

Data-driven evolutionary sampling optimization for expensive problems

ZHEN Huixiang¹, GONG Wenyin^{1,*}, and WANG Ling²

1. School of Computer Science, China University of Geosciences, Wuhan 430074, China;

2. Department of Automation, Tsinghua University, Beijing 100084, China

Abstract: Surrogate models have shown to be effective in assisting evolutionary algorithms (EAs) for solving computationally expensive complex optimization problems. However, the effectiveness of the existing surrogate-assisted evolutionary algorithms still needs to be improved. A data-driven evolutionary sampling optimization (DESO) framework is proposed, where at each generation it randomly employs one of two evolutionary sampling strategies, surrogate screening and surrogate local search based on historical data, to effectively balance global and local search. In DESO, the radial basis function (RBF) is used as the surrogate model in the sampling strategy, and different degrees of the evolutionary process are used to sample candidate points. The sampled points by sampling strategies are evaluated, and then added into the database for the updating surrogate model and population in the next sampling. To get the insight of DESO, extensive experiments and analysis of DESO have been performed. The proposed algorithm presents superior computational efficiency and robustness compared with five state-of-the-art algorithms on benchmark problems from 20 to 200 dimensions. Besides, DESO is applied to an airfoil design problem to show its effectiveness.

Keywords: evolutionary algorithm (EA), surrogate model, data-driven, evolutionary sampling, airfoil design.

DOI: 10.23919/JSEE.2021.000027

1. Introduction

Evolutionary algorithms (EAs) or meta-heuristic algorithms (MAs) have become a popular tool for optimization due to their promising global search ability. However, EAs require a large number of function evaluations during the optimization process, which limits their application in many real-world problems [1]. In real world, there

are many computationally expensive numerical simulations and costly experiments, such as computational fluid dynamics (CFD) [2], computational structural mechanics (CSM) [3], and finite element analysis (FEA) [4]. Solving the computationally expensive problems has become popular in science and engineering. Some of them, such as the computation of the objective function and constraints, are extremely expensive, e.g., single CFD simulation takes from minutes to hours [5]. Such expensive function evaluations (FEs) cannot be afforded by EAs, which typically require tens of thousands of FEs to obtain the global optimum.

Surrogate models (also known as meta-models [6]) have been widely used in EAs to reduce the computational time. The principle of surrogate-assisted EAs (SAEAs) is that the time cost of surrogate models for predicting the objective function is insignificant compared with expensive function evaluations [7]. Jin et al. pointed out that surrogates can be applied to almost all operations of EAs [8], consisting of population initialization, crossover, mutation, local search, fitness evaluation and so on [9–11]. A variety of machine learning models have been used to construct surrogates, such as polynomial regression (PR) [12,13], Gaussian processes (GP) [14–16] (also known as Kriging [17,18]), radial basis function (RBF) [19], artificial neural networks (ANNs) [20–22], and support vector machine (SVM) [23,24]. Moreover, ensemble of surrogates is also used to enhance the performance of SAEAs [25,26].

With limited computational budgets, researchers have conducted detailed studies on ways of employing surrogate models [7,8]. The choice of a new sampled point for evaluation is significant for updating the surrogate model and guiding the optimization process. The sampled points, which have good fitness values or high-uncertainty, are promising for fitness evaluation. The sampled points with good fitness values can accelerate the convergence, and the points with high-uncertainty can reduce the risk of

Manuscript received September 01, 2020.

*Corresponding author.

This work was supported by the National Natural Science Foundation of China (62076225; 62073300), the Natural Science Foundation for Distinguished Young Scholars of Hubei (2019CFA081), and the Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan) (CUGGC03).

falling into local minima. Generally, uncertainty information can be estimated according to the variance provided by the Kriging model [27], the distance to the evaluated sample points [28], and the discrepancies among the predictions of the ensemble surrogates [29]. Jones et al. [27] developed the efficient global optimization (EGO) method, where the Kriging model was used to approximate the exact function and the estimated mean square error was used to reflect the prediction variance. Several Kriging-based infill criteria, such as lower confidence bound (LCB) [15], expected improvement (EI) [30], and probability of improvement (PoI) [31], have shown promising performance for low-dimensional optimization problems with fewer than 15 dimensions. However, due to the curse of dimensionality, when dealing with high-dimensional expensive optimization problems, the performance of Kriging-based SAEAs becomes noticeably worse because it is difficult to build an accurate surrogate model in high-dimensional space. On the other hand, RBF-based surrogates are employed widely in high-dimensional expensive optimization [32,33]. Kriging and RBF are two widely studied surrogate models. Training of the Kriging model is time-consuming when the number of samples or dimensions is large [34]. RBF also is a popular modeling method with high efficiency and accuracy. RBF is more suitable for high-dimensional expensive optimization problems because it works well to approximate high-dimensional nonlinear functions and the training time required for RBF is insensitive to the increase of dimensions.

Various studies indicate that the ability to guide optimization is more worthy of attention than the accuracy of models when facing high-dimensional problems [8,32,33]. The surrogate-assisted strategies can be classified into two main categories, surrogate-assisted prescreening and surrogate-assisted local search. Surrogate-assisted prescreening uses the surrogate's response function to sort the candidate offspring points produced by the basic EAs and then select the points with small predicted responses as the offspring points. Sun et al. used RBF to sort the candidate offspring particles and then screen some potential offspring particles with small response values in the RBF-assisted social learning particle swarm optimization (SLPSO) [35]. Li et al. proposed a prescreening criterion based on the prediction difference of multiple surrogates [11]. Surrogate-assisted local search is also widely used in SAEAs, which searches promising sample points in the local area based on predicted fitness value by surrogate. Ong et al. employed a trust-region method for the interleaved use of exact models for the objective and constrained functions with computationally cheap RBF surrogates during the local search [36]. Wang et al. built a local RBF model for searching one candidate point by the

differential evolution (DE) optimizer [32]. Cai et al. used the surrogate-based trust region local search method to guide the genetic algorithm to search in an accurate way [34].

Recently, some considerable progress has been made on improving the SAEAs search schemes. Liu et al. presented a Gaussian process surrogate model assisted evolutionary algorithm for medium-scale computationally expensive optimization problems (GPEME) (20-50 decision variables), in which a Gaussian process model with LCB prescreen solutions in a differential evolution algorithm and a dimensional reduction technique was proposed to enhance the accuracy of the GP model [15]. Sun et al. proposed a surrogate-assisted cooperative swarm optimization (SA-COSO) algorithm, in which a surrogate-assisted particle swarm optimization (PSO) algorithm and a surrogate-assisted social learning based PSO algorithm cooperatively search for the global optimum [35]. Yu et al. proposed a surrogate-assisted hierarchical PSO (SHPSO) algorithm, which builds a local RBF model with a certain number of current best samples [33]. On one hand, the optimum of the RBF model is searched as a candidate point by SLPSO. On the other hand, the RBF is used to prescreen out some promising particles whose estimated values are smaller than their personal bests. Wang et al. proposed a novel evolutionary sampling assisted optimization (ESAO) method which consists of two major search strategies, in which a global RBF model is built to choose the best offspring generated by evolutionary operators (mutation and crossover), the other local RBF model is built for searching one candidate point by the DE optimizer, and the two strategies are employed alternately in optimization [32].

In order to improve the effectiveness of the SAEAs. In this paper, we present a data-driven evolutionary sampling optimization (DESO) framework for expensive optimization problems. DESO considers both the surrogate screening (SS) and surrogate local search (SLS) as two different degrees of evolutionary sampling strategies. RBF models are employed as surrogates in the two strategies because of its high accuracy and fast modeling for high dimension problems. SS is used as a global search to find new promising points over the large area. SLS is used to exploit the most promising area. To balance exploration and exploitation and avoid falling into local optima, DESO randomly employs SS or SLS to obtain a new sample point for updating the surrogate model and guiding the optimization process. In each generation, population and surrogate models are updated based on a predefined number of best sample points from database.

The main contributions of the paper are as follows:

- (i) Two data-driven evolutionary sampling strategies,

SS based on evolutionary operators and SLS based on the evolutionary optimizer, are proposed to sample promising points. They are used for the global search and local search, respectively.

(ii) The proposed method uses one of the two evolutionary sampling strategies randomly in each iteration to balance exploration and exploitation. Moreover, adaptive parameters are set for different dimensional problems.

(iii) The proposed method shows superiority compared to five state-of-the-art algorithms on 30 test functions and an airfoil design problem.

The remainder of this paper is organized as follows. Section 2 introduces the related techniques, such as DE and RBF networks. Section 3 details the data-driven evolutionary sampling optimization framework. Subsequently, experimental results and analysis are conducted in Section 4. Finally, Section 5 concludes this paper and discusses the possible future work.

2. Related techniques

2.1 DE

DE, proposed by Storn and Price [37], is a very popular EA, which contains four main operations: initialization, mutation, crossover, and selection. DE is widely used in various scientific and engineering optimization problems [38]. Researchers have proposed different variants of the basic algorithm to improve its performance [39–41]. In this work, DE is employed as the optimizer for SLS, and its mutation and crossover operations are also used to generate sample points for SS.

Suppose that a population with n individuals is expressed as $\mathbf{P} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$. The i th individual with d dimensions is $\mathbf{x}_i = [x_i^1, x_i^2, \dots, x_i^d]$ in \mathbf{P} . The mutation and crossover are briefly introduced as follows.

In mutation, the mutant vector $\mathbf{v}_i = [v_i^1, v_i^2, \dots, v_i^d]$ can be obtained by the mutation operator. The commonly used mutation operators are as follows.

(i) DE/rand/1

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (1)$$

(ii) DE/best/1

$$\mathbf{v}_i = \mathbf{x}_{\text{best}} + F(\mathbf{x}_{r_1} - \mathbf{x}_{r_2}) \quad (2)$$

(iii) DE/current-to-rand/1

$$\mathbf{v}_i = \mathbf{x}_i + F(\mathbf{x}_{r_1} - \mathbf{x}_i) + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (3)$$

(iv) DE/current-to-best/1

$$\mathbf{v}_i = \mathbf{x}_i + F(\mathbf{x}_{\text{best}} - \mathbf{x}_i) + F(\mathbf{x}_{r_1} - \mathbf{x}_{r_2}) \quad (4)$$

where \mathbf{x}_{r_1} , \mathbf{x}_{r_2} , and \mathbf{x}_{r_3} are three mutually different individuals chosen from the current population, and F is a

scaling factor which typically lies in the interval [0.4, 1]. DE/rand/1 is selected in this work.

In crossover, the trial vector, expressed as $\mathbf{u}_i = [u_i^1, u_i^2, \dots, u_i^d]$, is generated from \mathbf{x}_i and \mathbf{v}_i . There are two kinds of crossover methods: exponential and binomial method. In this work, we just introduce the binomial, whose formula can be expressed as

$$u_i^j = \begin{cases} v_i^j & \text{rand}_i^j \in [0, 1] \leq Cr \text{ or } j = j_{\text{rand}} \\ x_i^j & \text{otherwise} \end{cases} \quad (5)$$

where u_i^j is the j th component of the i th offspring. x_i^j and v_i^j are the j th component of the i th parent and mutated individual. Cr is a constant between 0 and 1. The rand_i^j indicates a uniformly distributed random number. j_{rand} is a randomly chosen index that ensures u_i has at least one component of v_i . More introduction of DE can refer to [42, 43]. Other EAs can also be used as the optimizer in SLS of DESO, such as PSO [44], Jingqiao's adaptive differential evolution (JADE) [39], and SLPSO [45], etc.

2.2 RBF network

RBF is widely used as the surrogate model for high-dimensional expensive problems ($n > 30$) [34]. Given the sampled points $\{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbf{R}^d, i = 1, 2, \dots, N\}$, RBF uses a weighted sum of basis-functions to approximate the fitness function landscape as follows:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^N \omega_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) \quad (6)$$

where $\|\cdot\|$ is the Euclidian norm, ω_i is the weight coefficient and $\phi(\cdot)$ is a kernel function. The commonly used radial basis kernels are the Gaussian function, thin-plate splines, linear splines, cubic splines, and multi-quadrics splines [46]. The Gaussian kernel function $\phi(x) = \exp(-x^2/\beta)$ is chosen in this paper, where β denotes the shape parameter. In addition, the weight vector $\boldsymbol{\omega} = [\omega_1, \omega_2, \dots, \omega_N]^T \mathbf{y}$ can be computed as follows [33]:

$$\boldsymbol{\omega} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{y} \quad (7)$$

where $\mathbf{y} = [y_1, y_2, \dots, y_N]^T$ is the output vector and $\boldsymbol{\Phi}$ is the following matrix:

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi(\mathbf{x}_1 - \mathbf{x}_1) & \cdots & \phi(\mathbf{x}_1 - \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_N - \mathbf{x}_1) & \cdots & \phi(\mathbf{x}_N - \mathbf{x}_N) \end{bmatrix}. \quad (8)$$

The shape parameter of the Gaussian kernel function, referred to [33], is set to $\beta = D_{\max}(dN)^{-1/d}$, where D_{\max} is the maximal distance between the training data. Once parameter β and the training samples are obtained, the RBF model can be constructed efficiently.

3. DESO

In this section, DESO is proposed, which consists of two evolutionary sampling strategies: SS for global search and SLS for surrogate local search. They can be seen as sampling methods with different evolutionary degrees, because SS contains once evolutionary operators and SLS finds one candidate by the whole evolution optimization process. DESO attaches importance to historical sample points very much. In each generation, evolutionary sampling strategies obtain a sample point for evaluation, then the evaluated sample point is added to the database, until the budget of fitness evaluation uses up. Sample points with good performance guide DESO to search promising areas by building RBF models and constituting population. In this paper, the evolutionary sampling process employs randomly one of the two sampling strategies, SS based on the evolutionary operators and SLS based on the evolutionary optimizer, to balance exploration and exploitation.

Algorithm 1 Pseudo code of the proposed DESO.

```

1 Initialize: Generate  $N_0$  initial samples  $\{\mathbf{x}_k, k = 1, 2, \dots, N_0\}$ , by Latin hypercube design, calculate their original objective function value  $f(\mathbf{x}_k)$ , and initialize the dataset:  $\mathbf{D} = \{(\mathbf{x}_k, f(\mathbf{x}_k)), k = 1, 2, \dots, N_0\}$ .
2 FEs =  $N_0$ ;
3 while FEs < FEmax do
4  $\mathbf{D}_{\text{sort}} = \text{sort}(\mathbf{D})$ ;
5 Randomly set  $S$  to 1 or 2 with same probability
// Evolutionary sampling
6 if  $S = 1$  then
//Strategy 1: SS based on evolutionary operators
7 Selet  $N$  best samples from  $\mathbf{D}_{\text{sort}}$  as population  $\mathbf{P}$ ;
8 Build the RBF model  $\hat{f}_s$  by  $m$  best samples;
9 Generate offspring  $\mathbf{U}$  according to (1) and (5);
10 Obtain the predictions of all offspring using  $\hat{f}_s$ ;
11 Select the most promising candidate solution  $\mathbf{x}_c$  according to predictions;
12 else
//Strategy 2: surrogate searching based on evolutionary optimizer
13 Selet  $\tau$  best samples from  $\mathbf{D}_{\text{sort}}$  as sample set  $\mathbf{s}$ ;
14 Set search area according to upper bound  $\mathbf{u}$  and lower bound  $\mathbf{l}$  of all samples in  $\mathbf{s}$ ;
15 Build the RBF model  $\hat{f}_l$  by  $\tau$  best samples;
16 Obtain the candidate solution  $\mathbf{x}_c$  by minimizing  $\hat{f}_l$  using evolutionary optimizer;
17 end if
18 Evaluate fitness  $f(\mathbf{x}_c)$  of the candidate solution;
19 Add the evaluated sample point into database  $\mathbf{D}$ ;
20 FEs = FEs + 1
21 end while

```

3.1 SS based on evolutionary operators

The pseudo code of SS based on evolutionary operators is shown in lines 7 to 11 of Algorithm 1. The strategy selects the best N samples in database to form population, and build an RBF model \hat{f}_s using the best m samples. In early optimization, all sample points in the database are used to train the RBF model. When the database is big enough, we set m to a fixed value avoiding too much training time. In this work, we set a maximum of m to 300. Then, the population generates the same number of offspring by evolutionary operators. Different EAs operators can be used here, such as DE, PSO, and GA. We employ DE/rand/1 mutation and crossover. The candidate point \mathbf{x}_c with the lowest prediction value will be evaluated by using the expensive fitness function. Finally, the evaluated sample point is added into the database. When the evaluated sample point is better than the worst of the sample in population, the new sample will be selected to form population and then helps guide the optimization process.

3.2 SLS based on evolutionary optimizer

The surrogate local search strategy aims to find the optimum of the local surrogate model to accelerate the convergence of the optimization process. The pseudo code can be seen in lines 13 to 16 of Algorithm 1. SLS firstly selects τ best samples from the database as a sample set \mathbf{s} for local search. The local surrogate model is built by training the RBF model with the τ sample points. Different from the SS strategy, the SLS strategy builds a surrogate model using a small number of sample points. Moreover, in order to solve problems of different dimensions, τ is set to $25+d$ and its maximum is 60, where d is the dimension of the problem. Search area $[\mathbf{l}, \mathbf{u}]$ is also determined by the sample set referring to (9) and (10). For each variable, we set the upper bound and the lower bound of the sample set as search ranges of the local search.

$$\begin{cases} \mathbf{l} = [l_1, l_2, \dots, l_d] \\ \mathbf{u} = [u_1, u_2, \dots, u_d] \end{cases} \quad (9)$$

$$\begin{cases} l_j = \min x_j^i \\ u_j = \max x_j^i \end{cases} \quad (10)$$

where $i = 1, 2, \dots, \tau$.

Then the evolutionary optimizer can search one candidate sample point by minimizing \hat{f}_l under preset stopping criteria. It should be noted that the stop criterion of the optimizer does not need to be set very strict since the surrogate model has error. The evolutionary optimizer

can be DE, PSO, and GA. For researching the performance of evolutionary optimizers, we will test different evolutionary optimizers in Section 4. After fitness evaluation, the new sample point is added into the database. Same as the surrogate screening strategy, the new sample point with good performance will help for building RBF models and updating search area $[l, u]$ in the next generation.

3.3 Framework of the proposed DESO

The proposed DESO method combines the global search and the local search to balance exploration and exploitation. They are realized by two different types of evolutionary sampling strategies.

The framework of DESO is shown in Fig.1. The solid arrows stand for the flows of the algorithm, and the dotted arrows represent the data flow. The pseudo code of DESO is shown in Algorithm 1. Initially, N_0 sample points are generated from the design space by using Latin hypercube sampling (LHS) [47], which segments the distribution into n intervals and makes sure that at least one value is randomly selected from each interval. The number of intervals, n , is the number of iterations of LHS.

Then, the fitness value of samples is evaluated, and the database is generated. Database D consists of all evaluated sample points. Before evolutionary sampling, D is sorted as D_{sort} according to the fitness value. Evolutionary sampling is used to find the promising sample point for fitness evaluation. In this phase, DESO randomly selects a strategy between SS and SLS in each iteration until the termination condition is met. Through this way, evolutionary sampling effectively accelerates the convergence and avoids falling into the local optimum. If SS is employed, N best sample points are used to form the population. All sample points are used to build the global RBF model. As the number of training sample points increases, the process of the training surrogate model becomes time-consuming. Thus when the number of sample points is large enough (more than m), the global RBF model is built by using m best sample points. If the SLS strategy is employed, τ best sample points are used to build the local RBF model. And the search space is set based on the τ best sample points. SS plays the role of exploration and SLS plays the role of exploitation.

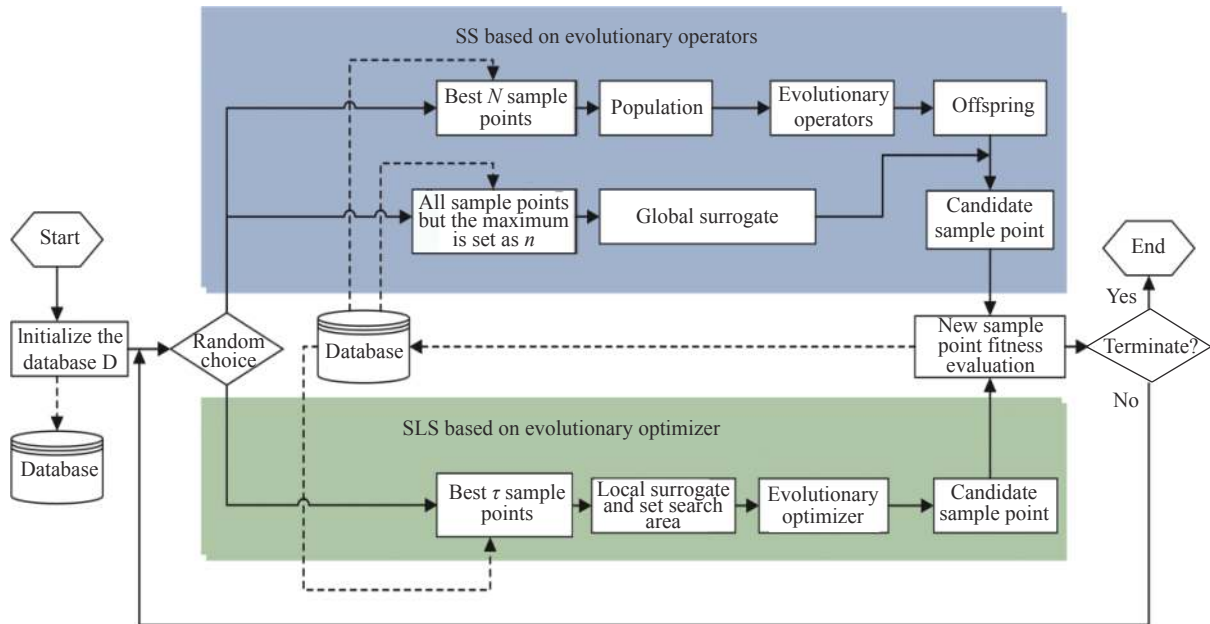


Fig. 1 A diagram of the proposed framework

4. Experimental study and discussion

The algorithm considers the characteristics of high-dimensional problems, and set adaptive parameters in SLS for different dimensions. Therefore, it can solve the problems with different dimensions. In this section, we set up 30 test functions including six types of test questions and five different dimensions, as shown in Table 1. Table 2 shows the result of DESO on 30 test functions. For con-

venience, we use simplified names for easy understanding of a specific problem. For example, G20 represents the 20-dimensional Griewank problem, and SRR100 represents the 100-dimensional shifted rotated rastrigin problem. Moreover, we take the discussion and experimental study of DESO. On one hand, we discuss the effect of sampling strategies (SS and SLS) on DESO. On the other hand, we research the impact of the evolutionary optimizer on SLS by testing different optimizers. Finally,

we compare the proposed DESO algorithm with five state-of-the-art algorithms. All test problems are limited in

1 000 function evaluations, and 20 independent runs are performed to obtain statistical results.

Table 1 Properties of the test problems

Abbreviation	Problem	d	Search space	Optimum	Property
E	Ellipsoid	20,30,50,100,200	$[-5.12, 5.12]^d$	0	Unimodal
R	Rosenbrock	20,30,50,100,200	$[-2.048, 2.048]^d$	0	Multimodal
A	Ackley	20,30,50,100,200	$[-32.768, 32.768]^d$	0	Multimodal
G	Griewank	20,30,50,100,200	$[-600, 600]^d$	0	Multimodal
SRR	Shifted rotated rastrigin (f10 in CEC05 [48])	20,30,50,100,200	$[-5, 5]^d$	-330	Very complicated multimodal
RHC	Rotated hybrid composition function (f19 in CEC05 [48])	20,30,50,100,200	$[-5, 5]^d$	10	Very complicated multimodal

Table 2 Statistical results of 30 test problems obtained by DESO

Problem	Best	Worst	Mean	Median	Standard
E20	1.64E-16	1.95E-14	3.74E-15	2.55E-15	4.57E-15
E30	1.00E-09	3.96E-07	9.04E-08	8.36E-08	9.19E-08
E50	1.12E-03	2.70E-02	7.67E-03	5.69E-03	5.80E-03
E100	6.38E+00	1.96E+01	1.23E+01	1.15E+01	4.26E+00
E200	3.12E+02	6.65E+02	4.80E+02	4.62E+02	1.04E+02
R20	1.16E+01	1.65E+01	1.30E+01	1.30E+01	1.14E+00
R30	2.35E+01	2.84E+01	2.47E+01	2.45E+01	1.27E+00
R50	4.50E+01	4.92E+01	4.66E+01	4.66E+01	9.83E-01
R100	9.78E+01	1.07E+02	1.01E+02	1.00E+02	2.35E+00
R200	2.72E+02	4.25E+02	3.49E+02	3.49E+02	3.68E+01
A20	8.52E-07	1.42E-05	3.96E-06	2.92E-06	3.10E-06
A30	7.64E-06	8.95E-05	3.21E-05	2.56E-05	1.82E-05
A50	4.24E-04	1.37E+00	2.62E-01	1.82E-03	4.80E-01
A100	2.04E+00	3.75E+00	2.89E+00	2.89E+00	4.71E-01
A200	5.10E+00	6.56E+00	5.94E+00	5.94E+00	4.00E-01
G20	2.04E-10	5.01E-01	5.68E-02	1.46E-06	1.52E-01
G30	1.60E-06	1.19E-02	7.96E-04	4.02E-05	2.63E-03
G50	1.47E-03	3.89E-02	9.60E-03	6.03E-03	1.02E-02
G100	9.20E-01	1.10E+00	1.03E+00	1.03E+00	5.66E-02
G200	9.45E+00	2.39E+01	1.61E+01	1.55E+01	3.38E+00
SRR20	-2.12E+02	-1.17E+02	-1.59E+02	-1.55E+02	2.38E+01
SRR30	-1.11E+02	-3.04E+01	-7.25E+01	-7.72E+01	2.35E+01
SRR50	1.63E+02	3.46E+02	2.30E+02	2.25E+02	4.64E+01
SRR100	9.99E+02	1.39E+03	1.20E+03	1.19E+03	1.18E+02
SRR200	4.55E+03	5.14E+03	4.85E+03	4.83E+03	1.52E+02
RHC20	1.11E+03	1.25E+03	1.19E+03	1.20E+03	3.59E+01
RHC30	9.41E+02	1.02E+03	9.66E+02	9.64E+02	1.89E+01
RHC50	1.02E+03	1.07E+03	1.04E+03	1.04E+03	1.75E+01
RHC100	1.19E+03	1.40E+03	1.35E+03	1.37E+03	4.99E+01
RHC200	1.19E+03	1.50E+03	1.35E+03	1.35E+03	7.49E+01

4.1 Parameter settings

The proposed DESO framework is flexible to implement

a specific algorithm. Parameters are set in this paper as follows. Initially, N_0 sample points are generated by LHS, then the fitness value is evaluated. N_0 is set to 100 when the dimension is less than 100, otherwise N_0 is 200. Database contains all sample points evaluated by the original objective function. In the SS strategy, we set N to 50 and m to the number of samples in the database, but limit the maximum value of m to 300 [33]. In the SLS strategy, in order to adapt to problems with different dimensions, τ is set to $25+d$, and the maximum of τ is set to 60, where d represents the dimension of the problem. The evolutionary operators of DE are used in the two strategies, and F and Cr are set to 0.5 and 0.9, respectively. In SLS, when using the evolutionary optimizer to search for candidate solution, the termination condition of the optimizer is that the number of surrogate model evaluations reaches 1 000+ $100d$ or the optimizer does not find better predictions for 10 consecutive generations.

4.2 Discussions

4.2.1 Discussion on sampling strategy

The key of DESO is the data-driven evolutionary sampling strategy. In order to verify effectiveness of the evolutionary sampling process in this paper, we compare DESO with its two variants that only use SS or SLS in the evolutionary sampling process, which are referred to here as DESO-SS and DESO-SLS. It can be seen from Figs. 2–5 that DESO-SS has a slower convergence rate comparing with DESO and DESO-SLS, where NFE stands for the number of fitness evaluation. The reason is that the SS strategy prefers exploration rather than exploitation. On the contrary, DESO-SLS initially converges faster than DESO. However, because the SLS strategy only focuses on local exploitation, it is easy to fall into local optimum in the search process. Randomly employing the SLS strategy and the SS strategy can balance the two capabilities of exploration and exploitation, and improve robustness of the algorithm.

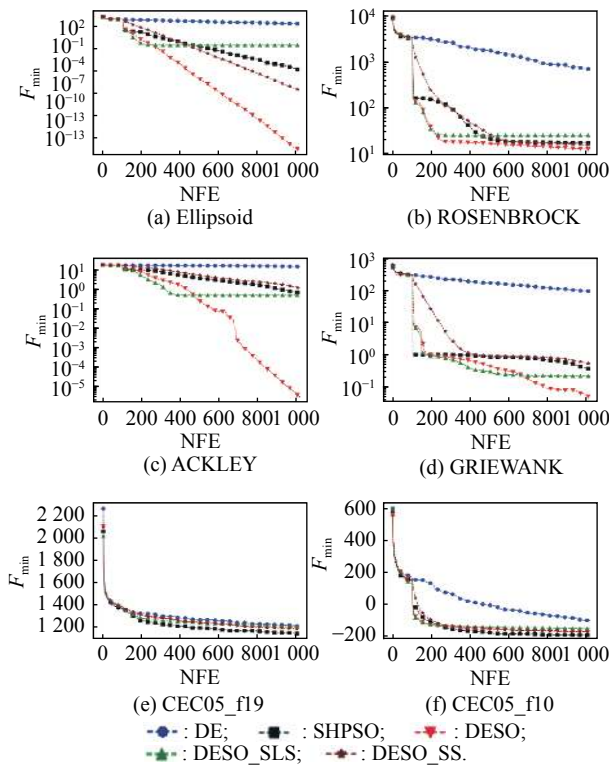


Fig. 2 Convergence curves of different algorithms for the 20D benchmark problems

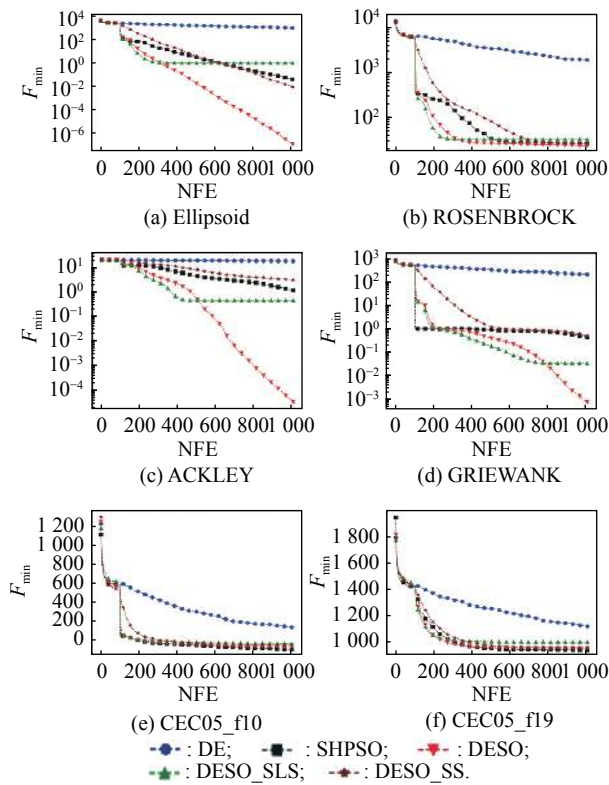


Fig. 3 Convergence curves of different algorithms for the 30D benchmark problems

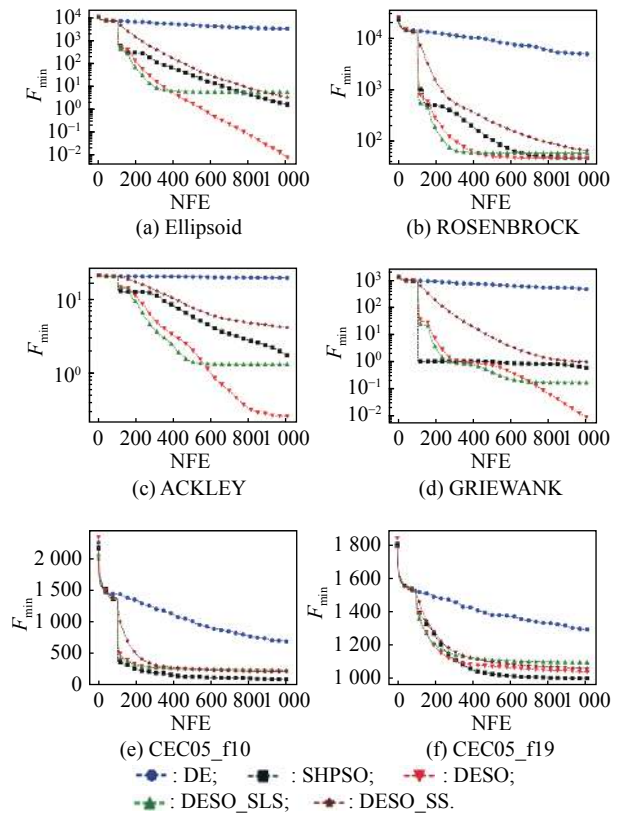


Fig. 4 Convergence curves of different algorithms for the 50D benchmark problems

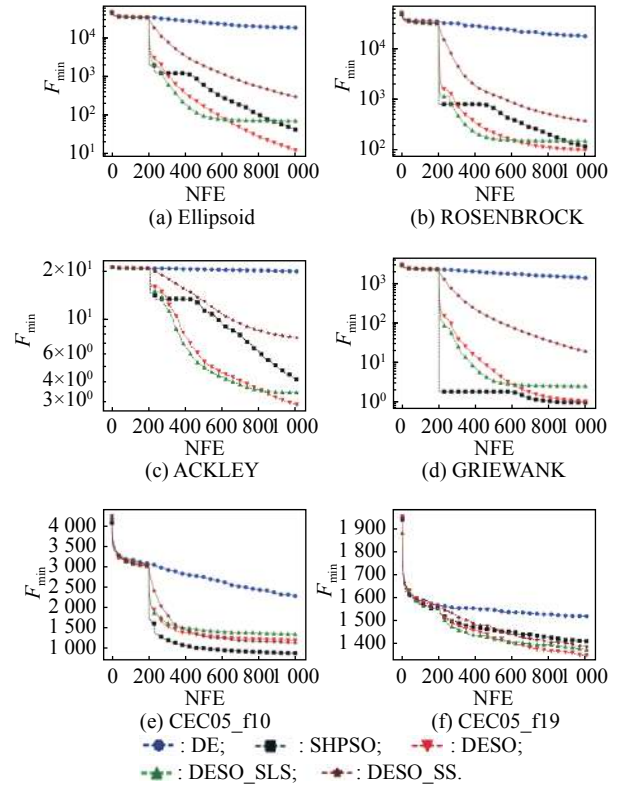


Fig. 5 Convergence curves of different algorithms for the 100D benchmark problems

As shown in Table 3, DESO is significantly better than DESO-SS and DESO-SLS in Ellipsoid, Rosenbrock, Ackley, and Griewank problems, but for SRR and RHC,

which are very complicated multimodal problems, their effects are not much different. Even for SRR50, SRR100 and RHC20, DESO-SS is slightly better than DESO.

Table 3 Comparison of DESO using different sampling strategies

Problem	DESO-SS	DESO-SLS	DESO
	Mean (Standard)	Mean (Standard)	Mean (Standard)
E20	3.6664E-07(1.9910E-07)	3.2815E-01(3.5169E-01)	3.7414E-15(4.5651E-15)
E30	7.4239E-03(1.0365E-02)	9.4273E-01(5.4385E-01)	9.0387E-08(9.1891E-08)
E50	3.4855E+00(2.3903E+00)	6.2837E+00(2.2834E+00)	7.6666E-03(5.8049E-03)
E100	2.9328E+02(6.2192E+01)	7.2900E+01(2.8170E+01)	1.2283E+01(4.2566E+00)
R20	1.6134E+01(1.3308E+00)	2.6261E+01(8.6217E+00)	1.3039E+01(1.1425E+00)
R30	2.7951E+01(9.6690E-01)	3.4825E+01(7.0368E+00)	2.4713E+01(1.2670E+00)
R50	6.6163E+01(2.0960E+01)	5.9539E+01(6.9527E+00)	4.6602E+01(9.8297E-01)
R100	3.7220E+02(6.3760E+01)	1.5384E+02(2.1127E+01)	1.0127E+02(2.3479E+00)
A20	1.4194E+00(4.9019E-01)	6.0124E-01(5.2485E-01)	3.9556E-06(3.0972E-06)
A30	3.1331E+00(5.5586E-01)	4.8188E-01(3.1148E-01)	3.2057E-05(1.8247E-05)
A50	4.1422E+00(6.2899E-01)	1.3383E+00(3.4250E-01)	2.6203E-01(4.7966E-01)
A100	7.6050E+00(9.1124E-01)	3.4714E+00(4.7813E-01)	2.8883E+00(4.7124E-01)
G20	5.8210E-01(2.1501E-01)	2.4515E-01(4.0269E-01)	5.6784E-02(1.5159E-01)
G30	5.3728E-01(1.8834E-01)	3.8896E-02(1.9702E-02)	7.9600E-04(2.6309E-03)
G50	1.0964E+00(1.2913E-01)	1.9459E-01(6.1384E-02)	9.5999E-03(1.0159E-02)
G100	1.8652E+01(4.9693E+00)	2.5451E+00(6.6982E-01)	1.0250E+00(5.6619E-02)
SRR20	-1.5860E+02(1.4435E+01)	-1.3860E+02(1.8281E+01)	-1.5919E+02(2.3821E+01)
SRR30	-5.6265E+01(2.6909E+01)	-3.8143E+01(3.4480E+01)	-7.2471E+01(2.3485E+01)
SRR50	2.2778E+02(5.4075E+01)	2.5261E+02(4.4738E+01)	2.3005E+02(4.6423E+01)
SRR100	1.1417E+03(6.8809E+01)	1.3542E+03(1.0738E+02)	1.2021E+03(1.1833E+02)
RHC20	1.1807E+03(5.0535E+01)	1.2103E+03(6.3443E+01)	1.1926E+03(3.5859E+01)
RHC30	9.6940E+02(2.3313E+01)	1.0127E+03(3.8582E+01)	9.6599E+02(1.8853E+01)
RHC50	1.0631E+03(2.1495E+01)	1.0988E+03(3.6477E+01)	1.0411E+03(1.7546E+01)
RHC100	1.3871E+03(3.8007E+01)	1.3784E+03(3.3023E+01)	1.3541E+03(4.9898E+01)

4.2.2 Discussion on SLS with different optimizers

We consider whether the optimizer used in the SLS strategy will have a great impact on DESO performance. Then we compare three variants of DESO, DESO-PSO using the PSO algorithm as the optimizer, DESO-JADE using the JADE algorithm as the optimizer and DESO-DE using the DE algorithm as the optimizer (that is, the DESO algorithm introduced in this article). The results can be seen in Table 4. It shows that their results have not much difference. Different optimizers in SLS just have slight import to the performance of DESO when the used optimizers are effective. The reason may be that the SLS strategy searches on the surrogate model. The candidate solution depends heavily on the accuracy of the surrogate model. In high-dimensional problems, it is difficult to guarantee a promising candidate solution on the surrogate model also performs well on the original objective

function. Hence, improving the accuracy of the surrogate model is a more important issue than the optimizer.

4.3 Comparison with other methods

As seen in Table 2, DESO performs very well on the Ellipsoid, Rosenbrock, Ackley, and Griewank problems. The median of E20-E50, A20-A50, and G20-G50 can all reach the order of 10^3 , and it can be considered that a suitable optimal solution has been found. Due to the inherent characteristics of the Rosenbrock function, the late convergence is very slow. SRR and RHC problems are very complicated. It is not easy for DESO to find the basin of attraction of the global minimum, then exert its efficient exploitation capability. For the same type of problem, the evaluation value of the low-dimensional problem is smaller than that of the high-dimensional problem, but G20 is an exception. According to Locatelli's research

[49], the reason is that the effect of the production of cosine components (the second term of Griewank function) can be neglected as the dimension increases. In addition, compared with A30, the difference between the best solution of A50 and the worst solution of A50 suddenly be-

comes larger. Because as the dimension increases, it becomes difficult to quickly find the global optimal solution. In A50, sometimes the global optimal region can be quickly found by exploration, but sometimes it cannot. The same problem also occurs on the G20.

Table 4 Comparison of DESO using different local search optimizers

Problem	DESO-DE		DESO-PSO		DESO-JADE	
	Mean (Standard)		Mean (Standard)		Mean (Standard)	
E30	9.0387E-08(9.1891E-08)		1.0431E-08 (1.2405E-08)		2.0660E-08(3.5522E-08)	
E50	7.6666E-03(5.8049E-03)		2.9575E-03(2.5580E-03)		2.0126E-03(2.1218E-03)	
E100	1.2283E+01(4.2566E+00)		3.1564E+00(1.1696E+00)		2.9194E+00(1.4148E+00)	
R30	2.4713E+01(1.2670E+00)		2.3802E+01(1.0361E+00)		2.4079E+01(5.9837E-01)	
R50	4.6602E+01(9.8297E-01)		4.5721E+01(6.5097E-01)		4.5523E+01(7.1813E-01)	
R100	1.0127E+02(2.3479E+00)		9.7918E+01(4.5785E-01)		9.8597E+01(6.2668E-01)	
A30	3.2057E-05(1.8247E-05)		4.9287E-01(6.3449E-01)		5.7762E-02(2.5830E-01)	
A50	2.6203E-01(4.7966E-01)		7.8886E-01(7.0006E-01)		6.0062E-01(6.9700E-01)	
A100	2.8883E+00(4.7124E-01)		2.8424E+00(3.9646E-01)		3.0345E+00(2.9678E-01)	
G30	7.9600E-04(2.6309E-03)		1.3273E-02(4.9904E-02)		5.0524E-05(1.5774E-04)	
G50	9.5999E-03(1.0159E-02)		2.7314E-03(5.1458E-03)		2.4616E-03(3.0528E-03)	
G100	1.0250E+00(5.6619E-02)		4.1510E-01(7.4805E-02)		5.2188E-01(1.2095E-01)	

Fig. 2 to Fig. 5 respectively show the convergence process of DESO in the 20-dimensional to 100-dimensional problems, where the x-axis is NFE and the y-axis is the average current fitness value. Compared with the DE algorithm, DESO significantly accelerates the convergence speed. With only 1 000 function evaluations, DESO can obtain an appropriate solution. We compare the DESO algorithm with success-history based adaptive DE (SHADE) [50], an excellent variant of DE, and four state-of-the-art SAEA algorithms, such as GPME [15], SACOSO [35], SHPSO [33] and ESAO [32]. Compared with the five algorithms, DESO has obvious advantages to different dimensions of the Ellipsoid, Rosenbrock, Ackley, and Griewank problems. DESO converges faster and has a higher accuracy. The comparison results of

DESO and the five algorithms can be seen in Table 5. In this table, results of GPME, SA-COSO, SHPSO and ESAO refer to their origin literature. N/A means there is no result of the test function in literature. The best results of test functions are bold. DESO significantly outperforms the SHADE on all benchmark functions. It shows that SAEA is better than traditional EA in expensive problems. For different dimensional Ellipsoid, Rosenbrock, Ackley, and Griewank problems, DESO is significantly better than GPME, SA-COSO, SHPSO, and ESAO. The result demonstrates DESO has powerful search capabilities. For SRR and RHC problems, the performance of DESO is not much different from GPME, ESAO and SA-COSO, and worse than SHPSO.

Table 5 Comparison of DESO with other algorithms

Problem	SHADE		GPME		SA-COSO		SHPSO		ESAO		DESO	
	Mean	Standard	Mean	Standard	Mean	Standard	Mean	Standard	Mean	Standard	Mean	Standard
E20	121.994	54.925	1.300E-05	2.180E-05	N/A	N/A	N/A	N/A	1.81E-04	4.68E-04	3.741E-15	4.565E-15
E30	475.636	121.912	0.076	4.010E-02	N/A	N/A	0.212	0.152	0.027	0.070	9.039E-08	9.189E-08
E50	1698.723	391.560	221.080	81.612	51.475	16.246	4.028	2.060	0.740	0.555	7.667E-03	5.805E-03
E100	9140.519	1495.311	N/A	N/A	1033.200	317.180	76.106	21.447	1282.900	134.390	12.283	4.257
E200	44123.612	6920.598	N/A	N/A	16382.000	2981.100	N/A	N/A	1.76E+04	1.17E+03	479.920	103.740
R20	363.972	128.456	22.428	18.795	N/A	N/A	N/A	N/A	15.162	1.629	13.039	1.143
R30	1058.046	352.854	46.177	25.520	N/A	N/A	28.566	0.404	25.036	1.570	24.713	1.267
R50	2257.047	932.045	258.280	80.188	252.580	40.744	50.800	3.031	47.391	1.712	46.602	0.983

Continued

Problem	SHADE		GPEME		SA-COSO		SHPSO		ESAO		DESO	
	Mean	Standard	Mean	Standard	Mean	Standard	Mean	Standard	Mean	Standard	Mean	Standard
R100	6676.190	1502.416	N/A	N/A	2714.200	117.020	165.590	26.366	578.840	44.767	101.270	2.348
R200	16730.022	3760.863	N/A	N/A	16411.000	4096.500	N/A	N/A	4.31E+03	2.84E+02	348.720	36.776
A20	15.800	1.184	0.199	0.577	N/A	N/A	N/A	N/A	6.865	3.259	3.956E-06	3.097E-06
A30	16.507	1.205	3.011	0.925	N/A	N/A	1.442	0.774	2.521	0.840	3.206E-05	1.825E-05
A50	17.817	0.841	13.233	1.585	8.932	1.067	1.839	5.637	1.431	0.249	0.262	0.480
A100	18.532	0.411	N/A	N/A	15.756	0.503	4.113	0.592	10.364	0.211	2.888	0.471
A200	18.865	0.323	N/A	N/A	17.868	0.022	N/A	N/A	14.696	0.2193	5.943	0.400
G20	54.934	17.333	0.031	0.068	N/A	N/A	N/A	N/A	0.972	0.039	0.057	0.152
G30	127.121	47.575	0.997	0.108	N/A	N/A	0.921	0.088	0.953	0.050	7.960E-04	2.631E-03
G50	280.808	43.906	36.646	13.176	6.006	1.104	0.945	0.061	0.940	0.042	9.600E-03	1.016E-02
G100	726.981	103.158	N/A	N/A	63.353	19.021	1.070	0.020	57.342	5.839	1.025	0.057
G200	1528.380	190.728	N/A	N/A	577.760	101.400	N/A	N/A	572.903	36.043	16.141	3.381
SRR20	-8.473	59.846	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	-1.59E+02	2.38E+01
SRR30	179.346	79.508	-21.861	36.449	N/A	N/A	-92.830	22.544	6.325	26.477	-72.471	23.485
SRR50	633.467	89.457	N/A	N/A	197.160	30.599	134.420	32.256	198.610	45.825	230.050	46.423
SRR100	2235.075	267.703	N/A	N/A	1273.100	117.190	801.730	72.252	713.470	26.454	1202.100	118.330
SRR200	6480.683	252.517	N/A	N/A	3927.500	272.540	N/A	N/A	5.39E+03	156.8544	4853.100	151.620
RHC20	1395.179	81.068	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1.19E+03	35.9
RHC30	1229.265	49.472	958.590	25.695	N/A	N/A	939.610	9.018	931.670	8.942	965.990	18.853
RHC50	1304.984	33.006	N/A	N/A	1080.900	32.859	996.600	22.145	975.320	37.110	1041.100	17.546
RHC100	1567.296	48.421	N/A	N/A	1365.700	30.867	1419.800	38.238	1372.400	27.539	1354.100	49.898
RHC200	1571.078	38.177	N/A	N/A	1347.300	24.665	N/A	N/A	1.456E+03	20.432	1345.500	74.897

4.4 Application of DESO in an airfoil design problem

After DESO has shown its effectiveness on benchmark problems, we apply it to an airfoil design problem. The airfoil geometry is parameterized by the parametric section method (PARSEC) geometry representation method [51] in this work. Design variables for PARSEC are shown in Fig. 6. The optimization problem includes 11 variables. We have chosen the NACA2411 airfoil as the baseline because it is considered to be a good demonstrator for low subsonic flows [52]. The design values of the initial airfoil and search range are provided in Table 6. As the original value is p and the search range is r , the search area is $[p-r, p+r]$. We employ an objective function that solves the coefficient of lift of an airfoil using the Vortex panel method given input parameters. The goal of the optimization problem is to obtain the maximum lift coefficient through searching PARSEC parameters. To compare the algorithm with ESAO, SHPSO, and DE, we set the same parameters in experiment. The initial sample points number is 50. The population size is set to 30. The budget of fitness evaluations is 500. For each algorithm, 20 independent runs are carried out and the statistical results are given. Table 7 shows the statistical results of the airfoil problem using DESO, ESAO, SHPSO, and DE.

DESO performs the best on all the criteria. ESAO and SHPSO obtain some worse designs, and they hold almost the same performance. DE performs the worst. Fig. 7 shows the best optimized geometries of the airfoil by the four algorithms. The results of DESO, ESAO and SHPSO are basically in the same shape. The shape of DE is slightly different from them. Fig. 8 shows the original shape and the DESO optimized shape. The optimized shape has obvious changes. In addition, the convergence history of the airfoil problem using four algorithms is showed in Fig. 9. The result shows DESO performs the best over the whole optimization process. Especially, when the fitness evaluation is less, for example, 150 or 300, DESO outperforms other algorithms.

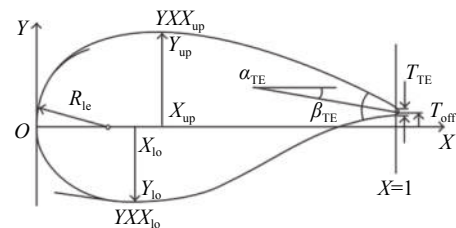


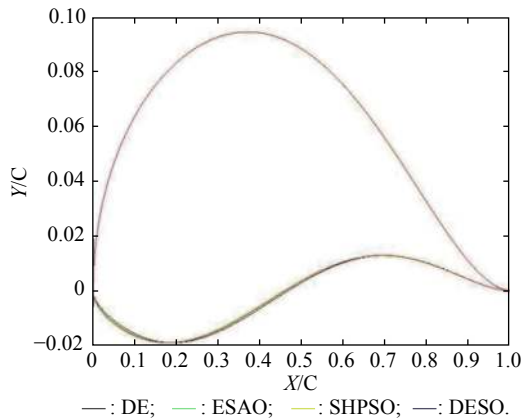
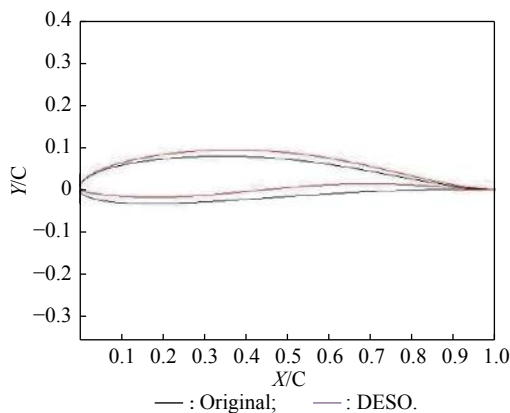
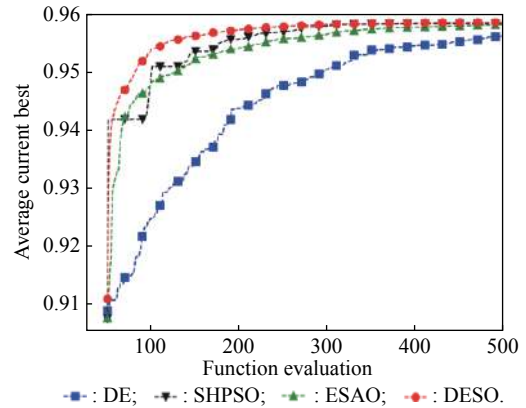
Fig. 6 Design variables for PARSEC

Table 6 Optimized PARSEC parameters

Design parameter	Original value	Search range
(R_{le}) Leading edge radius	0.021 6	0.001 5
(X_{up}) Position of upper crest	0.344 5	0.025
(Y_{up}) Upper crest point	0.079 12	0.015
(YXX_{up}) Upper crest curvature	-0.644 8	0.01
(X_{lo}) Position of lower crest	0.17	0.02
(Y_{lo}) Lower crest point	-0.033 797	0.015
(YXX_{lo}) Lower crest curvature	0.674 8	0.075
(T_{TE}) Trailing edge thickness	0	0
(T_{off}) Trailing edge offset	0	0
(α_{TE}) Trailing edge direction angle	0	0.175
(β_{TE}) Trailing edge wedge angle	0	0.05

Table 7 Results of comparison

Algorithm	Best	Worst	Mean	Median	Standard
DESO	0.958 81	0.958 62	0.958 76	0.958 78	0.000 049
DE	0.958 5	0.952 14	0.956 51	0.956 61	0.001 502
SHPSO	0.958 78	0.957 32	0.958 53	0.958 63	0.000 33
ESAO	0.958 68	0.957 91	0.958 38	0.958 4	0.000 221

**Fig. 7** Geometry of the best optimized airfoils of four algorithms**Fig. 8** Geometry of the initial and the best optimized airfoils of DESO**Fig. 9** Convergence history of the airfoil problem using four algorithms

5. Conclusions

To solve the expensive optimization problems, this paper proposes a DESO framework, which combines two different degrees of evolutionary sampling strategies, SS and SLS. These two sampling strategies use the historical data to construct global surrogate and local surrogate to assist sampling, respectively. Moreover, DESO sets adaptive parameters for different dimension problems. Experiments with different sampling strategies show that randomly employing two sampling strategies can balance global search and local search and avoid falling into the local optimum. The proposed algorithm is compared with five state-of-the-art algorithms on six widely used benchmark problems of different dimensions ranging from 20 to 200. The results indicate that DESO is effective and efficient. Finally, an application of the airfoil design problem also shows the superiority of DESO. In the future, more efficient sampling strategies need be studied for higher-dimensional problems. Moreover, DESO works well to accelerate the convergence, but its exploration ability should be further improved when handling very complicated problems.

References

- [1] LIU H, ONG Y S, CAI J. A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. *Structural and Multidisciplinary Optimization*, 2018, 57(1): 393–416.
- [2] FORRESTER A I, BRESSLOFF N W, KEANE A J. Optimization using surrogate models and partially converged computational fluid dynamics simulations. *Proc. of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2006: 2177–2204.
- [3] PAPADRAKAKIS M, LAGAROS N D, TSOMPANAKIS Y. Structural optimization using evolution strategies and neural networks. *Computer Methods in Applied Mechanics and Engineering*, 1998, 156(1–4): 309–333.
- [4] BONTE M H, FOURMENT L, DO T, et al. Optimization of

- forging processes using finite element simulations. *Structural and Multidisciplinary Optimization*, 2010, 42(5): 797–810.
- [5] JIN Y C, SENDHOFF B. A systems approach to evolutionary multiobjective structural optimization and beyond. *IEEE Computational Intelligence Magazine*, 2009, 4(3): 62–76.
- [6] EMMERICH M, GIOTIS A, OZDEMIR M, et al. Metamodel-assisted evolution strategies. Proc. of the International Conference on Parallel Problem Solving from Nature, 2002: 361–370.
- [7] JIN Y C. Surrogate-assisted evolutionary computation: recent advances and future challenges. *Swarm and Evolutionary Computation*, 2011, 1(2): 61–70.
- [8] JIN Y C, WANG H, CHUGH T, et al. Data-driven evolutionary optimization: an overview and case studies. *IEEE Trans. on Evolutionary Computation*, 2018, 23(3): 442–458.
- [9] CAI X W, QIU H B, GAO L, et al. An efficient surrogate assisted particle swarm optimization algorithm for high-dimensional expensive problems. *Knowledge-Based Systems*, 2019, 184: 104901.
- [10] GONG W Y, ZHOU A M, CAI Z H. A multioperator search strategy based on cheap surrogate models for evolutionary optimization. *IEEE Trans. on Evolutionary Computation*, 2015, 19(5): 746–758.
- [11] LI F, CAI X W, GAO L, et al. A surrogate-assisted multiswarm optimization algorithm for high-dimensional computationally expensive problems. *IEEE Trans. on Cybernetics*, 2020, 51(3): 1390–1402.
- [12] KRITHIKAA M, MALLIPEDDI R. Differential evolution with an ensemble of low-quality surrogates for expensive optimization problems. Proc. of the IEEE Congress on Evolutionary Computation, 2016: 78–85.
- [13] MYERS R H, MONTGOMERY D C, VINING G G, et al. Response surface methodology: a retrospective and literature survey. *Journal of Quality Technology*, 2004, 36(1): 53–77.
- [14] BUCHE D, SCHRAUDOLPH N N, KOUMOUTSAKOS P. Accelerating evolutionary algorithms with Gaussian process fitness function models. *IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2005, 35(2): 183–194.
- [15] LIU B, ZHANG Q, GIELEN G G. A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Trans. on Evolutionary Computation*, 2013, 18(2): 180–192.
- [16] ZHAN Y, SU G S, YAN L B. Cooperative optimization algorithm based on particle swarm optimization and Gaussian process. *Journal of Systems Engineering and Electronics*, 2013, 35(6): 1342–1347.
- [17] MARTIN J D, SIMPSON T W. Use of kriging models to approximate deterministic computer models. *AIAA Journal*, 2005, 43(4): 853–863.
- [18] KLEIJNEN J P. Kriging metamodeling in simulation: a review. *European Journal of Operational Research*, 2009, 192(3): 707–716.
- [19] PARK J, SANDBERG I W. Universal approximation using radial-basis function networks. *Neural Computation*, 1991, 3(2): 246–257.
- [20] JIN Y C, OLHOFFER M, SENDHOFF B. A framework for evolutionary optimization with approximate fitness functions. *IEEE Trans. on Evolutionary Computation*, 2002, 6(5): 481–494.
- [21] GASPAR-CUNHA A, VIEIRA A. A hybrid multi-objective evolutionary algorithm using an inverse neural network. Proc. of the Hybrid Metaheuristics, 2004: 25–30.
- [22] GASPAR-CUNHA A, VIEIRA A. A multi-objective evolutionary algorithm using neural networks to approximate fitness evaluations. *International Journal of Computers, Systems and Signals*, 2005, 6(1): 18–36.
- [23] LOSHCILLOV I, SCHOENAUER M, SEBAG M. Comparison-based optimizers need comparison-based surrogates. Proc. of the International Conference on Parallel Problem Solving from Nature, 2010: 364–373.
- [24] POLOCZEK J, KRAMER O. Local SVM constraint surrogate models for self-adaptive evolution strategies. Proc. of the Annual Conference on Artificial Intelligence, 2013: 164–175.
- [25] WANG H, JIN Y, DOHERTY J. Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Trans. on Cybernetics*, 2017, 47(9): 2664–2677.
- [26] WANG H D, JIN Y C, SUN C L, et al. Offline data-driven evolutionary optimization using selective surrogate ensembles. *IEEE Trans. on Evolutionary Computation*, 2018, 23(2): 203–216.
- [27] JONES D R, SCHONLAU M, WELCH W J. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 1998, 13(4): 455–492.
- [28] BRANKE J, SCHMIDT C. Faster convergence by means of fitness estimation. *Soft Computing*, 2005, 9(1): 13–20.
- [29] LI F, CAI X W, GAO L. Ensemble of surrogates assisted particle swarm optimization of medium scale expensive problems. *Applied Soft Computing*, 2019, 74: 291–305.
- [30] BHOSEKAR A, IERAPETRITOU M. Advances in surrogate based modeling, feasibility analysis, and optimization: a review. *Computers & Chemical Engineering*, 2018, 108: 250–267.
- [31] ZHOU Z Z, ONG Y S, NAIR P B, et al. Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2007, 37(1): 66–76.
- [32] WANG X, WANG G G, SONG B, et al. A novel evolutionary sampling assisted optimization method for high-dimensional expensive problems. *IEEE Trans. on Evolutionary Computation*, 2019, 23(5): 815–827.
- [33] YU H B, TAN Y, ZENG J C, et al. Surrogate-assisted hierarchical particle swarm optimization. *Information Sciences*, 2018, 454/455: 59–72.
- [34] CAI X W, GAO L, LI X Y. Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems. *IEEE Trans. on Evolutionary Computation*, 2019, 24(2): 365–379.
- [35] SUN C L, JIN Y C, CHENG R, et al. Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Trans. on Evolutionary Computation*, 2017, 21(4): 644–660.
- [36] ONG Y S, NAIR P B, KEANE A J. Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA Journal*, 2003, 41(4): 687–696.
- [37] STORN R, PRICE K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997, 11(4): 341–359.

- [38] FEOKTISTOV V. Differential evolution. New York: Springer, 2006.
- [39] ZHANG J, SANDERSON A C. JADE: adaptive differential evolution with optional external archive. *IEEE Trans. on Evolutionary Computation*, 2009, 13(5): 945–958.
- [40] GONG W Y, CAI Z H. Differential evolution with ranking-based mutation operators. *IEEE Trans. on Cybernetics*, 2013, 43(6): 2066–2081.
- [41] WU G H, SHEN X, LI H F, et al. Ensemble of differential evolution variants. *Information Sciences*, 2018, 423: 172–186.
- [42] DAS S, SUGANTHAN P N. Differential evolution: a survey of the state-of-the-art. *IEEE Trans. on Evolutionary Computation*, 2010, 15(1): 4–31.
- [43] DAS S, MULLICK S S, SUGANTHAN P N. Recent advances in differential evolution—an updated survey. *Swarm and Evolutionary Computation*, 2016, 27: 1–30.
- [44] KENNEDY J, EBERHART R. Particle swarm optimization. *Proc. of the International Conference on Neural Networks*, 1995: 1942–1948.
- [45] CHENG R, JIN Y C. A social learning particle swarm optimization algorithm for scalable optimization. *Information Sciences*, 2015, 291: 43–60.
- [46] BISHOP C M. *Neural networks for pattern recognition*. Oxford: Oxford University Press, 1995.
- [47] HELTON J C, DAVIS F J. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety*, 2003, 81(1): 23–69.
- [48] SUGANTHAN P N, HANSEN N, LIANG J J, et al. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Singapore: Nanyang Technological University, 2005.
- [49] LOCATELLI M. A note on the Griewank test function. *Journal of Global Optimization*, 2003, 25(2): 169–174.
- [50] TANABE R, FUKUNAGA A. Success-history based parameter adaptation for differential evolution. *Proc. of the IEEE Congress on Evolutionary Computation*, 2013: 71–78.
- [51] SALUNKE N P, JUNED AHAMAD R A, CHANNIWALA S A. Airfoil parameterization techniques: a review. *American Journal of Mechanical Engineering*, 2014, 2(4): 99–102.
- [52] MUKESH R, LINGADURAI K, SELVAKUMAR U. Airfoil shape optimization using non-traditional optimization technique and its validation. *Journal of King Saud University-Engineering Sciences*, 2014, 26(2): 191–197.

Biographies



ZHEN Huixiang was born in 1995. He is now pursuing his Ph.D. degree in the School of Computer Science, China University of Geosciences, Wuhan, China. His research interests include the evolutionary computation and its applications.
E-mail: zhenhuixiang@cug.edu.cn



GONG Wenyin was born in 1979. He is currently a professor and Ph.D. student supervisor in the School of Computer Science, China University of Geosciences, Wuhan, China. He has published over 50 research papers in journals and international conferences. His research interests include differential evolution, memetic algorithms, multi-objective optimization, and their applications.

E-mail: wygong@cug.edu.cn



WANG Ling was born in 1972. He is currently a professor and Ph.D. student supervisor in the Department of Automation, Tsinghua University, Beijing, China. He has authored five academic books and over 260 refereed papers. His research interests include intelligent optimization and production scheduling.
E-mail: wangling@tsinghua.edu.cn