



A bi-stage surrogate-assisted hybrid algorithm for expensive optimization problems

Zhihai Ren¹ · Chaoli Sun¹ · Ying Tan¹ · Guochen Zhang¹ · Shufen Qin²

Received: 16 July 2020 / Accepted: 15 January 2021 / Published online: 7 February 2021
© The Author(s) 2021

Abstract

Surrogate-assisted meta-heuristic algorithms have shown good performance to solve the computationally expensive problems within a limited computational resource. Compared to the method that only one surrogate model is utilized, the surrogate ensembles have shown more efficiency to get a good optimal solution. In this paper, we propose a bi-stage surrogate-assisted hybrid algorithm to solve the expensive optimization problems. The framework of the proposed method is composed of two stages. In the first stage, a number of global searches will be conducted in sequence to explore different sub-spaces of the decision space, and the solution with the maximum uncertainty in the final generation of each global search will be evaluated using the exact expensive problems to improve the accuracy of the approximation on corresponding sub-space. In the second stage, the local search is added to exploit the sub-space, where the best position found so far locates, to find a better solution for real expensive evaluation. Furthermore, the local and global searches in the second stage take turns to be conducted to balance the trade-off of the exploration and exploitation. Two different meta-heuristic algorithms are, respectively, utilized for the global and local search. To evaluate the performance of our proposed method, we conduct the experiments on seven benchmark problems, the Lennard–Jones potential problem and a constrained test problem, respectively, and compare with five state-of-the-art methods proposed for solving expensive problems. The experimental results show that our proposed method can obtain better results, especially on high-dimensional problems.

Keywords Surrogate models · Meta-heuristic algorithms · Global search · Local search

This work was supported in part by National Natural Science Foundation of China (Grant no. 61876123), Natural Science Foundation of Shanxi Province (201801D121131, 201901D111264, 201901D111262), Fund Program for the Scientific Activities of Selected Returned Overseas Professionals in Shanxi Province, Shanxi Science and Technology Innovation project for Excellent Talents (201805D211028), and the China Scholarship Council (CSC).

✉ Chaoli Sun
chaoli.sun.cn@gmail.com

Zhihai Ren
S20180529@stu.tyust.edu.cn

Ying Tan
tanying2015@163.com

Guochen Zhang
imzgc@hotmail.com

Shufen Qin
shufen.qin@stu.tyust.edu.cn

Introduction

Many real-world optimization problems, such as aircraft design [25], rainfall prediction [13], and drug design [16], often require minutes, hours or days to take one performance evaluation on the designed parameters [12], which are called computationally expensive or time-consuming optimization problems. The meta-heuristic algorithms based on evolutionary algorithms such as swarm optimization algorithms have been shown to be efficient in solving discontinuous, non-differentiable or black-box problems. However, they are impeded to solve computationally expensive problems because a large number of expensive fitness evaluations will be required to seek an accepted solution. Surrogate models (also known as meta-models or approximate models) have been proposed to replace the expensive fitness evaluation in

¹ Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China

² Department of Electronic Information and Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China

the meta-heuristic algorithms to save the computational cost, because compared to the time-consuming fitness evaluation, the time to train a surrogate model can be fully neglected. Therefore, the surrogate models are also called the cheap functions corresponding to the exact expensive fitness function in the surrogate-assisted optimization algorithms. The commonly used surrogate models include support vector regression (SVR) [3], polynomial response surface (PRS) [17,22], artificial neural networks (ANNs) [8], radial basis function (RBF) [9,10,23,30,31,41], and Gaussian process (GP)(also called Kriging) models [14,20,39].

Many surrogate-assisted meta-heuristic algorithms (SAMAs) have been proposed for solving computationally expensive problems, which can be classified into two categories: SAMAs with a single surrogate model and SAMAs with multiple surrogate models. In SAMAs with a single surrogate model, only one kind of surrogate model, either global or local model, is utilized. Liu et al. [20] proposed a global GP-assisted evolutionary algorithm for medium-scale computationally expensive optimization problems. Schneider et al. [24] employed the Bayesian optimization with Gaussian process model to optimize the optical structure with many degrees of freedom. Tian et al. [33] developed a GP assisted evolutionary algorithm and proposed to select individual for exact fitness evaluation by a multi-objective infill criterion. Yu et al. [37] integrated a restart strategy and a global RBF model to offer a powerful optimizer for the computationally expensive problems. In [32], Sun et al. proposed to use the fitness estimation strategy to approximate the fitness of an individual, and the individuals whose approximated values obtained by the RBF model and the fitness estimation strategy are both better than its personal best position will be evaluated using the real expensive function. The method was further extended to solve the large-scale expensive problems in [29].

Different to SAMAs with a single surrogate model only, in SAMAs with multiple surrogate models, multiple models (fallen either in the same or in different categories) are proposed to solve the computationally expensive problems cooperatively. It can be further utilized as the surrogate ensembles or the hierarchical surrogate models. In the surrogate ensemble methods, multiple surrogates approximate the fitness value of a solution simultaneously. Lim et al. [19] proposed to use a surrogate ensemble and a lower-order PR model in the local search, and the position with better approximated value will be the next position of the individual. Lu et al. [21] proposed to employ the regression and classification techniques in the DE algorithm to recognize the individuals to be evaluated using the expensive fitness function. Sun et al. [31] proposed to adaptively select the approximated values from either the global or the local model for assisting the PSO algorithm to solve the computationally expensive problems. Wang et al. [35] built an ensemble surrogate based

on ensemble learning techniques and used it to deal with offline optimization problems. Different from the surrogate ensemble methods, in the hierarchical surrogate methods, different surrogate models are used for different motivations. The global surrogate models are usually expected to smooth out the local optimum so that the population can be prevented to get stuck in a local optimum. The local surrogate models are normally used to generate reliable approximated fitness values for the exact expensive problems. A global RBF model was proposed by Sun et al. [30] to assist a PSO variant to look for an optimal solution, which would be utilized to assist the fitness estimation on each individual in the PSO. Yu et al. [38] applied the SL-PSO algorithm to find the optimum of the RBF surrogate model, which will be evaluated using the exact expensive problem and used to update the RBF model used in the PSO algorithm. Cai et al. [1] proposed to combine the optima predicted by the global and local surrogate models with a mutation operator to fasten the convergence of the differential evolution.

In this paper, we propose a bi-stage surrogate-assisted hybrid algorithm for solving the computationally expensive problems, in which the social learning particle swarm optimization (SL-PSO), a PSO variant proposed by Cheng and Jin [2], is utilized to search for the optimal solution of the surrogate ensemble in the first stage. The framework of the ensemble has been shown to be better in assisting optimization than a single surrogate model. Therefore, the surrogate ensemble is used to assist preventing searching far away from the region where the optimal solution locates. Normally, the best solution of the surrogate model found by the global search is often chosen for exact fitness evaluation; however, the best positions found by different global searches may be the same, which will waste the finite computational budget. Therefore, the solution with maximum approximation uncertainty in the final population of a global search will be chosen to be evaluated using the real expensive function instead of the best solution found so far of the surrogate ensemble in our method. Thus, different positions can be explored in the first stage and correspondingly, the possibility to find a better solution can be increased. In the second stage, a local search on the RBF model using the differential evolution(DE) algorithm is added to exploit the neighborhood of the best solution found so far to speed up finding a better solution. Note that in our method, the local search means the search in a sub-region of the decision space. The sub-region is defined to be the space bounded by a number of closest neighbors of the best solution found so far. The local search is normally used to find a better position than the best solution found so far. However, it is easy to fall into a local optimum if only the sub-region around the best solution found so far is searched. Thus, the global and local searches take turns in the second stages to explore the whole decision space and exploit the

sub-region of the decision space, respectively, to prevent the premature convergence.

The rest of this paper is organized as follows. The next section gives a brief overview of the related techniques used in the proposed method. In the following section, the details of the proposed method are described. Experimental results are given and compared to some state-of-the-art methods proposed for computationally expensive problems before the final section. Finally, the paper concludes with a summary and some ideas for future work.

Related techniques

Social learning particle swarm optimization

The social learning particle swarm optimization (SL-PSO) [2] is a variant of the particle swarm optimization (PSO) [15], which has been shown to have a good capability on the exploration. In SL-PSO, an individual will be updated using the following equations:

$$\begin{aligned}
 v_{ij}(t + 1) &= r_1 * v_{ij}(t) + r_2 * (x_{kj}(t) - x_{ij}(t)) + r_3 * \varepsilon * (\bar{x}_j(t) - x_{ij}(t)) \quad (1) \\
 x_{ij}(t + 1) &= x_{ij}(t) + v_{ij}(t + 1), \quad (2)
 \end{aligned}$$

where $\mathbf{v}_i(t) = (v_{i1}(t), v_{i2}(t), \dots, v_{iD}(t))$ and $\mathbf{x}_i(t) = (x_{i1}(t), x_{i2}(t), \dots, x_{iD}(t))$ are the velocity and position of individual i at generation t . k represents an individual which has a better fitness value than individual i , $\bar{\mathbf{x}}(t) = (\bar{x}_1(t), \bar{x}_2(t), \dots, \bar{x}_D(t))$ is the average position of the population at generation t . r_1, r_2 and r_3 are random numbers generated in the range of [0,1]. ε is called the social influence factor which is used to control the influence of the average position of the population on the position update of each individual.

Differential evolution algorithm

The differential evolution(DE) [27] algorithm is a very efficient and robust stochastic real-parameter optimization algorithms [5]. To improve the performance of the DE algorithm, several variants have been proposed [5]. In our method, the canonical version of DE will be used. The offspring will be generated by sequentially executing the DE/rand/1 mutation (Eq.(3)) and binomial crossover (Eq.(4)), and will be used to update the positions of individuals in the population (Eq.(5)).

$$\begin{aligned}
 \mathbf{v}_i(t + 1) &= \mathbf{x}_k(t) + F \cdot (\mathbf{x}_p(t) - \mathbf{x}_q(t)) \quad (3) \\
 u_{i,j}(t + 1) &= \begin{cases} v_{i,j}(t + 1), & \text{if } (r \leq C \text{ or } j = j_{rand}), \\ x_{i,j}(t), & \text{otherwise.} \end{cases} \quad (4)
 \end{aligned}$$

$$\mathbf{x}_i(t + 1) = \begin{cases} \mathbf{u}_i(t + 1), & \text{if } (f(\mathbf{u}_i(t + 1)) < f(\mathbf{x}_i(t))), \\ \mathbf{x}_i(t), & \text{otherwise.} \end{cases} \quad (5)$$

In Eqs. (3) and (4), $\mathbf{x}_i(t) = (x_{i1}(t), x_{i2}(t), \dots, x_{iD}(t))$ is the position of individual i at generation t . $\mathbf{v}_i(t) = (v_{i1}(t), v_{i2}(t), \dots, v_{iD}(t))$ is an temporary solution and $\mathbf{u}_i(t) = (u_{i1}(t), u_{i2}(t), \dots, u_{iD}(t))$ is the offspring of individual i . k, p and q are three individuals randomly selected from the population which are not same to individual i . F is a scaling factor that typically lies in the interval [0.4, 1] [5], C is called the crossover rate which is a constant in the range of [0,1]. j_{rand} is a integer number randomly selected from the range [1,D], which is used to ensure that at least one dimension of an offspring is inherited from the solution \mathbf{v}_i . r is a random number in the range [0, 1]. $f(\mathbf{x}_i(t))$ and $f(\mathbf{u}_i(t + 1))$ are the fitness values of solution $\mathbf{x}_i(t)$ and $\mathbf{u}_i(t + 1)$, respectively.

Radial basis function

The radial basis function (RBF) network has been shown a good performance for problems with different fitness landscape characteristics [7,11,36,40]. The Gaussian process surrogate model has shown better performance for low-dimensional problems; however, it is not suitable for high-dimensional problems because more data are required to train a GP model, which is not possible for expensive optimization problems, and also, its training time will be greatly increased since the whole training data needs to be taken into account in the global search. For the polynomial surrogate model, as the accuracy of a first-order polynomial surrogate model is low, which we think will affect the approximation accuracy, and the second-order polynomial model cannot approximate strong nonlinear problems. Therefore, in our method, the RBF is adopted to be the surrogate model used in our method. The RBF network is composed of an input layer, a hidden layer and an output layer. Each node in the hidden layer performs a non-linear radially symmetric kernel function $\phi(\mathbf{x}) = \varphi(\|\mathbf{x} - \mathbf{c}_i\|)$ to measure the similarity between the input and the center of the function, and the output of the RBF network will be the weighted sum of the output from the hidden layer, which is given in the following:

$$\hat{f}(x) = \sum_{i=1}^{n_c} \omega_i \varphi(\|\mathbf{x} - \mathbf{c}_i\|), \quad (6)$$

where n_c is the number of nodes in the hidden layer.

Different kernel functions, such as Gaussian, multi-quadratic, thin-plate splines, cubic splines, invmulti-quadratics splines, etc., have been proposed to be used as the similarity function in the RBF. In our method, the cubic and invmulti-

quadratics kernel functions are adopted to be used in the RBF models for assisting both the global search and local search.

The bi-stage surrogate-assisted hybrid algorithm

As is known to all, the trade-off between the exploration and exploitation is important for the meta-heuristic algorithms to find the global optimal solution, and it is especially significant for algorithms when the computational budget is very limited. Therefore, in this paper, we propose a bi-stage surrogate-assisted hybrid algorithm, called BiS-SAHA, in which a number of global searches will be conducted in sequence in the first stage by a surrogate ensemble assisted SL-PSO to explore different regions of decision space, while in the second stage, a local search using an RBF-assisted differential evolution algorithm will be conducted after each global search to improve the exploitation capability to speed up finding an optimal solution. Note that the surrogate models used in every global search are different to each other because one solution in the population with the maximum uncertainty on the approximated value would be evaluated using the expensive problem and be used to update the model. The social learning particle swarm optimization is adopted to be the algorithm for the global search because it has been shown to have a good diversity to find the optimal solution [2]; however, its performance of the convergence is not good. Therefore, we adopted DE in our method instead of SL-PSO since the differential evolution algorithm has quicker convergence speed than SL-PSO.

Figure 1 gives a flowchart to show our proposed method. In Fig. 1, an archive DB will be initialized by generating a number of samples using the Latin hypercube sampling technique [26], which will be evaluated using the real expensive fitness function and saved to DB. The best solution with the minimum fitness value among all these samples, denoted as \mathbf{x}_b , will be kept and will be updated by the solutions, obtained either in the global or local search, that have been evaluated using the exact expensive function. Note that the local search will start to be conducted alternating with the global search when the transfer criterion is satisfied, i.e., a pre-defined computational resource for the first stage is met. In the following, we will describe the global and local search in detail.

The global search

Figure 2 gives the flowchart of the global search used in our method.

Before the method enters into the second stage, the following process will be repeated. N_m RBF models with different kernel functions, respectively, will firstly be trained using all data of the archive DB. The maximum of the values approx-

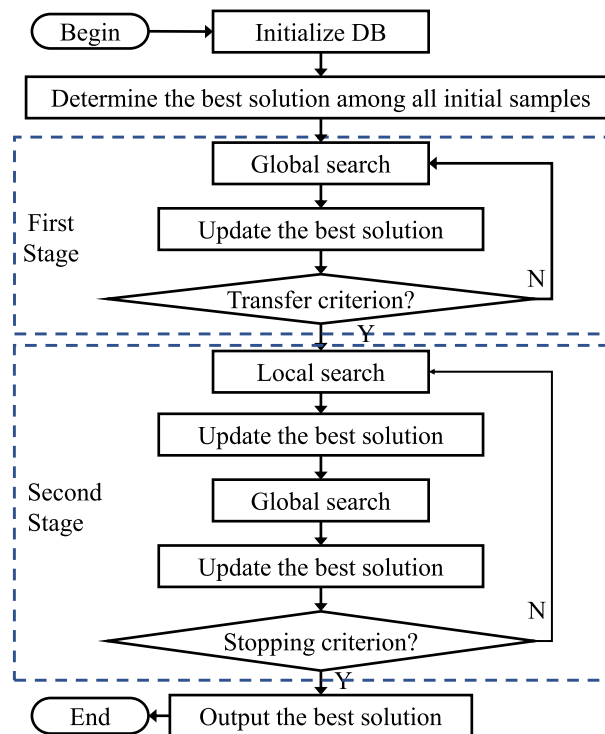


Fig. 1 The flowchart of BiS-SAHA

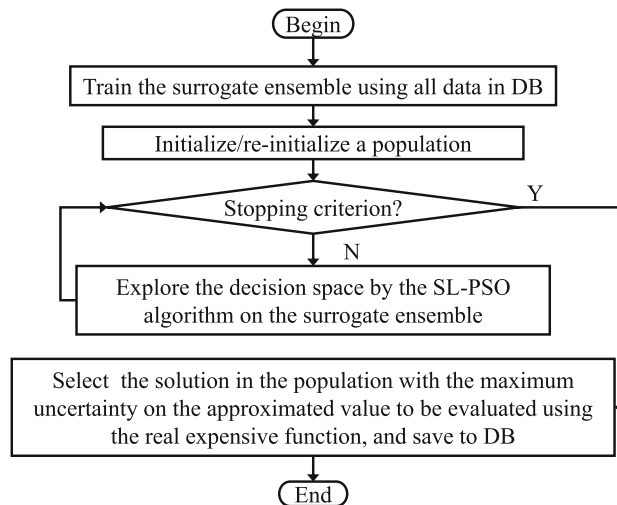


Fig. 2 The flowchart of the global search

imated by N_m RBF models on position \mathbf{x} will be its fitness value, as is shown in Eq. (7). The approximation uncertainty for each solution \mathbf{x} will be calculated using Eq. (8), which is the variance of the approximated fitness values on solution \mathbf{x} . In Eqs. (7) and (8), N_m is the number of surrogate models used for global search, $\hat{f}_{\text{RBF}_i}(\mathbf{x})$, $i = 1, 2, \dots, N_m$ is the value approximated by the i -th RBF model, $\hat{f}(\mathbf{x})$ is the mean value of N_m approximated values, i.e., $\hat{f}(\mathbf{x}) = \frac{\sum_{i=1}^{N_m} \hat{f}_{\text{RBF}_i}(\mathbf{x})}{N_m}$. Thus, from Eq. (8), we can see that larger the value $U(\mathbf{x})$

is, more different the approximated values on position \mathbf{x} are, resulting in a larger approximation uncertainty on this position.

$$\hat{f}(\mathbf{x}) = \max\{\hat{f}_{\text{RBF}_1}(\mathbf{x}), \hat{f}_{\text{RBF}_2}(\mathbf{x}), \dots, \hat{f}_{\text{RBF}_{N_m}}(\mathbf{x})\} \quad (7)$$

$$U(\mathbf{x}) = \frac{\sum_{i=1}^{N_m} (\hat{f}_{\text{RBF}_i}(\mathbf{x}) - \hat{f}(\mathbf{x}))^2}{N_m}. \quad (8)$$

Next, an initial population of the global search will be generated. To prevent the search deviating from the correct direction to find the global optimum of the surrogate ensemble, a number of solutions, which are selected using the clustering technique to ensure the diversity of a population, will be chosen from the archive DB in our method to be the initial population for the global search. Note that the number of clusters is not set the same to the size of the initial population, which we will give a sensitivity analysis on different settings in “Parameter sensitivity analysis of the BiS-SAHA”. Algorithm 1 gives the pseudocode of population initialization. The initial population of SL-PSO pop_g is set to null at first and then, individuals will be selected in a round turn from each clusters which is not null and added into pop_g until the size of population is met.

Algorithm 1 Pseudocode of population initialization/re-initialization

Input:

DB : the archive saving all solutions that have been evaluated using the exact fitness function;

N : the size of the population;

k : the number of clusters;

Output:

pop_g : an initial population of SL-PSO;

1: Classify the data in DB into k clusters, $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_k\}$, using k -means method;

2: $pop_g = \emptyset$;

3: **while** $|pop_g| < N$ **do**

4: **for** $i = 1$ to k **do**

5: **if** $|\mathbf{C}_i| \neq 0$ **then**

6: Randomly select one solution of \mathbf{C}_i , put it into pop_g and simultaneously remove it from \mathbf{C}_i ;

7: **if** $|pop_g| = N$ **then**

8: Exit;

9: **end if**

10: **end if**

11: **end for**

12: **end while**

13: Output pop_g

After that, the SL-PSO algorithm is utilized to search for the optimal solution of the surrogate ensemble. To prevent searching in a wrong direction that may deviate from the optimal solution of the expensive optimization problem, a pre-defined iteration g is given for each global search in the

first stage, and the individual in the population of t -th generation with the maximum uncertainty on the approximated fitness will be evaluated using the exact expensive fitness function. Note that the first stage is used to explore the whole decision space and the search on the surrogate ensemble for a number of generations is expected to find a region where a good solution of the expensive problem may locate. Simultaneously, to decrease the approximation uncertainty of the next surrogate model on this region, in our method, the individual with the maximum uncertainty on the approximated fitness is proposed to be evaluated using the real expensive fitness function.

The local search

The trade-off between exploration and exploitation is significantly essential for solving expensive optimization problems. In our method, the global searches applied in the first and second stages play the role to explore the decision space to improve the probability to find the optimal solution. However, the exploitation capability of the global search is not well because only the solution among the current population whose approximation uncertainty is maximum will be evaluated using the expensive problem. Therefore, in our method, we further employ a local search on a surrogate model to exploit a local region around the best position found so far, which is expected to locate accurately at the optimal solution of the expensive problem. Different from the global search that tries to find an optimal solution in the whole decision space, the local search is conducted to find a local optimum of the sub-space around the best position found so far. Furthermore, one RBF model is used instead of RBF surrogate ensemble for the local search, which is expected to prevent the loss of chances to locate at the optimal solution of the expensive optimization problem.

Figure 3 gives the flowchart of the local search. In the local search, an RBF surrogate model with the cubic kernel function is trained on all data in DB . The region where the local search is conducted will then be defined, which is surrounded by the maximum and minimum values on each dimension of all N_n data in DB that is closest to the best solution found so far. After that, an initial population will be generated randomly in this region, and a DE algorithm will be used to find a better solution in this region assisted by the RBF surrogate model. The individual in the population with the best approximated fitness value will be evaluated using the real expensive function and saved in DB for further usage.

Note that different from the method used in the global search to select the individual for real expensive fitness evaluation, in the local search, the individual with the best approximated fitness will be evaluated using the real time-consuming fitness function. The reason is that in the global search, we would like to find a region that a solution with

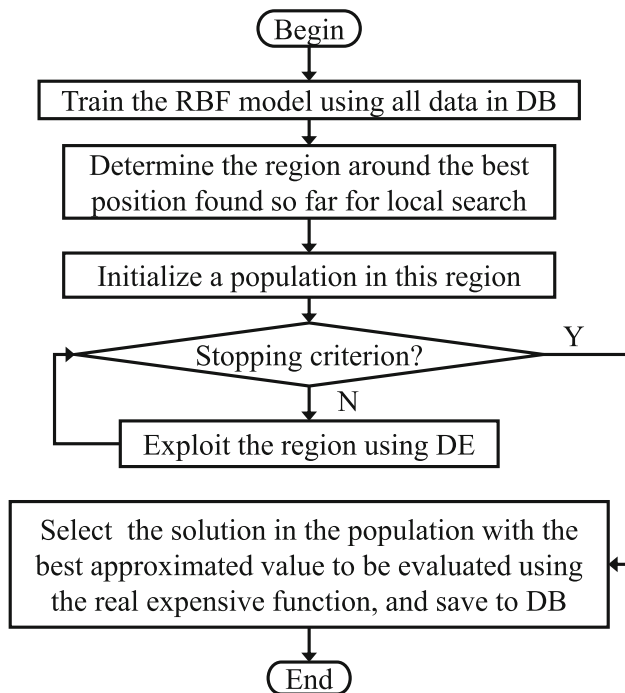


Fig. 3 The flowchart of the local search

good fitness value may locate. Therefore, it is essential to improve the approximation accuracy in this region by evaluating the solution, which has the maximum uncertainty in the final generation of the global search, using the exact expensive fitness function. While the motivation to use the local search is to find a better solution than the best position found so far, so it would be more meaningful to evaluate a solution whose approximated fitness value is better than the current best solution found so far.

Experimental study

To evaluate the performance of our proposed BiS-SAHA, we conduct a number of experiments on seven benchmark problems with either unimodal (F1) or multimodal (F2–F7) characteristics, a real-world application and a constrained problem with nonlinear constraints. The characteristics of seven benchmark problems are listed in Table 1. Five state-of-the-art algorithms, including the generation-based optimal restart strategy for surrogate-assisted social learning particle swarm optimization (GORS-SSLPSO) [37], the committee-based active learning for surrogate-assisted particle swarm optimization (CAL-SAPSO) [34], the offline data-driven evolutionary optimization using selective surrogate ensembles (DDEA-SE) [35], the surrogate-assisted hierarchical particle swarm optimization (SHPSO) [38], and the multiobjective infill criterion driven Gaussian process-assisted particle swarm optimization (MGP-SLPSO) [33], are used

to compare with our proposed method. Among these algorithms, GORS-SSLPSO [37] and MGP-SLPSO [33] are the surrogate-assisted meta-heuristics with a single surrogate model. CAL-SAPSO [34], DDEA-SE [35] and SHPSO [38] are the surrogate-assisted metaheuristics with multiple surrogate models, in which DDEA-SE utilizes the surrogate ensemble technique, SHPSO utilizes the hierarchical framework, and CAL-SAPSO uses the ensemble technique in the hierarchical framework. All parameters used in these algorithms are the same as those given in their papers.

Parameter settings

Each algorithm will be run independently for 20 times. The Wilcoxon rank sum test with Bonferroni correction for a significance level of 0.05 will be applied to assess whether the performance of a solution obtained by one of the two compared algorithms is expected to be better than the other. The signs, +, \approx , and – represent that the proposed BiS-SAHA is significantly better, equivalent to, and worse than the compared algorithms, respectively, according to the Wilcoxon rank sum test on the median fitness values.

The parameters used in our proposed BiS-SAHA algorithm are set as follows: the total number of exact expensive fitness evaluations $MaxFE$, which is the stopping criterion of each algorithm, is set the same as other algorithms given in their papers to give a fair comparison with them, i.e., $11 \times D$ on 10-, 20- and 30-dimensional problems, and 1000 on 50- and 100-dimensional ones. $\lfloor \frac{5}{11} \times MaxFE \rfloor$ initial solutions will be generated and evaluated using the exact expensive fitness function. $\lfloor \frac{1}{11} \times MaxFE \rfloor$ fitness evaluations will be allowed to be spent in the first stage, which is used as the condition to transfer from the first stage to the second one, and the remainder $MaxFE - \lfloor \frac{5}{11} \times MaxFE \rfloor - \lfloor \frac{1}{11} \times MaxFE \rfloor$ fitness evaluations will be spent in the second stage. The population size of the SL-PSO is set the same to the size of the initial archive, i.e., $\lfloor \frac{5}{11} \times MaxFE \rfloor$. The data in DB will be classified into 10 clusters for selection of the initial population in the first stage. The maximum iteration of the SL-PSO for each global search is set to 100, and other parameters used in SL-PSO are the same as those given in [2]. In the local search, the number of the closest data to the best position found so far is set to $N_n = \lfloor \frac{D}{2} \rfloor$. Both the scaling factor F and the mutation probability C of the DE algorithm are set to 0.8. The population size and the maximum number of iteration for each local search are set to $5 \times D$ and 150, respectively.

Parameter sensitivity analysis of the BiS-SAHA

The number of clusters k and the number of exact fitness evaluation costed in the first stage are two key parameters that may affect the performance the proposed method. Therefore,

Table 1 Test problems

	Problem	Global optimum	Characteristics
F1	Ellipsoid	0	Unimodal
F2	Rosenbrock	0	Multimodal with narrow valley
F3	Ackley	0	Multimodal
F4	Griewank	0	Multimodal
F5	Rastrigin	0	Multimodal
F6	Shifted rotated Rastrigin (F10 in [28])	−330	Very complicated multimodal
F7	Rotated hybrid composition function (F19 in [28])	10	Very complicated multimodal

we conduct experiments on 5-, 10-, 20-, and 30-dimensional F1 (unimodal), F5 (multimodal) and F7 (very complicated multimodal) test problems using different values on these two parameters and try to get proper parameter values for BiS-SAHA. Figure 4 shows the performance of BiS-SAHA using 5, 10, 20 and 30 clusters for selecting individuals of the initial population in the global search. From Fig. 4, we can see that the performance of BiS-SAHA is best when the archive DB is classified into 10 clusters on 10-, 20-, and 30-dimensional F1, F5 and F7 test problems. However, 5 clusters is much better for 5-dimensional problems. Therefore, $k = 10$ is suggested to be used in our method for problems with dimension is larger than 10, and $k = 5$ for others.

The number of exact expensive fitness evaluation costed in the first stage means how many rounds the global search will be conducted in the first stages. The global search in our method is used to explore the whole decision space to find the optimal solution of the surrogate model. However, we choose the solution with the maximum uncertainty instead of the solution with the minimum approximated value in the last generation of the global search to be evaluated using the exact expensive fitness function, which is expected to improve the correctness of the model to outline the fitness landscape. Therefore, the number of exact expensive fitness evaluation spent in the first stage is significant. From Fig. 5, we can see that when we spend $\frac{1}{11}$ of the maximum number of fitness evaluations in the first stage, the performance of BiS-SAHA is better than other settings on F1, F5 and F7 with 5, 10, 20, and 30 dimensions. Especially, the method spending $\frac{1}{11}$ of the maximum number of fitness evaluations in the first stage obtains best performance on all 5-dimensional and 30-dimensional F1, F5 and F7 problems. Therefore, in our method, $\frac{1}{11}$ of the maximum number of fitness evaluations will be conducted in the first stage.

Experimental results in different strategies

Comparisons on the methods using different frameworks

Two stages are used in our proposed method, in which the global search will be repeatedly conducted in the first stage to

explore the decision space to find a region where an optimal solution may locate. The global and local searches will then be run by turn in the second stage to balance the exploitation and exploration of the search. To see whether it is an efficient way to be organized like that in this paper, we conduct some comparison experiments on F1 (unimodal) and F5 (multimodal) benchmark problems with the methods that only the first stage is used (denoted as FS), that only the second stage is used (denoted as SS), and that the first and second stages are swapped (denoted as BiS-SAHA-swapped). Table 2 shows the statistical results obtained by the algorithms with different frameworks on F1 and F5 problems with 10, 20 and 30 dimensions. From Table 2, we can see that the results obtained by our proposed method on these problems are not worse than any other three methods, which shows that the framework of our method is best compared to other frameworks.

Comparisons on methods using different search algorithms

The SL-PSO and DE algorithms are utilized to assist the global search and local search, respectively, in the proposed BiS-SAHA. To see the efficiency of the method, we further compare BiS-SAHA to its three variants which are BiS-SAHA using the SL-PSO in both the global and local search (denoted as BiS-SAHA-SL-PSO), BiS-SAHA using the DE in both the global and local search (denoted as BiS-SAHA-DE), and BiS-SAHA using the DE and SL-PSO in the global and local search, respectively (denoted as BiS-SAHA-DE-SL-PSO). The statistical results obtained by these four algorithms on F1 and F5 are given in Table 3. From Table 3, we can see that our BiS-SAHA obtains better or competitive results than all other three variants. More specifically, the BiS-SAHA gets 5/6, 3/6, and 4/6 better results than BiS-SAHA-DE, BiS-SAHA-SL-PSO and BiS-SAHA-DE-SL-PSO, respectively. Furthermore, the proposed BiS-SAHA algorithm obtains no worse results than all of the other three algorithms. Thus, we can get that the strategy used in BiS-SAHA is efficient for solving expensive optimization problems.

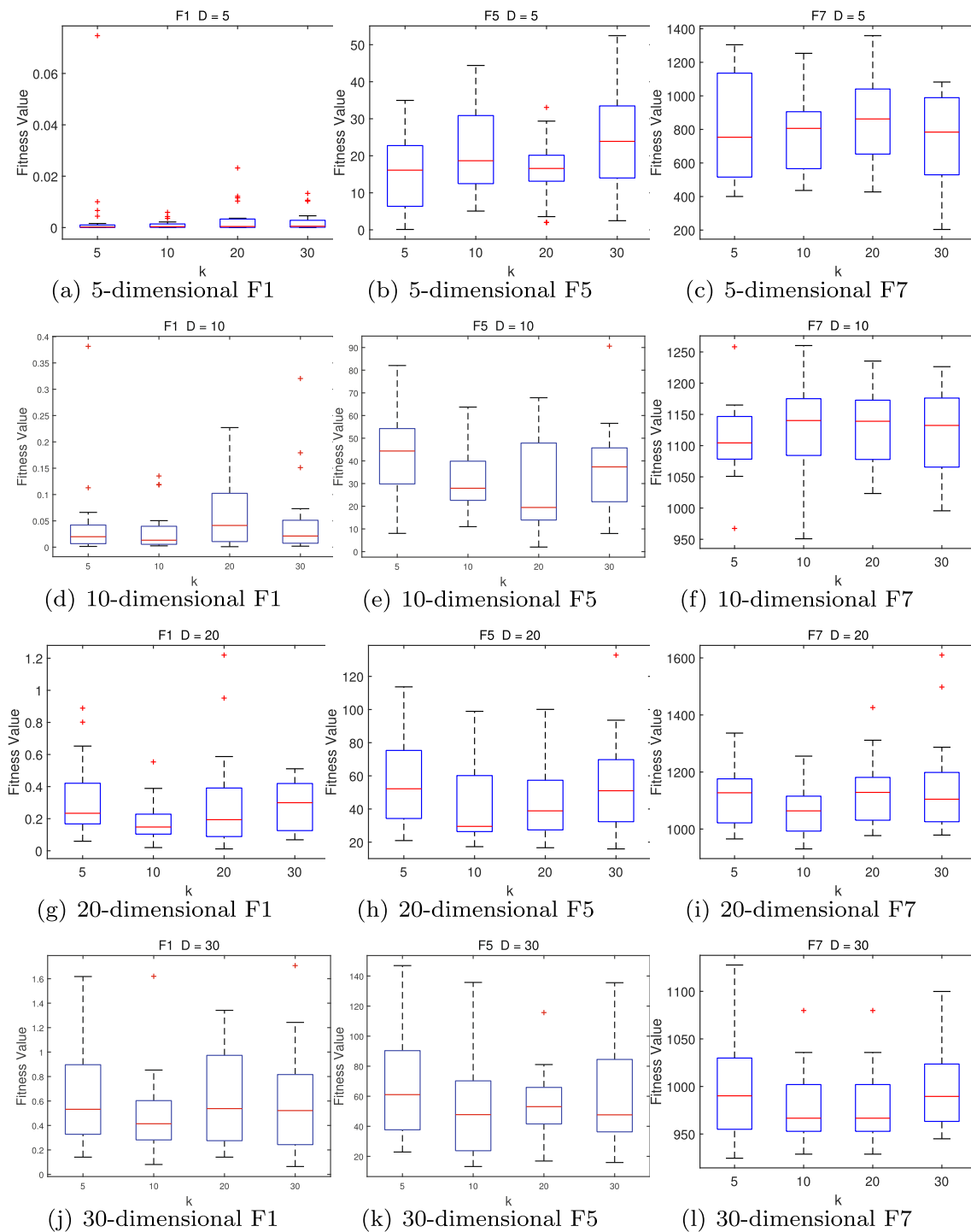


Fig. 4 The Performances of BiS-SAHA with different number of clusters used for selecting individuals of the initial population

Comparisons on the methods with and without clusters in the global search

In BiS-SAHA, the archive DB is classified into a number of clusters, which is provided for data selection to form the initial population of SL-PSO. To see the contribution of the clustering, we compare two methods, called BiS-SAHA-RG

and BiS-SAHA-RS, respectively, where BiS-SAHA-RG represents that the initial population of SL-PSO is generated randomly in the decision space, and BiS-SAHA-RS denotes that the individuals of an initial population are randomly selected from the archive DB . Figure 6 plots the convergence trends of different methods on 20-dimensional F1 and F5 test problems to generate the initial population of the global

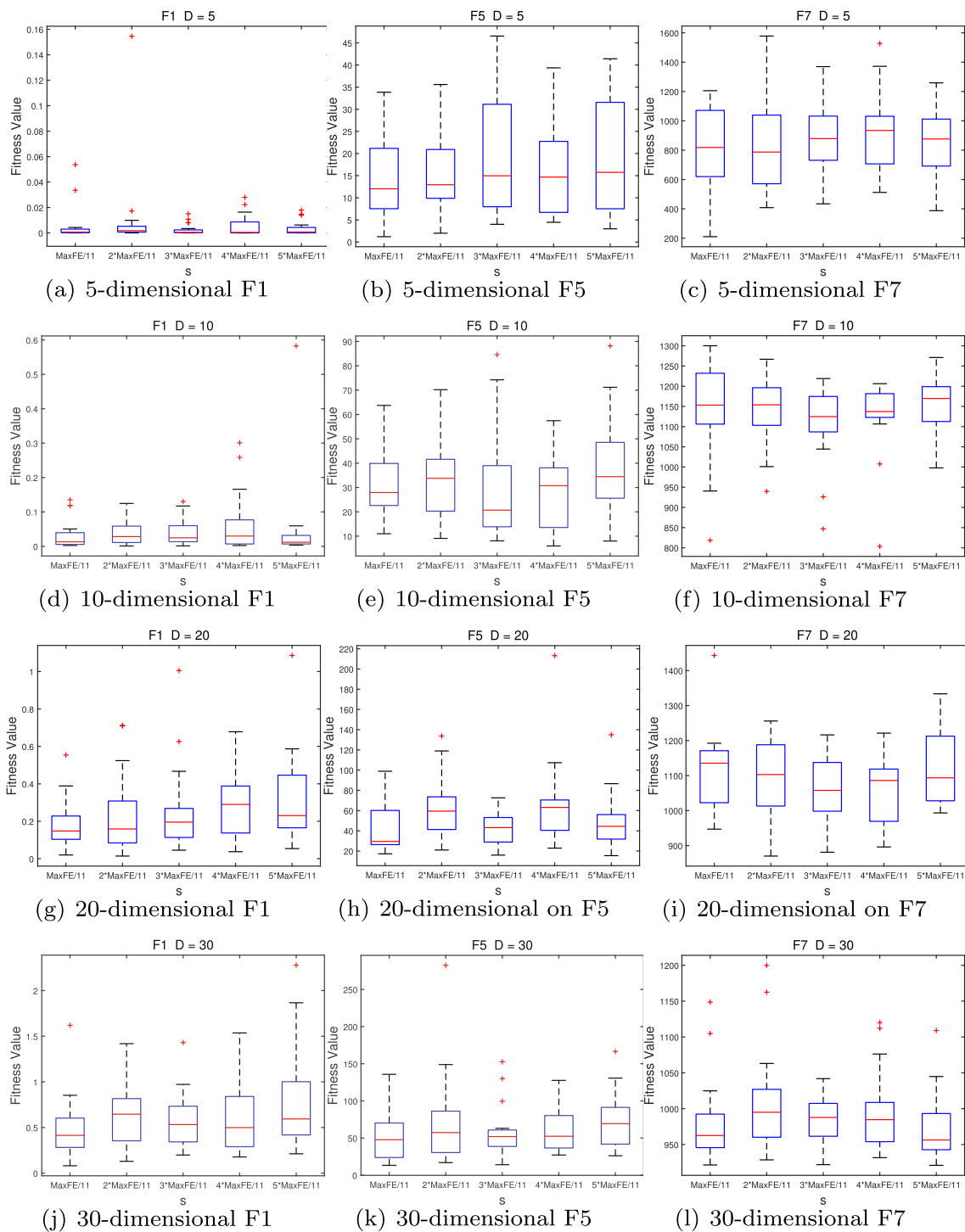


Fig. 5 The Performances of BiS-SAHA with different number of fitness evaluations expended in the first stage

search. From Fig. 6a, we can see that there is no obvious difference between SAHA-RG and the other two algorithms on the unimodal F1 problem. However, it is clearly shown in Fig. 6b that on the multimodal F5 problem, our proposed BiS-SAHA method can obtain better results and has quicker convergence speed than both BiS-SAHA-RG and

BiS-SAHA-RS. Therefore, we can get that the diversity of the initial population is better than randomly selecting in the decision space or the archive *DB* for the global search by selecting individuals round in turns from the clusters.

Table 2 The statistical results (median and median absolute deviation) obtained by different algorithms with different frameworks on F1 and F5 problems with 10, 20 and 30 dimensions

	D	BiS-SAHA	FS	SS	BiS-SAHA-swap
F1	10	1.33E−02 (4.20E−02)	4.30E−02(4.01E−02)[+]	1.80E−02(4.55E−02)[≈]	6.29E−02(7.53E−02)[+]
	20	1.48E−01 (1.29E−01)	1.85E−01(1.42E−01)[≈]	3.68E−01(3.29E−01)[+]	3.78E−01(1.93E−01)[+]
	30	4.14E−01 (3.46E−01)	6.68E−01(3.92E−01)[+]	6.66E−01(6.04E−01)[+]	6.65E−01(4.56E−01)[+]
F5	10	2.79E+01(1.52E+01)	2.69E+01 (2.23E+01)[≈]	4.10E+01(1.52E+01)[+]	3.16E+01(3.08E+01)[≈]
	20	2.95E+01 (2.50E+01)	4.66E+01(2.96E+01)[+]	4.94E+01(3.81E+01)[≈]	5.03E+01(3.91E+01)[+]
	30	4.78E+01 (4.03E+01)	5.68E+01(4.14E+01)[≈]	6.06E+01(7.03E+01)[≈]	7.33E+01(2.21E+01)[+]
+ / ≈ / −			3 / 3 / 0	3 / 3 / 0	5 / 1 / 0
p value			2.52E−01	1.96E−01	3.74E−02

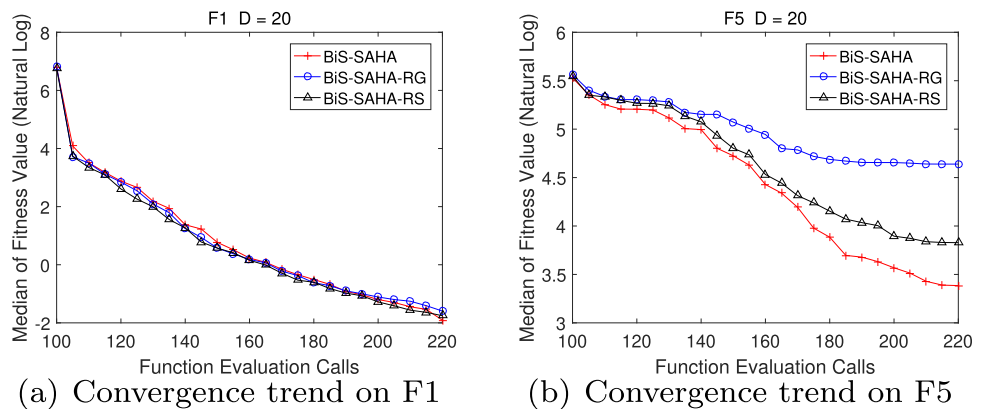
FS and SS represent the method that only the first stage and second stage is utilized, respectively. BiS-SAHA-swapped is a BiS-SAHA variant that the first and second stages are swapped. The best median results in each row is highlighted

Table 3 The statistical results (median and median absolute deviation) obtained by BiS-SAHA and its three variants on F1 and F5 problems with 10, 20 and 30 dimensions

	D	BiS-SAHA	BiS-SAHA-DE	BiS-SAHA-SL-PSO	BiS-SAHA-DE-SL-PSO
F1	10	1.33E−02(4.20E−02)	1.03E−02(2.13E−02)[≈]	2.85E−02(1.67E−01)[≈]	8.39E−03 (1.06E−02)[≈]
	20	1.48E−01 (1.29E−01)	2.75E−01(3.58E−01)[+]	3.81E−01(3.17E−01)[+]	1.86E−01(1.12E−01)[≈]
	30	4.14E−01 (3.46E−01)	4.67E+00(2.53E+00)[+]	1.07E+00(2.02E+00)[+]	2.45E+00(1.02E+00)[+]
F5	10	2.79E+01 (1.52E+01)	6.51E+01(1.91E+01)[+]	4.45E+01(2.75E+01)[+]	6.14E+01(1.76E+01)[+]
	20	2.95E+01 (2.50E+01)	1.11E+02(2.57E+01)[+]	4.38E+01(3.82E+01)[≈]	1.32E+02(2.14E+01)[+]
	30	4.78E+01 (4.03E+01)	2.63E+02(2.87E+01)[+]	4.84E+01(6.28E+01)[≈]	1.55E+02(3.83E+01)[+]
+ / ≈ / −			5 / 1 / 0	3 / 3 / 0	4 / 2 / 0
p value			4.17E−02	2.46E−01	8.25E−02

BiS-SAHA-DE is a BiS-SAHA variant that uses DE in both the global and local search, BiS-SAHA-SL-PSO is a BiS-SAHA variant that uses SL-PSO in both the global and local search; and BiS-SAHA-DE-SL-PSO is a BiS-SAHA variant using DE and SL-PSO in the global and local search, respectively. The best median results in each row is highlighted

Fig. 6 The convergence trends of different methods to generate the initial population of the global search



Performance comparisons with other algorithms

To evaluate the performance of our proposed method, in this subsection, we further conduct experiments on five benchmark problems with 10, 20, and 30 dimensions, and on six problems with 50 and 100 dimensions, respectively. The experimental results are compared to some state-of-the-art algorithms, including GORS-SSLPSO [37], CAL-SAPSO

[34], DDEA-SE [35], SHPSO [38], and MGP-SLPSO [33]. The statistical median value of the optimal solutions obtained by all algorithms on 20 independent runs are given in Tables 4 and 5, respectively. Table 4 shows the results on 10-, 20-, and 30-dimensional problems, while Table 5 gives the results on 50- and 100-dimensional problems. The best results among each row will be highlighted.

From Tables 4 and 5, we can see that our proposed BiS-SAHA method can obtain better or competitive results than the other five algorithms proposed for expensive problems. In Table 4, our proposed BiS-SAHA method is only compared to GORS-SSLPSO, CAL-SAPSO and DDEA-SE. The SHPSO and MGP-SLPSO methods are not included in Table 4 because both of them were proposed specially for solving high-dimensional problems. From Table 4, we can see that compared to CAL-SAPSO and DDEA-SE that utilize the ensemble techniques, our BiS-SAHA method can obtain 12 and 11 better results, respectively, among all 15 benchmark problems. Compared to GORS-SSLPSO, the BiS-SAHA algorithm gets only 1/15 worse result on problems with lower dimensions (not more than 30 dimensions). From Table 5, we can see that the CAL-SAPSO algorithm does not compare to our method on high-dimensional problems (not less than 50 dimensions). The reason is that CAL-SAPSO was proposed for solving the lower dimensional problems. As seen from Table 5, we can see that our BiS-SAHA method can obtain 8/12, 7/12, and 11/12 better results, respectively, than SHPSO, MGP-SLPSO and DDEA-SE. While compared to GORS-SSLPSO, BiS-SAHA achieves six better results of 12 high-dimensional problems, from which we can further find that BiS-SAHA obtains 4/6 better results on 100-dimensional problems.

Furthermore, we also plot the convergence trends of each method on 30- and 100-dimensional test problems, which are shown in Figs. 7 and 8, respectively. From Fig. 7, we can see that our proposed BiS-SAHA converge much quicker than both GORS-SSLPSO and CAL-SAPSO on most of lower-dimensional F1–F5 problems. From Fig. 8, we can find that MGP-SLPSO converges much quicker than others on F1 and F4 problems and GORS-SSLPSO does better on F2. The MGP-SLPSO is pretty good for unimodal high-dimensional problems. However, for the multimodal problems, our BiS-SAHA method has better convergence characters on four among five test problems. Also, from Fig. 8, we can see that the proposed BiS-SAHA obtains better results on F6 and F7, which are very complicated multimodal problems, than others, showing the efficiency of BiS-SAHA for solving multimodal problems.

Optimization of the Lennard–Jones potential problem

To investigate the performance of BiS-SAHA, we further conducted an experiment on a molecular conformation problem to minimize the Lennard–Jones potential [4,6]. In the Lennard–Jones potential optimization problem, the potential energy of a molecule is modeled by the sum of the potential energies between pairs of atoms. The mathematical model of the Lennard–Jones potential optimization is given in the following:

Table 5 The statistical results (median and median absolute deviation) obtained by BiS-SAHA, GORS-SSLPSO, SHPSO, MGP-SLPSO and DDEA-SE on F1–F4 and F6–F7 problems with 50 and 100 dimensions

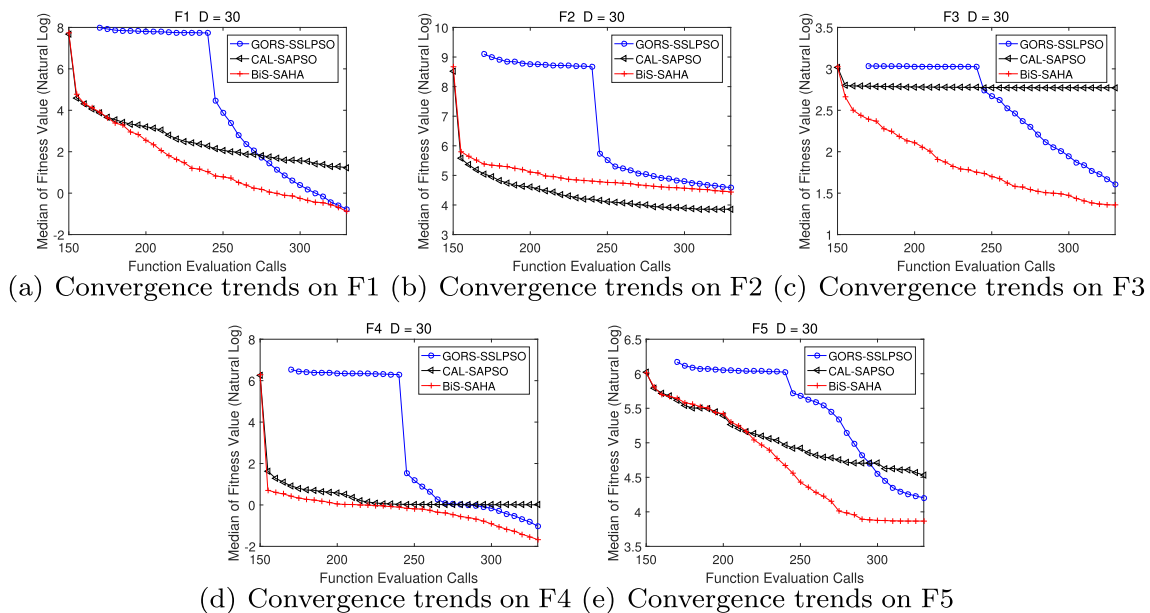
	D	BiS-SAHA	GORS-SSLPSO	SHPSO	MGP-SLPSO	DDEA-SE
F1	50	7.22E-02(6.36E-02)	1.77E-01(1.48E+02)[+]	6.45E+00(2.57E+00)[+]	4.32E-26 (1.27E-26)[-]	1.44E+01(3.76E+00)[+]
	100	7.11E+00(2.96E+00)	9.43E+00(3.19E+02)[+]	1.18E+02(2.54E+01)[+]	2.93E-4 (5.78E-03)[-]	2.91E+02(6.67E+01)[+]
F2	50	5.33E+01(2.17E+00)	4.82E+01 (1.23E+02)[-]	5.07E+01(1.73E+00)[-]	8.50E+01(8.81E+00)[+]	9.39E+01(5.39E+00)[+]
	100	3.68E+02(7.15E+01)	9.76E+01 (1.08E+01)[-]	1.95E+02(4.80E+01)[-]	6.07E+02(1.05E+02)[+]	2.35E+02(2.91E+01)[-]
F3	50	1.96E+00 (8.28E-01)	3.52E+00(9.44E-01)[+]	2.46E+00(3.18E-01)[+]	1.33E+01(1.96E+00)[+]	4.22E+00(3.26E-01)[+]
	100	4.02E+00 (7.26E-01)	8.80E+00(1.79E+00)[+]	5.19E+00(6.24E-01)[+]	1.81E+01(3.47E-01)[+]	6.90E+00(6.45E-01)[+]
F4	50	2.62E-02(1.21E-02)	1.04E-04 (4.17E-03)[-]	9.54E-01(8.78E-02)[+]	1.52E-02(4.14E-03)[-]	1.95E+00(3.01E-01)[+]
	100	6.05E-01(6.50E-02)	1.97E-01(2.76E+01)[-]	1.12E+00(4.50E-02)[+]	7.25E-02 (2.51E-02)[-]	1.69E+01(2.69E+00)[+]
F6	50	1.61E+02(2.81E+01)	-7.00E+01 (5.31E+01)[-]	1.25E+02(3.00E+01)[-]	4.74E+01(5.30E+01)[-]	7.10E+02(4.63E+01)[+]
	100	4.71E+02 (2.23E+02)	1.00E+03(1.30E+02)[+]	7.66E+02(7.58E+01)[+]	9.65E+02(1.09E+02)[+]	1.81E+03(1.29E+02)[+]
F7	50	9.91E+02 (3.83E+01)	1.00E+03(6.62E+01)[≈]	9.93E+02(2.28E+01)[≈]	1.15E+03(3.29E+01)[+]	1.19E+03(1.98E+01)[+]
	100	1.18E+03 (6.92E+01)	1.43E+03(2.36E+01)[+]	1.42E+03(5.00E+01)[+]	1.41E+03(4.12E+01)[+]	1.25E+03(5.58E+01)[+]
+/≈/-			6/1/5	8/1/3	7/0/5	11/0/1
p value			2.14E-01	4.09E-02	6.54E-05	1.09E-05

The best median results in each row is highlighted

Table 4 The statistical results (median and median absolute deviation) obtained by BiS-SAHA, GORS-SSLPSO, CAL-SAPSO and DDEA-SE on F1–F5 problems with 10, 20 and 30 dimensions

	D	BiS-SAHA	GORS-SSLPSO	CAL-SAPSO	DDEA-SE
F1	10	1.33E−02 (4.20E−02)	2.75E−02(5.35E−02)[≈]	8.84E−01(9.95E−01)[+]	9.38E−01(4.64E−01)[+]
	20	1.48E−01(1.29E−01)	1.45E−01 (2.12E−01)[≈]	1.60E+00(5.15E−01)[+]	2.40E+00(9.90E−01)[+]
	30	4.14E−01 (3.46E−01)	4.62E−01(3.85E−01)[≈]	3.45E+00(1.47E+00)[+]	4.39E+00(1.23E+00)[+]
F2	10	1.43E+01 (4.99E+00)	1.62E+01(8.08E+00)[≈]	1.49E+01(3.54E+00)[≈]	2.56E+01(5.29E+00)[+]
	20	5.26E+01(2.01E+01)	5.44E+01(1.54E+01)[≈]	3.39E+01 (5.57E+00)[−]	4.39E+01(5.69E+00)[≈]
	30	8.48E+01(1.87E+01)	9.85E+01(2.14E+01)[≈]	4.73E+01 (9.98E+00)[−]	5.64E+01(7.18E+00)[−]
F3	10	5.68E+00(5.87E+00)	3.49E+00 (2.77E+00)[−]	2.01E+01(4.43E−01)[+]	5.84E+00(8.87E−01)[≈]
	20	4.64E+00(1.52E+00)	4.28E+00 (1.92E+00)[≈]	2.01E+01(3.61E−15)[+]	5.07E+00(6.21E−01)[+]
	30	3.88E+00 (1.32E+00)	4.97E+00(1.28E+00)[+]	1.60E+01(5.62E−01)[+]	4.72E+00(3.78E−01)[+]
F4	10	8.13E−01 (1.71E−01)	8.45E−01(1.58E−01)[≈]	1.13E+00(1.14E−01)[+]	1.30E+00(1.54E−01)[+]
	20	3.35E−01 (1.72E−01)	3.46E−01(1.35E−01)[≈]	1.04E+00(3.57E−02)[+]	1.22E+00(8.64E−02)[+]
	30	1.87E−01 (9.77E−02)	3.58E−01(1.13E−01)[+]	1.01E+00(4.36E−02)[+]	1.23E+00(9.87E−02)[+]
F5	10	2.79E+01(1.52E+01)	2.32E+01 (1.93E+01)[≈]	8.99E+01(2.37E+01)[+]	5.23E+01(2.78E+01)[+]
	20	2.95E+01 (2.50E+01)	4.36E+01(2.01E+01)[≈]	7.36E+01(1.53E+01)[+]	4.39E+01(5.69E+00)[≈]
	30	4.78E+01 (4.03E+01)	6.68E+01(3.06E+01)[≈]	9.29E+01(1.93E+01)[+]	1.13E+02(2.81E+01)[+]
	+/≈/−		2/12/1	12/1/2	11/3/1
	p value		3.52E−01	2.75E−02	6.06E−02

The best median results in each row is highlighted

**Fig. 7** Convergence profiles of GORS-SSLPSO, CAL-SAPSO and BiS-SAHA on benchmark problems with 30 dimensions

$$V = \sum_{i=1}^{N-1} \sum_{j=1}^N (d_{ij}^{-12} - 2d_{ij}^{-6}), \quad (9)$$

where N is the number of atoms in the molecule, d_{ij} represents the Euclidean distance between two points \mathbf{p}_i and \mathbf{p}_j , where each point \mathbf{p}_i , $i = 1, 2, \dots, N$ has three dimensions. Therefore, the dimension of an optimization problem

is three times the number of atoms in the given molecule. In our experiment, ten atoms is adopted. Thus, the number of decision variable is 30. The upper and lower bounds of each dimension is set to $[0, 4]$, $[0, 4]$, and $[0, \pi]$, respectively.

Table 6 gives the statistical results obtained by 20 independent runs on the Lennard–Jones potential problem with 30 decision variables. The total number of exact fitness evaluations is set to 330. ‘Mean’, ‘median’, ‘worst’, and ‘best’

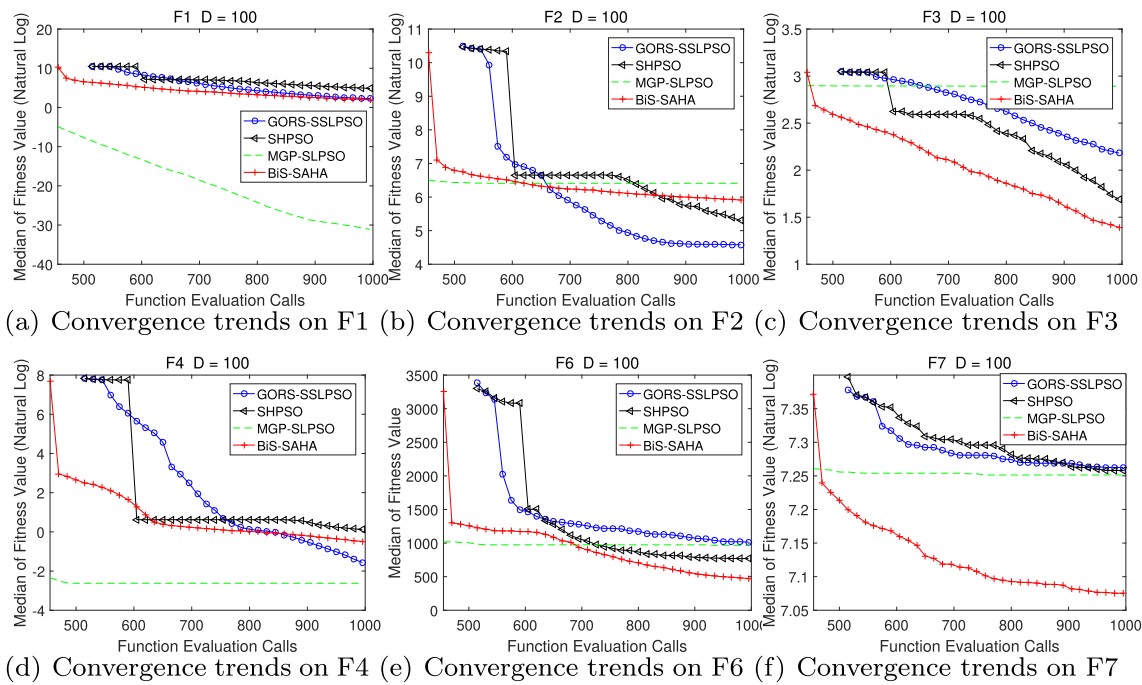


Fig. 8 Convergence profiles of GORS-SSLPSO, SHPSO, MGP-SLPSO and BiS-SAHA on benchmark problems with 100 dimensions

represent the mean, the median, the worst, and the best result among the results of 20 independent runs, respectively. ‘std’ represents the standard deviation of these 20 results. From Table 6, we can see that compared to GORS-SSLPSO, CAL-SAPSO and DDEA-SE, the proposed BiS-SAHA method obtains better results on the best, the worst, the mean, and the median values among the best solutions found so far of 20 independent runs, respectively, which further show the competitive performance of our proposed BiS-SAHA.

Optimization of a constrained optimization problem with nonlinear constraints

In the real-world engineering applications, most optimization problems have a number of constraints. Therefore, in this paper, we also evaluate the performance of our proposed method by conducting experiment on g07 test problem given in CEC 2006 special session on constrained real-parameter optimization [18]. In g07 problems, there are totally eight linear or nonlinear inequality constraints. The dimension of g07 is 10. The mathematical model of g07 is given in Eq. (10).

$$\begin{aligned} \min f(\mathbf{x}) &= x_1^2 + x_2^2 + x_1x_2 - 14x_1 \\ &\quad - 16x_2 + (x_3 - 10)^2 \\ &\quad + 4(x_4 - 5)^2 + (x_5 - 3)^2 + \\ &\quad 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 \\ &\quad + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 \\ \text{s.t. } g_1(\mathbf{x}) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0 \end{aligned}$$

$$\begin{aligned} g_2(\mathbf{x}) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \\ g_3(\mathbf{x}) &= -8x_1 + 2x_2 \\ &\quad + 5x_9 - 2x_{10} - 12 \leq 0 \\ g_4(\mathbf{x}) &= 3(x_1 - 2)^2 \\ &\quad + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0 \\ g_5(\mathbf{x}) &= 5x_1^2 + 8x_2 \\ &\quad + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \\ g_6(\mathbf{x}) &= x_1^2 \\ &\quad + 2(x_2 - 2)^2 - 2x_1x_2 \\ &\quad + 14x_5 - 6x_6 \leq 0 \\ g_7(\mathbf{x}) &= 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 \\ &\quad + 3x_5^2 - x_6 - 30 \leq 0 \\ g_8(\mathbf{x}) &= -3x_1 + 6x_2 \\ &\quad + 12(x_9 - 8)^2 - 7x_{10} \leq 0 \\ &\quad - 10 \leq x_i \leq 10, \quad i = 1, 2, \dots, 10. \end{aligned} \tag{10}$$

In the experiment, 20 independent runs are conducted on this problem and the penalty function is utilized to transform the constrained problem to an unconstrained one so that it can be solved by our proposed method. Eq. (11) gives the unconstrained problem that will be solved by different algorithms.

$$G(\mathbf{x}) = f(\mathbf{x}) + \rho \sum_{i=1}^8 \max\{g_i(\mathbf{x}), 0\}, \tag{11}$$

Table 6 The statistical results obtained by BiS-SAHA, GORS-SSLPSO, CAL-SAPSO and DDEA-SE on the Lennard–Jones potential problem

	Mean	Median	Worst	Best	Std
BiS-SAHA	−3.25E+00	−3.03E+00	−1.83E+00	−6.36E+00	1.06E+00
GORS-SSLPSO	−2.12E+00	−2.05E+00	−1.53E+00	−3.38E+00	4.63E−01
CAL-SAPSO	−2.42E+00	−2.34E+00	−1.66E+00	−3.34E+00	5.34E−01
DDEA-SE	3.17E+05	−3.33E−01	6.25E+06	−2.30E+00	1.40E+06

The best results in each column is highlighted

Table 7 The statistical results obtained by BiS-SAHA, GORS-SSLPSO, CAL-SAPSO and DDEA-SE on g07 benchmark problem of CEC 2006 special session

	N_{fs}	Mean	Median	Worst	Best	Std
BiS-SAHA	5	1.09E+03	1.28E+03	1.43E+03	3.17E+02	4.46E+02
GORS-SSLPSO	3	1.58E+03	1.65E+03	2.40E+03	6.81E+02	8.62E+02
CAL-SAPSO	0	–	–	–	–	–
DDEA-SE	0	–	–	–	–	–

The best results in each column is highlighted

where ρ is the penalty function. In our experiment, ρ is set to 10^{15} . Table 7 gives the statistical results obtained by GORS-SSLPSO, CAL-SAPSO, DDEA-SE and our proposed BiS-SAHA method on g07 benchmark problem of CEC 2006 special session. N_{fs} is the times that the algorithm finds the feasible solution. ‘Mean’, ‘Median’, ‘Worst’ and ‘Best’ represent the mean, the median, the worst and the best fitness value of the feasible optimal solutions, respectively. ‘Std’ is the standard variance. ‘–’ represents the method does not find a feasible solution. From Table 7, we can see that our proposed BiS-SAHA method can find a feasible solution for five times among 20 independent runs, GORS-SSLPSO can obtain a feasible solution for three times, while both CAL-SAPSO and DDEA-SE are not able to get feasible solutions among all 20 independent runs. Furthermore, compared to GORS-SSLPSO, our proposed BiS-SAHA can obtain better feasible solution than GORS-SSLPSO.

Conclusion

A bi-stage surrogate-assisted hybrid algorithm was proposed in this paper to solve computationally expensive problems. The proposed method tried to explore the decision space by searching for the optimal solution of the surrogate ensemble to improve the approximation accuracy of the sub-space where the optimal solution may locate in the first stage. In the second stage, the local search conducted by an RBF-assisted DE algorithm is run by turn with the global search to speed up finding the optimal solution. The comparisons on the experimental results on seven benchmark problems, the Lennard–Jones potential problem and g07 problem with nonlinear constraints showed the better performance of our proposed method than others. However, only global surrogate models are built in both the global and local searches, which,

we think, may affect the precision of locating at the optimal solution. Therefore, in the future, local surrogate models will be considered to improve the precision of the optimal solution location.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Cai X, Gao L, Li X, Qiu H (2019) Surrogate-guided differential evolution algorithm for high dimensional expensive problems. *Swarm Evol Comput* 48:288–311
2. Cheng R, Jin Y (2015) A social learning particle swarm optimization algorithm for scalable optimization. *Inf Sci* 291:43–60
3. Clarke SM, Griebisch JH, Simpson TW (2004) Analysis of support vector regression for approximation of complex engineering analyses. *J Mech Des* 127(6):1077–1087
4. Das S, Suganthan PN (2010) Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. *Jadavpur University, Nanyang Technological University, Kolkata*, pp 341–359

5. Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. *IEEE Trans Evol Comput* 15(1):4–31
6. Deep K, Arya M (2010) Minimization of Lennard-Jones potential using parallel particle swarm optimization algorithm. In: Ranka S, Banerjee A, Biswas KK, Dua S, Mishra P, Moona R, Poon SH, Wang CL (eds) *Contemporary computing*. Springer, Berlin, Heidelberg, pp 131–140
7. Diazmanriquez A, Toscano G, Coello CAC (2016) Comparison of metamodeling techniques in evolutionary algorithms. *Soft Comput* 21(19):5647–5663
8. Ferrari S, Stengel R (2005) Smooth function approximation using neural networks. *IEEE Trans Neural Netw* 16(1):24–38
9. Gonzalez J, Rojas I, Ortega J, Pomares H, Fernandez F, Daz A (2003) Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. *IEEE Trans Neural Netw* 14(6):1478–1495
10. Gutmann HM (2000) A radial basis function method for global optimization. *J Global Optim* 19(3):201–227
11. Jin R, Chen W, Simpson TW (2001) Comparative studies of metamodeling techniques under multiple modelling criteria. *Struct Multidiscip Optim* 23(1):1–13
12. Jin Y (2011) Surrogate-assisted evolutionary computation: recent advances and future challenges. *Swarm Evol Comput* 1(2):61–70
13. Karahan H, Ceylan H, Ayvaz MT (2007) Predicting rainfall intensity using a genetic algorithm approach. *Hydrol Process* 21(4):470–475
14. Kattan A, Agapitos A, Ong YS, Alghamedi AA, O'Neill M (2016) GP made faster with semantic surrogate modelling. *Inf Sci* 355–356:169–185
15. Kennedy J, Eberhart R (2002) Particle swarm optimization. In: *Proceedings of ICNN'95—International Conference on Neural Networks* (2002)
16. Lameijer E, Back T, Kok JN, Ijzerman AP (2005) Evolutionary algorithms in drug design. *Nat Comput* 4(3):177–243
17. Lian Y, Liou M (2005) Multiobjective optimization using coupled response surface model and evolutionary algorithm. *AIAA J* 43(6):1316–1325
18. Liang J, Runarsson T, Mezura-Montes E, Clerc M, Suganthan P, Coello C, Deb K (2006) Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. Nanyang Technological University, Singapore, Tech. Rep, vol 41
19. Lim D, Jin Y, Ong Y, Sendhoff B (2010) Generalizing surrogate-assisted evolutionary computation. *IEEE Trans Evol Comput* 14(3):329–355
20. Liu B, Zhang Q, Gielen G (2014) A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Trans Evol Comput* 18(2):180–192
21. Lu X, Tang K (2012) Classification- and regression-assisted differential evolution for computationally expensive problems. *J Comput Sci Technol* 27(5):1024–1034
22. Paenke I, Branke J, Jin Y (2006) Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation. *IEEE Trans Evol Comput* 10(4):405–420
23. Regis R (2014) Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions. *IEEE Trans Evol Comput* 18:326–347
24. Schneider PI, Santiago XG, Rockstuhl C, Burger S (2017) Global optimization of complex optical structures using Bayesian optimization based on Gaussian processes. In: Kress BC, Schelkens P (eds) *Digital Optical Technologies 2017*, vol 10335. International Society for Optics and Photonics, SPIE, Washington, pp 141–149
25. Simon D (2009) Biogeography-based optimization. *IEEE Trans Evol Comput* 12:702–713
26. Stein ML (1987) Large sample properties of simulations using Latin hypercube sampling. *Technometrics* 29(2):143–151
27. Storn R, Price K (1997) Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 11(4):341–359
28. Suganthan P, Hansen N, Liang J, Deb K, Chen Yp, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *Nat Comput* 341–357
29. Sun C, Ding J, Zeng JC, Jin Y (2016) A fitness approximation assisted competitive swarm optimizer for large scale expensive optimization problems. *Memet Comput*
30. Sun C, Jin Y, Cheng R, Ding J, Zeng J (2017) Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Trans Evol Comput* 21(4):644–660
31. Sun C, Jin Y, Zeng J, Yu Y (2015) A two-layer surrogate-assisted particle swarm optimization algorithm. *Soft Comput* 19(6):1461–1475
32. Sun C, Zeng J, Pan J, Xue S, Jin Y (2013) A new fitness estimation strategy for particle swarm optimization. *Inf Sci* 221:355–370
33. Tian J, Tan Y, Zeng J, Sun C, Jin Y (2019) Multiobjective infill criterion driven Gaussian process-assisted particle swarm optimization of high-dimensional expensive problems. *IEEE Trans Evol Comput* 23(3):459–472
34. Wang H, Jin Y, Doherty J (2017) Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Trans Cybern*:1–14
35. Wang H, Jin Y, Sun C, Doherty J (2019) Offline data-driven evolutionary optimization using selective surrogate ensembles. *IEEE Trans Evol Comput* 23(2):203–216
36. Younis A, Dong Z (2010) Trends, features, and tests of common and recently introduced global optimization methods. *Eng Optim* 42(8):691–718
37. Yu H, Tan Y, Sun C, Zeng J (2019) A generation-based optimal restart strategy for surrogate-assisted social learning particle swarm optimization. *Knowl Based Syst* 163:14–25
38. Yu H, Tan Y, Zeng J, Sun C, Jin Y (2018) Surrogate-assisted hierarchical particle swarm optimization. *Inf Sci* 454–455:59–72
39. Zhang Q, Liu W, Tsang E, Virginas B (2010) Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Trans Evol Comput* 14:456–474
40. Zhao D, Xue D (2009) Performance comparison of metamodeling methods from the perspective of sample quality merits. In: *35th Design Automation Conference*, vol 5. ASME
41. Zhou Z, Ong YS, Nair PB, Keane AJ, Lum KY (2007) Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Trans Syst Man Cybern Part C (Appl Rev)* 37(1):66–76

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.