evaluation in shape design optimization is quite time consuming. So it is suitable to introduce the fitness evaluation calculation method to analyze the time complexity of the proposed CMA-ES-CC. Since CMA-ES-CC is generation-based algorithm, we first calculate the number of fitness evaluations used in each generation. According to Algorithm 1, the number of fitness evaluations used in each generated can be calculated as follows.

1) For the "if-else" block from pseudocode line 16 to 32, λ fitness evaluations are consumed in each generation.
2) From pseudocode line 9 to 15, the "if" condition is true in $\lceil \frac{gap}{\lambda} \rceil$ generations. Since μ fitness evaluations are consumed in the "if" block, there will be $\lceil \mu \frac{\lambda}{gap} \rceil$ fitness evaluations consumed in each generation.
3) As a result, the fitness evaluations used in each generation is $\lambda + \lceil \mu \frac{\lambda}{gap} \rceil$.

For generation-based algorithms, the number of generations is a user-determined parameter, which is usually set empirically based on the size of problems. If we assume the size of shape design optimization problems is n, and the number of generations allocated is kn, the total number of fitness evaluations used in each run will be $kn(\lambda + \lceil \mu \frac{\lambda}{gap} \rceil)$.

The time complexity of CMA-ES-CC can be calculated using the number of fitness evaluations times the time consumed in each fitness evaluation. Assuming the time complexity of each fitness evaluation is $O(T(n))$, the overall time complexity of CMA-ES-CC is $O(knT(n)(\lambda + \lceil \mu \frac{\lambda}{gap} \rceil))$. In particular, the number of sampled points in the target shape can be regarded as problem size for TSDOPs. Based on the fitness evaluation method given in Eq. (8), the time complexity of each fitness evaluation is $O(n^2)$. So overall time complexity of CMA-ES-CC is $O(kn^3(\lambda + \lceil \mu \frac{\lambda}{gap} \rceil))$ for TSDOPs.

IV. Experimental Studies

A. Target Shapes and Initialization

To evaluate the performance of CMA-ES-CC, two two-dimensional (2D) closed curves were used as the target shapes, which are shown in Fig. 2. The first one is a turbine blade built from B-spline with 5 control points and degree $p = 2$. It was utilized as a benchmark problem in [12], [28]. The bottom-right corner is often the most difficult part to fit for EAs. To be consistent with previous work, 200 points will be sampled from both the target shape and the designed shape for fitness evaluation. The other target is a 2D dolphin provided in the online benchmark repository [29]. It is represented in point cloud format with 500 points, and the coordinate ranges of sampled points are $x \in [0.0164, 0.99]$, $y \in [0.178, 0.8182]$. To make its search space consistent with the blade target, we added a scale factor 200 to the coordinates of all the sampled points. The target has been used for shape design in [30], and a competition of CEC2001. The fins and tails of the dolphin are the most difficult parts to fit. Since this target is more complex, all the 500 points will be used for fitness evaluation.

TABLE I
KEY FEATURES AND PARAMETER SETTINGS OF ALL THE IMPLEMENTED ALGORITHMS FOR COMPARISON.

Notation	Features and Parameter Setting
CMA-ES-CC	The algorithm proposed in this paper; Control points and knot vector were evolved in coevolutionary paradigm; CMA-ES with crossover and implemented according to [27]. Parameters: $\mu = 3$, $\lambda = 10$, $cp_g = 10$ and $kv_g = 5$.
CMA-ES	CMA-ES with crossover and implemented according to [27]; Control points and knot vector were evolved simultaneously in the same population. Parameters: $\mu = 3$ and $\lambda = 10$
iES[12]	Evolution Strategy with Individual Step Size [19], implemented the same as [12]; No crossover; Only control points were evolved. Parameters are set to the same as in [12]: $\mu = 2$ and $\lambda = 10$.
CMA-ES[12]	CMA-ES **without** crossover and implemented the same as [12]; Only control points were evolved. Parameters are set to the same as in [12]: $\mu = 2$ and $\lambda = 10$.
CMA-ES^c	CMA-ES with crossover and implemented according to [27]; Only control points were evolved. Parameters: $\mu = 3$ and $\lambda = 10$.
RCGA^c	Real-coded GA implemented according to [20]; PCX was used as crossover operators; Only control points were evolved. Parameters: $popsize = 10$, $\mu = 3$ and $\lambda = 2$.
DE^c	The classical Differential Evolution implemented according to [21]; Discrete crossover was used; Only control points were evolved. Parameters: $NP = 15$, $CR = 0.9$ and $F = 0.5$.

For initialization, we still follow the method used in [8], [12] and [28]. The basic requirement for the initial individuals is that no loop is permitted in the initial curve. Because, in principle, it takes extremely long time for optimization algorithms to remove loops in a looped curve, i.e., the solution searched starting from a looped curve tends to be looped [28]. In this study, the initial shape is a B-spline curve with degree $p = 2$, and 3 control points sampling from a given circle in the search space with equal knot span. The process of how the initial shape generated is demonstrated Fig. 2. Since adaptive encoding is used as representation scheme, the number of control points will grow gradually during evolution.

B. Setup

Based on the targets and initialization methods described in Section IV-A, experiments have been conducted to verify the efficacy of the proposed coevolutionary paradigm and the formulated algorithm CMA-ES-CC. Eq. (8) was used as the fitness function, and 200 and 500 points were sampled to calculate the fitness for the blade target and the dolphin target, respectively. To avoid the loops in the designed curve, a penalty factor, 2, was applied to the fitness value of curve including loops [12].

To analyze the performance of CMA-ES-CC, 6 different algorithms or configurations were implemented for comparison. Key features and detailed parameter settings of these algorithms are listed in Table I. Two of them, i.e. iES[12] and CMA-ES[12], are given in the same form as in [12], in order to provide a fair comparison with previous results. Note that we used the classical Differential Evolution (DE), although we were aware that a lot of adaptive and self-adaptive DE variants have been proposed [31], [32], [33]. The reason is based on two simple considerations. Firstly, most of these adaptation mechanisms of DE variants intend to adapt its parameters automatically against different types of benchmark problems, while in this paper we focus on only one application (i.e. TSDOPs). Secondly, most of these adaptation mechanisms are starting from a random parameter setting and then adjust it according to its searching experience, but for design optimization the algorithms do not really have much time to "learn" because the number of fitness evaluations (FEs) is often very limited.

For both of the considerations, an empirical parameter setting is expected to work better than self-adaptive ones. All the tested algorithms were implemented based on the publicly available NURBS++ library [34] and SHARK library [35]. The CMA-ES variant we used for CMA-ES-CC as internal optimizer was proposed in [27]. The code was provided by its authors as part of the SHARK library.

For computational budget, as recommended in [12], [28], a number of 5,000 FEs was assigned to each algorithm. For the first 70% of the FEs, knot insertion was used according to adaptive encoding in [8] to expand the representation every 300 FEs for the blade target, and 150 FEs for the dolphin target. The representation was kept constant for the algorithms to converge within the last 30% of the FEs.

C. Results

TABLE II

EXPERIMENTAL RESULTS OF ALL IMPLEMENTED ALGORITHMS OVER 30 INDEPENDENT RUNS. THE BEST RESULTS OF EACH ROW ARE GIVEN IN **BOLD FACE**. THE "†" INDICATES THAT THE RESULT IS SIGNIFICANT BETTER THAN ALL OTHERS BY WILCOXON RANK SUM TEST WITH $p < 0.001$.

Blade Target							
	CMA-ES-CC	CMA-ES	iES[12]	CMA-ES[12]	CMA-ES^c	RCGA^c	DE^c
Best	**80.39**	167.65	247.60	214.50	102.18	198.25	151.85
Median	**129.13**	214.59	-	-	177.70	237.26	269.78
Mean	**132.78**†	216.50	659.00	739.20	178.78	252.13	812.76
Worst	**228.08**	283.01	-	-	232.53	366.89	5217.56
Std	31.12	**29.34**	552.9	721.00	30.99	46.82	1124.88
Dolphin Target							
	CMA-ES-CC	CMA-ES	iES[12]	CMA-ES[12]	CMA-ES^c	RCGA^c	DE^c
Best	**116.92**	221.12	489.24	242.55	123.13	237.86	1359.72
Median	**172.53**	962.78	1417.19	591.08	549.49	564.62	5146.46
Mean	**224.28**†	1654.53	1931.09	900.14	767.88	1032.57	5963.86
Worst	**627.73**	10131.42	5581.60	3236.71	2906.37	6936.84	16295.87
Std	**129.07**	2384.77	1337.82	697.42	719.51	1320.91	3839.15

The results for all the tested algorithms were obtained by 30 independent runs. The best designed shapes, and the distributions of designed shapes in all the 30 independent runs using CMA-ES-CC were shown in Fig. 3. The detailed statistical results (including best value, median value, mean value, worst value and standard deviation) of all the tested algorithms are presented in Table II. Note that the results of iES[12] and CMA-ES[12] for the blade target were taken from [12] directly. For the dolphin target, the results of them were carried out by ourselves simply because this target was not investigated previously.

From Fig. 3, it can be found that the best designed shapes fit the targets very well. To be specific, the best designed blade is exactly the same as the target blade. For the more complex dolphin target, there is only very slight difference in the top fin of the dolphin between the designed and the target shapes. For the distribution of all the 30 independent runs, it is clear that all the sampled points from designed shapes are just around the targets closely. The points distributed in the fins and tails of the dolphin are a little scattered, which confirms the fact that these parts are most difficult to fit in practice. CMA-ES-CC was able to fit these parts accurately in some of the 30 runs, which can be seen from the best designed dolphins.

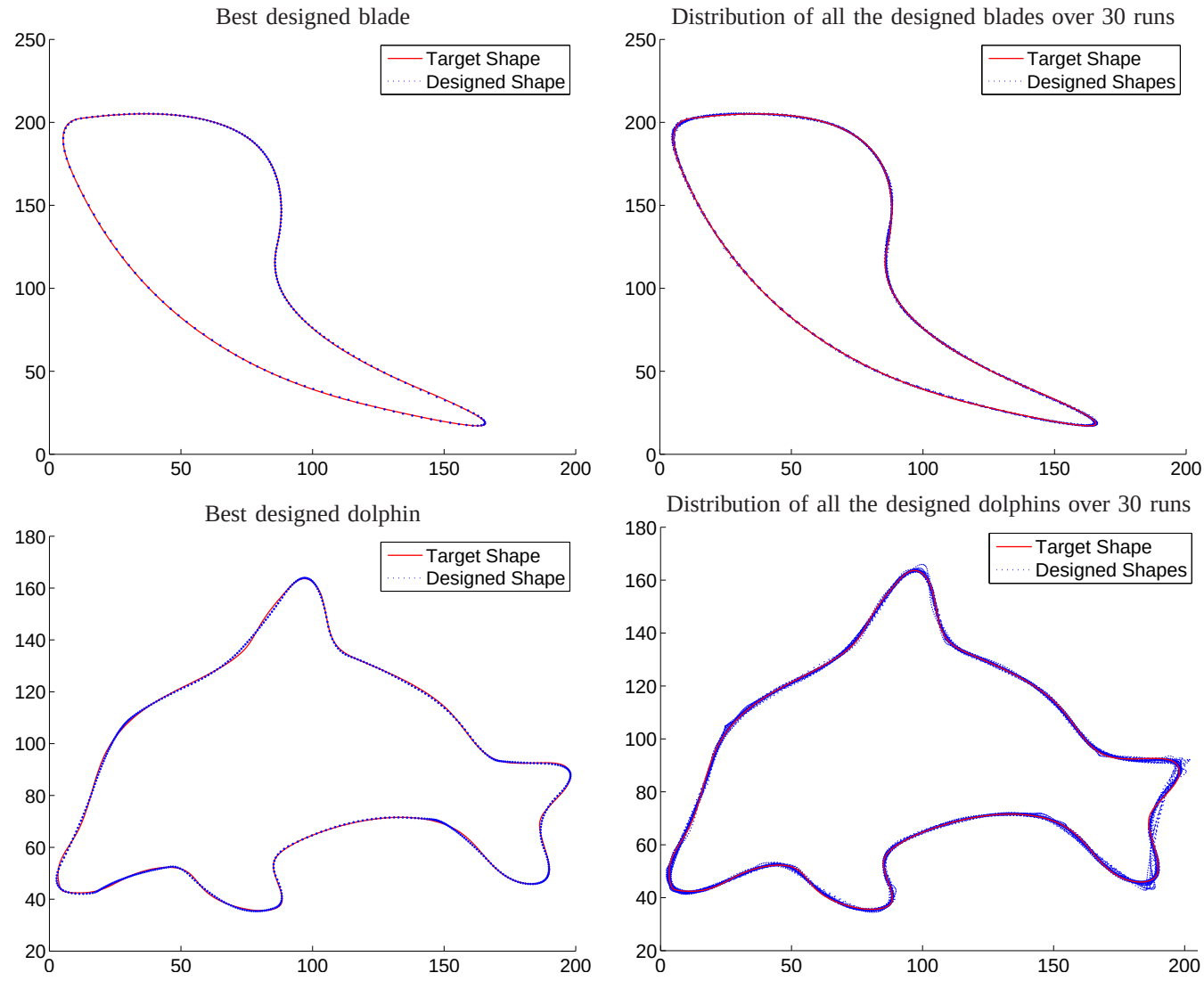

Fig. 3. The best designed shapes (left figures), and the distributions of designed shapes (right figures) in all the 30 independent runs by CMA-ES-CC.

From the detailed results given in Table II, it can be concluded that CMA-ES-CC is the best one among all the tested algorithms. The results of Wilcoxon rank sum test indicate the advantages of CMA-ES-CC are significant. Comparing to the results of iES[12] and CMA-ES[12] reported in [12], CMA-ES-CC is the clear winner in all aspects. Even the mean values of CMA-ES-CC are still better than the best values of previous results. This means the performance of CMA-ES-CC is not only good, but also stable. Comparing to algorithms CMA-ES^c, RCGA^c and DE^c, which evolve only control points, CMA-ES-CC also showed the overwhelming superiority. CMA-ES^c achieved the most closed results with CMA-ES-CC, which might be because they utilized the same basic optimizer. Although CMA-ES, which evolves the control points and knot vector simultaneously, showed slightly smaller standard deviation that CMA-ES-CC on the blade shape, the situation is reversed on the more complex dolphin target. Among all the tested algorithms, CMA-ES-CC showed the most stable performance from the simpler blade target to the more complex dolphin targets. These comparisons indicate that the proposed cooperative coevolutionary paradigm and the formulated algorithm CMA-ES-CC are very effective for TSDOPs.

The evolution curves for control point growth and fitness variation of the best three algorithms were also shown in Fig. 4. Since the same adaptive encoding scheme was used for all algorithms, there is not much difference among the three control point growth curves. It is interesting to find that in the CMA-ES-CC version there was a clear step growth curve in optimizing the dolphin target. That is because CMA-ES-CC is the most stable algorithm for this target, which implies it has very similar behaviors in all the tested 30 runs. As for the fitness variation curves, although the bias to control points optimization made CMA-ES and CMA-ES^c converge very quickly at the beginning, they slowed down soon after that, which might be because they lack the reasonable balance between optimizing the control points and the knot vector. In contrast, CMA-ES-CC was able to make good progress stably, and finally it found much better solutions than others.

Fig. 5 is provided to analyze how the optimization of control points and knot vector in CMA-ES-CC contributed to its overall performance. First of all, it is easy to observe that both the optimization of control points and knot vector were able to reduce the fitness values, which means they both have made contributions to the refinement of the designed curves. It is also notable that the optimization of control points is generally more effective than the optimization of knot vector. This might explain why some conventional methods still have comparable performance by optimizing only the control points. However, there exist some situations, which can seen from the black circles in Fig. 5, where great contributions were achieved in optimizing the control points after a cycle of knot vector optimization. This confirms the fact that the variation of control points and knot vector is highly correlated. A good selected knot vector might be also helpful for the subsequent optimization of control points. Without proper manipulation of knot vector, the optimization of control points could be stuck or even stagnated (imagining we have a very bad knot vector). Rather than ignoring the necessary freedom of knot vector variation, which is true in some conventional methods for design optimization, CMA-ES-CC provided an option to optimize both the control points and knot vector in a new cooperative coevolution paradigm.

V. Conclusions and Future Work

In this paper, we investigated how to effectively tackle the target shape design optimization problems (TSDOPs) with B-spline as the geometry representation. Based on the analysis of the relation between B-spline control points and knot vector, we proposed a new cooperative coevolution (CC) paradigm to evolve both of them separately. Comparing to straightforward methods that evolve the control points and knot vector simultaneously, it provides a platform to evolve these components of B-spline in a reasonable cooperated fashion. Comparing to other methods that evolve only the control points, it could increase the freedom of knot vector greatly, which is desired for a better exploration in the search space. Moreover, the difficulty of TSDOPs is reduced with the natural decomposition of control points and knot vector, while might be also helpful for EAs exploring the search space easier in a tight time budget.

To evaluate the performance of the new coevolutionary paradigm, an algorithm named CMA-ES-CC was formulated. Experiments were conducted based on two target shapes. It was found that the performance of CMA-ES-CC was not only good, but also stable. The comparison with six other EAs for TSDOPs suggested that the results of CMA-ES-CC were significantly better.

One direction of future work is to study coevolutionary optimization for three-dimensional (3D) shape design problems. The most difficult challenge is that existing fitness functions based on Hausdorff distance often fail to evaluate 3D shapes correctly, because they are not able to reflect whether the designed shape is reasonably smooth. Another interesting topic is to apply the proposed algorithm to real-world design optimization problems using Computational Fluid Dynamics (CFD)

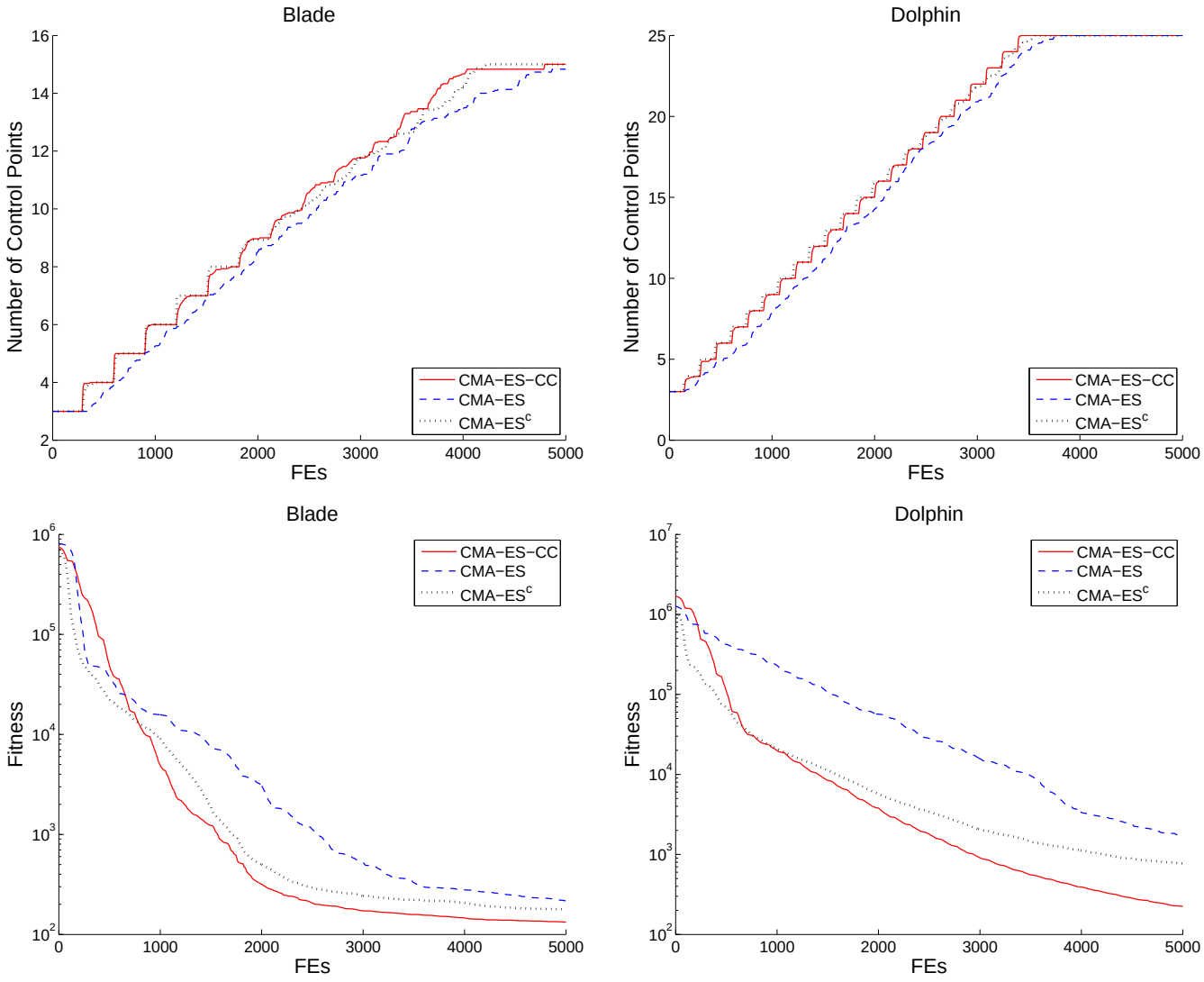

Fig. 4. The evolution curves for control point growth and fitness variation of the best three algorithms. Results have been averaged over 30 independent runs.

tools, which would provide a more accurate test bed. The smoothness requirement in 3D shape design could also be satisfied if we use CFD tools for fitness evaluation.

REFERENCES

[1] J. Sobieszczanski-Sobieski and R. T. Haftka, "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," *Structural and Multidisciplinary Optimization*, vol. 14, no. 1, pp. 1–23, 1997.

[2] R. Kicinger, T. Arciszewski, and K. De Jong, "Evolutionary Computation and Structural Design: A Survey of the state-of-the-art," *Computers and Structures*, vol. 83, no. 23-24, pp. 1943–1978, 2005.

[3] B. Sendhoff and Y. Jin, "A systems approach to evolutionary multi-objective structural optimization," *IEEE Computational Intelligence Magazine*, vol. 4, no. 3, pp. 62–76, 2009.

[4] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with computers*, vol. 29, no. 1, pp. 17–35, 2013.

[5] M. Olhofer, T. Arima, T. Sonoda, and B. Sendhoff, "Optimisation of a Stator Blade used in Transonic Compressor Cascade with Evolution Strategies," in *Adaptive Computing in Design and Manufacture*, I. Parmee, Ed. Springer-Verlag, 2000, pp. 45–54.

[6] T. Sonoda, Y. Yamaguchi, T. Arima, M. Olhofer, B. Sendhoff, and H. A. Schreiber, "Advanced High Turning Compressor Airfoils for Low Reynolds Number Condition – Part I: Design and Optimization," *Journal of Turbomachinery*, vol. 126, pp. 350–359, 2004.

[7] K. Giannakoglou, D. Papadimitriou, and I. Kampolis, "Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 44, pp. 6312–6329, 2006.

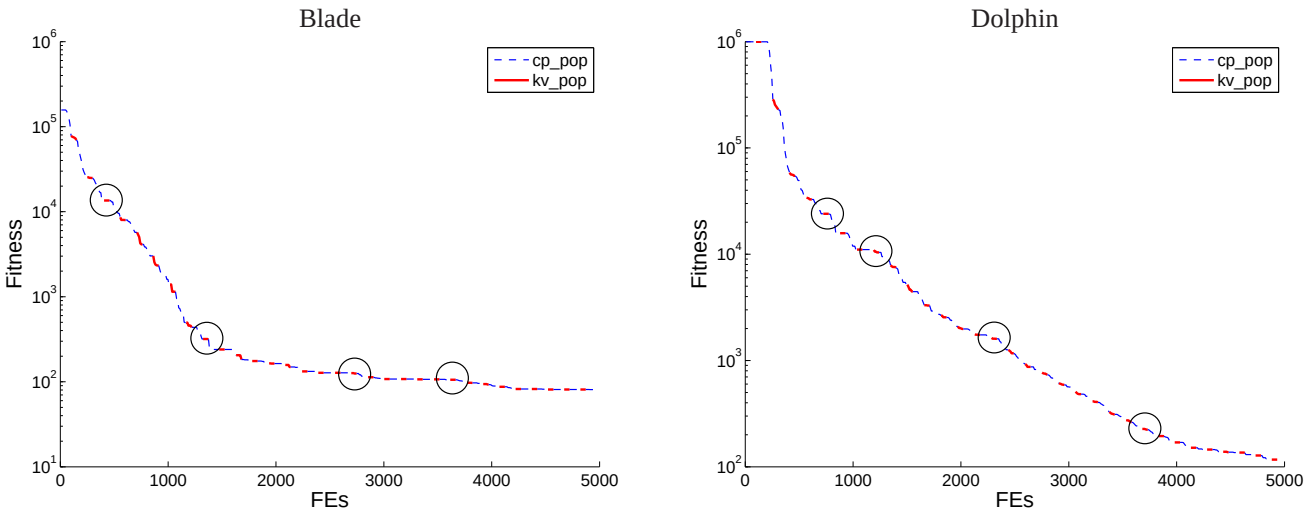

Fig. 5. The contributions of CMA-ES-CC in optimizing the control points (cp_pop) and knot vector (kv_pop) for the best designed blade and dolphin curves. The black circles indicate the places where great contributions were achieved in optimizing the control points after a cycle of knot vector optimization.

[8] M. Olhofer, Y. Jin, and B. Sendhoff, "Adaptive Encoding for Aerodynamic Shape Optimization Using Evolution Strategies," in *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, 2001, pp. 576–582.

[9] M. S. Khan, A. F. Ayob, A. Isaacs, and T. Ray, "A novel evolutionary approach for 2D shape matching based on B-spline modeling," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2011, pp. 655–661.

[10] W. Chang, C. Chung, and B. Sendhoff, "Target Shape Design Optimization with Evolutionary Computation," in *Proceedings of the 2003 IEEE Congress on Evolutionary Computation*, 2003, pp. 1864–1870.

[11] M. Nashvili, M. Olhofer, and B. Sendhoff, "Morphing Methods in Evolutionary Design Optimization," in *Proceedings of the 2005 Genetic and Evolutionary Computation Conference*. ACM New York, NY, USA, 2005, pp. 897–904.

[12] P. Zhang, X. Yao, L. Jia, B. Sendhoff, and T. Schnier, "Target Shape Design Optimization by Evolving Splines," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, 2007, pp. 2009–2016.

[13] L. Piegl and W. Tiller, *The NURBS Book*. Springer-Verlag, 1997.

[14] M. Potter, "The Design and Analysis of a Computational Model of Cooperative Coevolution," Ph.D. dissertation, George Mason University, 1997.

[15] Y. Liu, X. Yao, Q. Zhao, and T. Higuchi, "Scaling up Fast Evolutionary Programming with Cooperative Coevolution," in *Proceedings of the 2001 Congress on Evolutionary Computation*, 2001, pp. 1101–1108.

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

[17] J. A. Samareh, "A Survey of Shape Parameterization Techniques," in *NASA Conference Publication*, 1999, pp. 333–344.

[18] Wikipedia, "Nurbs — Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/NURBS, [Online; accessed June 22, 2009].

[19] H. Schwefel, *Evolution and Optimum Seeking*. John Wiley & Sons, 1995.

[20] K. Deb, A. Anand, and D. Joshi, "A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization," *Evolutionary Computation*, vol. 10, no. 4, pp. 371–395, 2002.

[21] R. Storn and K. Price, "Differential Evolution–A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[22] Z. Yang, K. Tang, and X. Yao, "Large Scale Evolutionary Optimization Using Cooperative Coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985–2999, 2008.

[23] N. García-Pedrajas, C. Hervás-Martínez, and J. Muñoz Pérez, "COVNET: a cooperative coevolutionary model for evolving artificial neural networks," *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 575–596, 2003.

[24] L. Panait and S. Luke, "Cooperative Multi-Agent Learning: The State of the Art," *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pp. 387–434, 2005.

[25] L. Panait, S. Luke, and R. Wiegand, "Biasing Coevolutionary Search for Optimal Multiagent Behaviors," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 629–645, 2006.

[26] J. He and X. Yao, "Drift analsis and average time complexity of evolutionary algorithms," *Artificial Intelligence*, vol. 127, no. 1, pp. 57–85, 2001.

[27] N. Hansen and S. Kern, "Evaluating the CMA Evolution Strategy on Multimodal Test Functions," in *Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, 2004, pp. 282–291.

[28] X. Yao, T. Schnier, and P. Zhang, "Dynamic Optimization with Adaptive Representations – Final Project Report," The Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA), University of Birmingham, Tech. Rep., 2007.

[29] B. Sendhoff, M. Roberts, and X. Yao, "Evolutionary Computation Benchmarking Repository," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 50–60, 2006. http://www.cs.bham.ac.uk/research/projects/ecb/.

[30] T. Yates, "Adaptive Encoding for Evolutionary Shape Optimisation – Summer Project Report," The Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA), University of Birmingham, Tech. Rep., 2007.

[31] Z. Yang, K. Tang, and X. Yao, "Self-adaptive Differential Evolution with Neighborhood Search," in *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*, 2008, pp. 1110–1116.

[32] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[33] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 482–500, 2012.
[34] P. Lavoie, "The NURBS++ package," http://libnurbs.sourceforge.net/index.shtml, 2002.
[35] C. Igel, T. Glasmachers, and V. Heidrich-Meisner, "SHARK," *Journal of Machine Learning Research*, vol. 9, pp. 993–996, 2008.

LaTeX Source Files

Click here to download LaTeX Source Files: texsource.rar