A fast surrogate-assisted particle swarm optimization algorithm for computationally expensive problems

Fan Li [a], Weiming Shen [a], Xiwen Cai [a], Liang Gao [a,*], G. Gary Wang [b]

[a] *State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science & Technology, Wuhan 430074, PR China*
[b] *School of Mechatronic Systems Engineering, Simon Fraser University, BC, Canada*

ARTICLE INFO

ABSTRACT

Although many surrogate-assisted evolutionary algorithms (SAEAs) have been proposed to solve computationally expensive problems, they usually need to consume plenty of expensive evaluations to obtain an acceptable solution. In this paper, we proposed a fast surrogate-assisted particle swarm optimization (FSAPSO) algorithm to solve medium scaled computationally expensive problems through a small number of function evaluations (FEs). Two criteria are applied in tandem to select candidates for exact evaluations. The performance-based criterion is used to exploit the current global best and accelerate the convergence rate, while the uncertainty-based criterion is used to enhance the exploration of the algorithm. The distance-based uncertainty criterion in SAEAs does not consider the fitness landscape of different problems. Therefore, we developed a criterion to estimate uncertainty by considering the distance and fitness value information simultaneously. This criterion can make up for the disadvantage of the conventional distance-based uncertainty criterion by considering the fitness landscape of a problem. In addition, it can be applied in any surrogate-assisted evolutionary algorithm irrespective of the used surrogate model. Twenty-three benchmark functions widely adopted in the literature and a 10-dimension propeller design problem are used to test the proposed approach. Experimental results demonstrate the superiority of the proposed FSAPSO algorithm over seven state-of-the-art algorithms.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Evolutionary algorithms (EAs) have been successfully applied to solve various problems in different fields [1–4]. These algorithms can be used to solve black box optimization problems whose gradient information and mathematical expression are unknown. Most EAs need many FEs to find a good solution, so they are not suitable for computationally expensive problems whose FEs are time-consuming. To address this challenge, surrogate models have been applied in EAs to replace some FEs or select promising individuals for exact FEs. Such algorithms are called surrogate-assisted evolutionary algorithms (SAEAs). Commonly used surrogate models include the polynomial regression [5–7], the radial basis function (RBF) [8–11], Kriging, [5,12–14], and support vector regression [15,16]. Comparisons of different models have been made in [5,17,18]. Different types of models are suitable for their corresponding problems [19]. Characteristics of most real-world problems are often unknown. Hence, an ensemble of surrogates and hybrid surrogates are often used for better approximation and uncertainty quantification [20–22].

Generally, predictions from surrogate models can replace some exact FEs. Surrogate models can help to select individuals for exact evaluations by using some pre-screening criteria. These pre-screening criteria are quite important for SAEAs. First, not all fitness values are important in the iterative processes of EAs, so many FEs can be saved by just selecting some promising individuals for exact evaluations. Second, the pre-screening criterion, deciding on which individuals could be calculated by exact evaluations, has great effects on balancing the exploration and exploitation of an algorithm, indirectly influencing the convergence rate of an algorithm. Therefore, it is generally hard to judge how many individuals or which individuals should be exactly evaluated.

Commonly used pre-screening criteria in SAEAs can be roughly divided into three categories. First, the performance-based criterion evaluates individuals with good predicted fitness values. For example, in [23], each particle had multiple trail velocities and positions, and an RBF surrogate model was used to select a position with the minimum predicted fitness value. One drawback of this method is that all the particles are exactly evaluated at each iteration, so it needs many FEs to obtain good solutions. Sun et al. [24] used a criterion where particles with better

Nomenclature

Symbols

x	input variables
Nt	the number of training points
$y(x)$	the true function
λ	coefficients of the RBF model
c_i	the ith center of basis functions
p	a polynomial model
φ	a basis function
w	inertia factor
v_i	speed of the ith particle
c_1	learning factor
c_2	learning factor
x_i	current location of the particle i
p_i	the best position of the particle i
p_g	the global optimum position of the particle swarm.
D	dimensions of parameter space
f	expensive functions
K	the number of initial sample points
$maxNFE$	maximum NFEs
N	swarm size
η	thresholds for determining whether the sample is evaluated by f
k	the number of nearest neighbors
d_{ij}	the distance from the candidate i to its jth nearest neighbor
d_{i1}	the distance from the candidate i to its nearest neighbor
y_j	fitness value of the jth nearest neighbor
ub	upper boundary of variables
lb	the lower boundary of variables
$Vmax$	the maximum velocity of particles
$Vmin$	the minimum velocity of particles

Abbreviations

FSAPSO	fast surrogate-assisted particle swarm optimization
FEs	function evaluations
NFEs	number of function evaluations
PSO	particle swarm optimization
EAs	evolutionary algorithms
SAEAs	surrogate-assisted evolutionary algorithms
RBF	radial basis function
MSE	mean square error
SPSO	standard PSO
DB	database
DF	distance and fitness value

FSAPSO-WOMO	FSAPSO without evaluating the model optimum
FSAPSO-WOBP	FSAPSO without evaluating the best particle
FSAPSO-WOU	FSAPSO without the uncertainty-based criterion
CAL-SAPSO	surrogate-assisted PSO algorithm with the help of committee-based active learning
SAPSO-PBEST	an RBF model assisted PSO
SL-PSO	social learning-based PSO
SHPSO	surrogate-assisted hierarchical particle swarm optimizer
GORS-SSLPSO	surrogate-assisted SL-PSO with a generation-based optimal restart strategy
SA-COSO	surrogate-assisted cooperative swarm optimization
ESAO	evolutionary sampling assisted optimization

approximate fitness values than their current personal optima (PBEST-based criterion) were exactly evaluated in a two-layer surrogate-assisted particle swarm optimization algorithm. This pre-screening criterion was also used in [25–28]. Although the PBEST-based criterion can reduce some unimportant FEs, some particles with a little better improvement than their personal best will be exactly evaluated. This will also induce that many FEs are consumed in some iterations as the PBEST-based criterion has no limitation for the number of evaluated individuals. In addition, there are some algorithms that select the best individual in the population for exact evaluations [27,29–31]. However, this method cannot be used solely as the algorithm will quickly fall into the local optima of the surrogate model.

Second, the uncertainty-based criterion evaluates individuals with great uncertainties. Generally, evaluating individuals with great uncertainties can effectively improve the accuracy of a surrogate model and push the search to unexplored or not-well-explored regions in SAEAs [32,33]. The uncertainty in SAEAs usually refers to the reliability of fitness estimations, and less reliable fitness estimations indicate greater uncertainties [34]. Searching regions with complex fitness landscapes usually implies great uncertainties. In addition, the reliability of the estimated fitness values cannot be known in advance. Therefore, regions with sparse sample points are usually assumed to be with great uncertainties. However, there may be regions where samples are sparse but the estimated fitness values are accurate, e.g., problems with flat fitness landscapes in some regions. Three methods are commonly used to estimate uncertainty. First, the Kriging model [12,35] can estimate the mean square error (MSE) of a prediction. However, this method can only be used with the Kriging model, thereby limiting its applications. Second, differences in predictions of multiple models can estimate the degree of uncertainty [35,36], a great difference usually indicates a larger degree of uncertainty. However, multiple surrogates should have different characteristics, otherwise, the predictions may be similar and the calculated uncertainties may be be not reliable. Meanwhile, the method of adopting multiple surrogates is time-consuming. Finally, the distance from a candidate to the existing training data has been used as a measure of the uncertainty. Branke et al. [34] used Euclidean distances from a candidate point to its k nearest estimated neighborhoods to estimate the uncertainty [34]. Regis et al. [37] used the minimum distance from a candidate to previously evaluated points to estimate the uncertainty. However, the distance criterion can only be used to discover a sparsely sampled region, and the predicted uncertainty is similar if the dispersion of candidate points is similar. In addition, the uncertainty-based pre-screening criterion should not be solely used as many FEs will be spent on exploring not-well-explored areas. This may result in the delay of the convergence rate.

The third type of criterion incorporates exploration and exploitation simultaneously. For instance, in [36], a lower confidence bound criterion consisting of predicted mean values and standard deviations of a heterogeneous ensemble model was proposed to solve expensive multi-objective problems. In [38], a lower confidence bound criterion based on two different RBF models was proposed to select the most promising trail position for each particle, and the weight coefficients of the lower confidence bound criterion were changed with iterations to control the exploratory space of particles. In [39–43], some individuals with high scores of better fitness values or better lower confidence bound values are evaluated. However, the number of selected individuals in these criteria is set based on users' knowledge. There is no strict rule to set this value. Therefore, there may be not enough FEs left for exploitation if too many individuals are evaluated at each iteration.

The pre-screening criterion only considers the predicted fitness values may result in the premature convergence, while the pre-screening criterion only considers uncertainties may result in a slow convergence rate. Therefore, the two criteria cannot be solely used. The pre-screening criterion considering predicted fitness values and uncertainty simultaneously has difficulty in setting the number of exactly evaluated individuals (Ns). Many FEs are used to explore the search space if the value of Ns is too big or too small. A good solution may not be obtained if the total computational resource is limited. Therefore, in this paper, a performance-based criterion and an uncertainty-based criterion are collaboratively used in a surrogate-assisted PSO algorithm to solve medium scaled computationally expensive problems through a small number of FEs. The two criteria only select the individual with the best fitness value or the greatest uncertainty for exact evaluation to reduce the consumed FEs. The performance-based criterion is used to exploit the current global best, while the uncertainty-based criterion is used to enhance the exploration of the algorithm. The proposed algorithm can quickly exploit the current promising area by using the performance-based criterion, and the probability of falling into local optimum is reduced by using the uncertainty-based criterion. It can achieve a better solution in a limited number of FEs as no more than three candidates are evaluated at each iteration. Therefore, we name the proposed algorithm as a fast surrogate-assisted PSO (FSAPSO) algorithm. In addition, the uncertainty criterion based on Kriging model or multiple models either has a requirement for the model or needs more than one model. The distance-based uncertainty criterion does not consider the fitness landscape of different problems. Considering the defects of the three criteria, we developed a criterion to estimate uncertainty by considering the distance and fitness value information simultaneously. This criterion can make up for the disadvantage of the conventional distance-based uncertainty criterion by considering the fitness landscape of a problem. In addition, it can be used for any SAEA irrespective of the used surrogate model.

The main contribution of the paper can be summarized as follows.

(1) Different from conventional SAEAs using one pre-screening criterion to balance exploration and exploitation, two criteria are collaboratively used in the proposed FSAPSO algorithm to select individuals for exact FEs. Its main advantage is that the performance-based criterion can promote the exploitation and accelerate convergence rate, while the uncertainty-based criterion can promote exploration and relieve the premature stagnation. The proposed algorithm can balance the exploration and exploitation well by evaluating several individuals at each criterion, and it can obtain a good solution through a small number of FEs.

(2) Different from the conventional method to estimate uncertainty, an uncertainty-based criterion considering the distance and fitness value information simultaneously is proposed. The main advantage of this criterion is that it makes up for the disadvantage of the conventional distance-based uncertainty criterion by considering the fitness landscape of a problem. In addition, it can be used for any SAEAs irrespective of the used surrogate model.

The rest of this paper is organized as follows. Section 2 provides the preliminaries of the work presented in this paper. Section 3 describes the main components of the proposed fast surrogate-assisted PSO (FSAPSO) algorithm and the proposed uncertainty-based criterion. Section 4 presents a behavior study of the proposed algorithm and comparisons between the proposed algorithm and other state-of-the-art SAEAs on some numerical instances and an engineering instance. Section 5 concludes the paper and discusses some future work.

2. Preliminaries

2.1. RBF Models

The RBF model was originally developed for scattered multivariate data interpolation [8]. It uses a weighted sum of basis functions to approximate complicated landscapes [44]. For a data set consisting of the values of the input variables and response values at Nt training points, the true function $y(x)$ can be approximated as

$$\hat{y}(x) = \sum_{i=1}^{Nt} \lambda_i \varphi(\|x - c_i\|) + p(x) \tag{1}$$

where λ are coefficients calculated by solving linear equations; c_i denotes the ith center of basis functions; p is either a polynomial model or a constant value, and a linear polynomial is used in the paper [23,37]; φ is a basis function.

As (1) is underdetermined, the orthogonality condition is further imposed on coefficients λ as

$$\sum_{i=1}^{Nt} \lambda_i p_j(x_i) = 0, \qquad \text{for } j = 1, 2, \ldots, m \tag{2}$$

$$\begin{bmatrix} \Phi & P \\ P^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ b \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix} \tag{3}$$

where m is the number of terms of $p(x)$, $\Phi_{ij} = \varphi(\|x_i - x_j\|), (i = 1, 2, \ldots, Nt), (j = 1, 2, \ldots, Nt), P_{ij} = p_j(x_i), (i = 1, 2, \ldots, Nt), (j = 1, 2, \ldots, m), \lambda = [\lambda_1, \lambda_2, \ldots, \lambda_{Nt}]^T$, and $b = [b_1, b_2, \ldots, b_m]^T$. Eq. (3) consists of $(Nt + m)$ equations and its solution gives coefficients λ and b in (1).

2.2. Standard PSO algorithm

In the standard PSO (SPSO) [45], the particle updates its velocity and position according to (4) and (5). We used the two formulas to update the particle swarm in this paper. Hereinafter, we also use the standard SPSO to denote the PSO with the inertia weight.

$$v_i(t + 1) = w \cdot v_i(t) + c_1 \cdot rand \cdot (p_i(t) - x_i(t))$$
$$+ c_2 \cdot rand \cdot (p_g(t) - x_i(t)) \tag{4}$$

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \tag{5}$$

where w is inertia factor; v_i is the speed of the ith particle; c_1 and c_2 are learning factors; $rand$ is random number in [0, 1]; x_i is the current location of the particle; p_i is the best position of the ith particle; and p_g is the global optimum position of the particle swarm.

3. The proposed FSAPSO algorithm

The particle swarm often needs to experience a process of exploration and exploitation before converging to a good solution. A large number of FEs will be consumed if too many particles are exactly evaluated at each iteration. Therefore, it is important to select the "right" particles for exact evaluations for computationally expensive problems.

Two criteria, the performance-based criterion and the uncertainty-based criterion, are used in tandem to select candidate solutions for exact FEs in the proposed FSAPSO. In the performance-based criterion, the particle with the best predicted fitness value and the optimum of the surrogate model are exactly evaluated at each iteration to control the searching directions of the particle swarm. The optimum of the surrogate model will replace the current global best of the swarm if the optimum is better. If only the performance-based criterion is used, particles may concentrate in the neighborhood of the optimum of the model after a few iterations. Hence, the swarm may quickly converge to a local optimum. Therefore, a novel uncertainty-based criterion is used in the paper to select the particle with the greatest uncertainty for exact evaluation to enhance the exploration of the algorithm. The accuracy of the model is also improved as particles located in sparse regions usually have large uncertainties. Generally, the particle with great uncertainty can only make a small improvement to the current global best of the swarm. If the particle with the greatest uncertainty is exactly evaluated at every iteration, some FEs may be wasted. Referring to the method in [35], the particle with the greatest uncertainty is exactly evaluated only when the current global best of the swarm is not improved.

The proposed FSAPSO algorithm is different from a surrogate-assisted PSO algorithm with the help of committee-based active learning (CAL-SAPSO) [35] although both the performance-based and uncertainty-based criteria are used in the two algorithms. First, the FSAPSO algorithm uses two criteria to select particles from the swarm for exact FEs, while CAL-SAPSO mainly works to find optima of models corresponding to the two criteria. Therefore, CAL-SAPSO seems like a global optimization algorithm based on surrogate models, while FSAPSO is a surrogate-assisted PSO algorithm. Second, CAL-SAPSO uses differences in predictions of multiple surrogate models to estimate uncertainty, while FSAPSO uses distance and fitness value information to estimate uncertainty.

3.1. Overall flowchart of FSAPSO

A generic diagram of the proposed FSAPSO algorithm is presented in Fig. 1. The detailed flow of the proposed algorithm is shown in Algorithm. 1. At the beginning of the process, Latin Hypercube Sampling is used to generate the initial samples, and N samples with the best fitness values are taken as the initial population. Then, an RBF model is built with all the samples, and the optimum of the RBF model ($xmin_{RBF}$) is exactly evaluated if the minimum distance between $xmin_{RBF}$ and other evaluated samples is larger than the threshold η. The threshold η is used to avoid the samples being too closed. The $xmin_{RBF}$ will replace the global best of the swarm if it is better. Then, velocities and positions of the particles are updated. Finesse values of all particles are first predicted by the RBF model, then the particle with the minimum prediction is exactly evaluated if the minimum distance between the particle and other samples is bigger than η. The particle with the greatest uncertainty is also exactly evaluated if the current global best is not improved. Note that three kinds of solutions will be stored in the database (DB): initial sample points, the optimum of the RBF model and the exactly evaluated particles. The last two kinds of solutions will be cumulatively stored in the DB at each iteration.
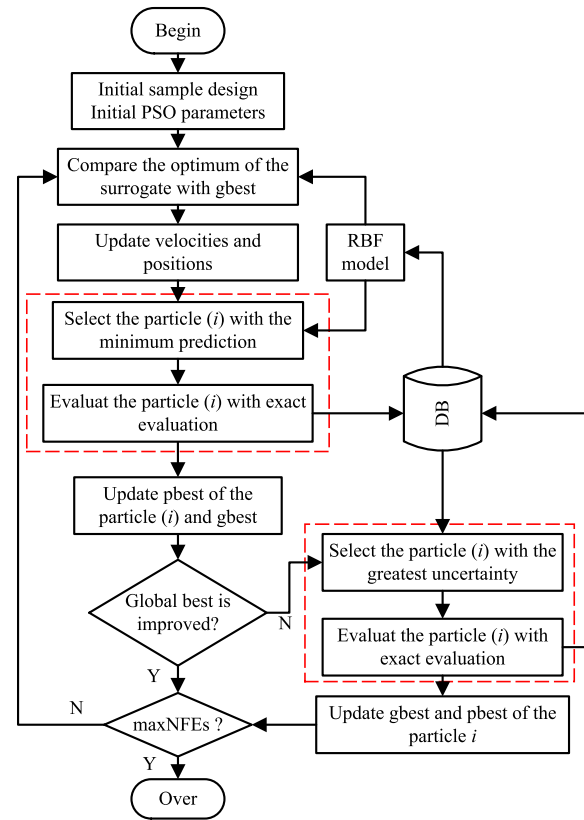

Fig. 1. Flowchart of the FSAPSO algorithm.

3.2. Model management

A global RBF model with cubic basis functions is constructed to predict the fitness values of all particles as cubic basis functions have shown good performance on other SAEAs [23,37]. The optimum of the RBF model is exactly evaluated to update the global best of the swarm if the optimum is better. The RBF model has the ability to approximate high-order nonlinear problems, so the approximation model may have many local optima. Although evaluating all the local optima is beneficial for the exploration, many FEs may be consumed. The optimum of the surrogate is mainly used to improve the current global best, so a local optimization algorithm is used to find the optimum. In this paper, the fimincon solver with the interior point method in Matlab R2015a is used to find the optimum of the model. The starting point for the local optimization algorithm is the current global best of the swarm. The searching space for finding the optimum is restricted to space where the current population is located as the optimum in this area may have a bigger probability to improve the current optimum.

3.3. The proposed uncertainty-based criterion

We consider the following three situations. First, if the dispersion of samples is similar, areas with complex fitness landscapes usually have greater uncertainties for they are more difficult to approximate. Second, if the complexity of the landscapes is similar across space, candidates near the evaluated points usually have smaller uncertainties as the nearest sample points have the greatest influence on it. Finally, similar to the second situation, the uncertainty of a candidate usually becomes smaller with more evaluated points existing in its neighbor. Overall, the uncertainty

---

Algorithm 1. The Pseudo Code of the FSAPSO algorithm

Input: Dimensions of parameter space (D); expensive functions (f) ;
the number of initial sample points (K); the number of maximum FEs ($maxNFE$); swarm size (N);
Thresholds for determining whether the sample is evaluated by f (η);
Output: The solution with the best fitness value: $gbest$ and f_{gbest}

1:   **Initial sample design:** the initial samples are generated by Latin Hypercube Sampling; evaluate
     the initial samples with the real function; save the initial samples in the DB; $NFEs=K$
2:   **Initial population design**: Select N points with the best fitness values from the initial samples;
     determine the initial velocities of the particles v;
     determine the personal best of each particle;
     determine the global best of the swarm;
3:   **While** $NFEs<maxNFE$
4:     Use all the samples to establish an RBF model
5:     Find the minimum of the surrogate: $xmin_{RBF}$;
6:     Calculate the minimum distance between $xmin_{RBF}$ and other samples in the DB: dx_{RBF}
7:     **if** $dx_{RBF} > \eta$
8:      Evaluate $xmin_{RBF}$ and add it to the DB; $NFEs=NFEs+1$
9:      Update the global best of the swarm
10: **end if**
11: **for** j=1: N **do**
12:    Update velocities and positions of the particles by using (1) and (2)
13: **end for**
14: Predict the fitness values of the particles by using the RBF model: pre
15: Find the particle with the minimum value of pre, and name it as the particle i
16: Calculate the minimum distance between the particle i and other samples in the DB: dx
17: **if** $dx > \eta$
18:    Evaluate the particle i with the exact function: $fpop_i=f(pop_i)$
19:    Add the new sample to the DB; $NFEs=NFEs+1$
20:    Update the personal best position of the particle i
21:    Update the global best of the swarm
22: **end if**
23: **if** the global best is not improved
24:    Find the particle with the greatest uncertainty, and name it as particle i
25:    Evaluate the particle i with the exact function: $fpop_i=f(pop_i)$
26:    Add the new sample to the DB; $NFEs=NFEs+1$
27:    Update the personal best position of the particle i
28:    Update the global best of the swarm
29: **end if**
30: **End while**

---

is influenced by the fitness landscape of a problem, the distance from a candidate to its nearest evaluated points and the number of its nearest evaluated points. The conventional distance criterion used in other works only considers distances between the candidate and its evaluated neighboring points, so the predicted uncertainty may be unreliable.

Considering the defect of the distance criterion, we improved it by adding the information of fitness values and named it as the distance and fitness value (DF) criterion. In a small region, the distribution of fitness values can indicate the smoothness of the fitness landscape to some degree. The fitness landscape with a small variance of fitness values is likely to be slightly flatter than the fitness landscape with a bigger one. However, this assumption only holds in a small local area for a rugged landscape may have a small variance of fitness values in a big search space. The improved uncertainty-based criterion considers distances from a candidate to its k nearest evaluated points and the fitness values variance of its k nearest evaluated points. The formula of calculating the DF value of a candidate is shown in (6), where s is a modified sigmoid function. s is used to express the relation between the distance from a candidate to its nearest evaluated point and the uncertainty. The uncertainty of a candidate becomes smaller when the distance from it to its nearest neighbor becomes smaller. The uncertainty is zero on the exactly evaluated point as shown in Fig. 2. The function value of the sigmoid function is close to a stable value when the variable is

close to five. It is hard to distinguish the effect of the distance on uncertainty if most of the nearest distances are larger than five, so the nearest distances of different particles are scaled to (0, 5) by using (12). The problem caused by the different magnitude of distances is also avoided. The mean value of distances from a candidate to its k nearest neighbors and the fitness values variance of its k nearest neighbors are taken as a weight of s as both of them have influence on the uncertainty. To solve the scaling problem, the two values are divided by their respective sums as shown in (9) and (10).

$$u_i = s_i(dm_{\text{norm},i} + \sigma_{\text{norm},i}) \tag{6}$$

$$dm_i = \frac{1}{k}\sum_{j=1}^{k} d_{ij} \tag{7}$$

$$\sigma_i = \sqrt{\frac{1}{k}\sum_{j=1}^{k}(y_j - \bar{y})^2}, \bar{y} = \frac{1}{k}\sum_{j=1}^{k} y_j \tag{8}$$

$$dm_{\text{norm},i} = dm_i / \sum_{i=1}^{N} dm_i \tag{9}$$

$$\sigma_{\text{norm},i} = \sigma_i / \sum_{i=1}^{N} \sigma_i \tag{10}$$
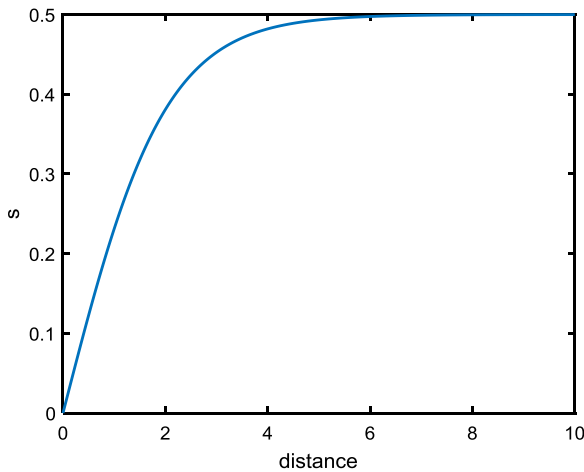
$$s_i = 1/(1 + e^{-d'_{i1}}) - 0.5 \tag{11}$$

Fig. 2. The relation between the nearest distance and the uncertainty.

$$d'_{i1} = 5d_{i1} / \sum_{i=1}^{N} d_{i1} \qquad (12)$$

where k is the number of nearest neighbors, d_{ij} is the distance from the candidate i to its jth nearest neighbor, d_{i1} is the distance from the candidate i to its nearest neighbor, y_j is the fitness value of the jth nearest neighbor. Fig. 3(a) shows the Kriging model of the one-dimension Forrester function with four sample points distributed uniformly in [0,1], and Fig. 3(b) shows the minimum distance (dmin) from candidates to sample points, DF value of the candidates, and the MSE as predicted by Kriging. It is evident that the dmin criterion is only related to the distribution of sample points but the DF criterion can reflect the complexity of the landscape.

To test the effectiveness of the proposed DF criterion, the one-dimension Forrester function with four non-uniformly distributed sample points is used to compare uncertainties predicted by the three criteria. The variation trends of the three criteria are similar without considering peak values as shown in Fig. 4. However, peak values of the dmin criterion and DF criterion are not identical as in the MSE of the Kriging model. The dmin criterion only considers the distance to the nearest neighbor, so peak values are only related to the distance. The DF criterion also considers fitness values, so the area with a larger variance of fitness values tends to have bigger uncertainties with a similar distribution of sample points.

In the high-dimensional searching space, samples are sparsely distributed in the search space at the initial stage of the algorithm. Distances between particles to its nearest samples are relatively big. For instance, Fig. 5 shows the distances of particles to each evaluated point from the DB in different iterations by using FSAPSO to test a 50-dimension Rastrigin function. If the nearest distances are large, the variance of fitness values may not indicate the real fitness landscape of the area where the candidate is located. In this situation, the particles may all have large uncertainties as the surrogate model is clearly not accurate anywhere, so the DF criterion can also be used to select the candidate with the largest uncertainty. More evaluated points are located in the area where the swarm exists with the iteration going on. Distances between particles to its nearest samples become smaller gradually. The variance of fitness values is reliable, and the DF criterion has the ability to identify the most uncertain candidate.

4. Experimental study

4.1. Parameter settings

During experiments, 30 independent runs are performed for each algorithm. Eight benchmark functions in Table 1 with different characteristics are taken to evaluate the effectiveness of the proposed algorithm [26,27,29,31,35]. For all the used algorithms in this paper, the termination condition is that the number of consumed FEs is less than *maxNFE*, and *maxNFE* is set as 11*D* [31,35]. For all the used PSO algorithms in this paper, w= 0.792-(0.792-0.2)$NFE/maxNFE$, $c_1 = c_2 = 1.491$ as they are commonly used in the literature [3,46]. The number of initial sample points K is max(D,20) as the search space increases with the increase of variables of the problems. D is the dimension of the problem. The swarm size N is 20. $Vmax$=0.1(ub-lb), $Vmin$=-Vmax. ub is the upper boundary of variables, and lb is the lower boundary. For FS-APSO algorithm, η is min(sqrt(0.001^2D), 5.0×10^{-5}D×min(ub-lb)) inspired by the method in [23]. The number of nearest neighbors (k) is 3, and the effects of it on the proposed algorithm is analyzed on Section 4.4.

4.2. Behavior study of FSAPSO

4.2.1. Effects of the performance-based criterion

To show the effects of the performance-based criterion on the proposed FSAPSO algorithm, we compare FSAPSO with its two variants (FSAPSO-WOMO and FSAPSO-WOBP). FSAPSO-WOMO is similar to FSAPSO except that it does not evaluate the optimum of the model. FSAPSO-WOBP is similar to FSAPSO except that it does not evaluate the best particle. Five benchmark problems with dimensions D = 10, 20, 30 are used to test the three algorithms. The average best objective function values obtained by the three algorithms over 30 independent runs are shown in Table 2, and the best mean values of individual instances are highlighted. The last column of the tables lists the results of the Wilcoxon rank-sum test calculated at a significance level of α =0.05, where '≈' indicates that there is no statistically significant difference between the results obtained by FSAPSO algorithm and the compared algorithms, '+' indicates that FSAPSO algorithm is significantly better than the compared algorithms, while '−' means that FSAPSO algorithm is significantly outperformed by the compared algorithms. FSAPSO algorithm significantly outperforms FSAPSO-WOMO algorithm on all problems as shown in Table 2. However, FSAPSO and FSAPSO-WOBP algorithms show similar performances on thirteen out of fifteen problems, and FSAPSO-WOBP algorithm slightly outperforms FSAPSO algorithm on the 30-dimension Ackley function, while FSAPSO algorithm is slightly better than it on the 20-dimension Rosenbrock function. This indicates that evaluating the optimum of the model can effectively improve the quality of the final solutions, and evaluating the best particle in the swarm improves solutions of some problems. The convergence curves of the three algorithms on 10-dimension Rastrigin is shown in Fig. 6. FSAPSO achieves significantly better solutions than its two variances after about 60 FEs. This means that evaluating the best particle and the optimum of the surrogate model can make a good guidance for PSO and promote exploitation. In addition, FSAPSO-WOBP obtains better results than FSAPSO-WOMO on most of the problems. This indicates the optimum of the surrogate model can produce more promising information for the swarm. Overall, the effectiveness of the performance-based criterion on FSAPSO algorithm is demonstrated.
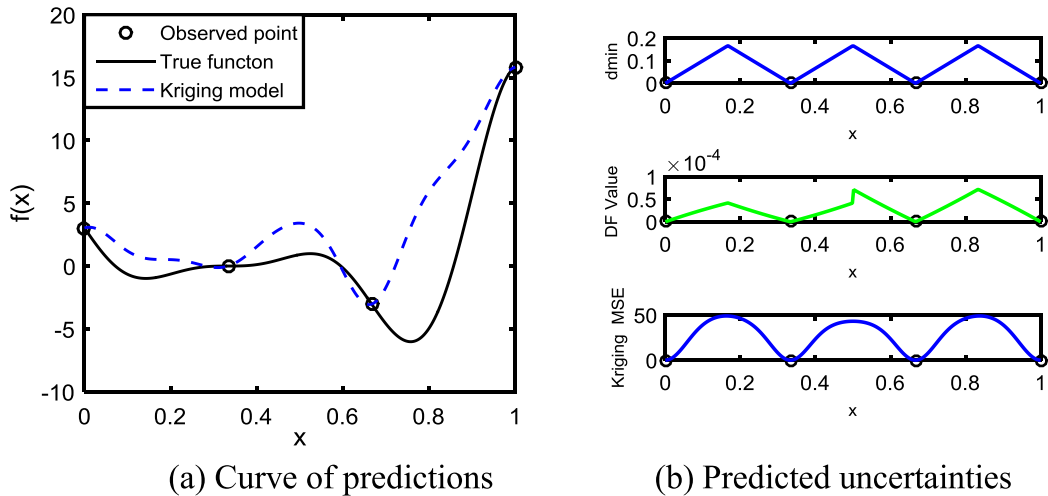
(a) Curve of predictions   (b) Predicted uncertainties

Fig. 3. Forrester function with four uniformly distributed sample points.

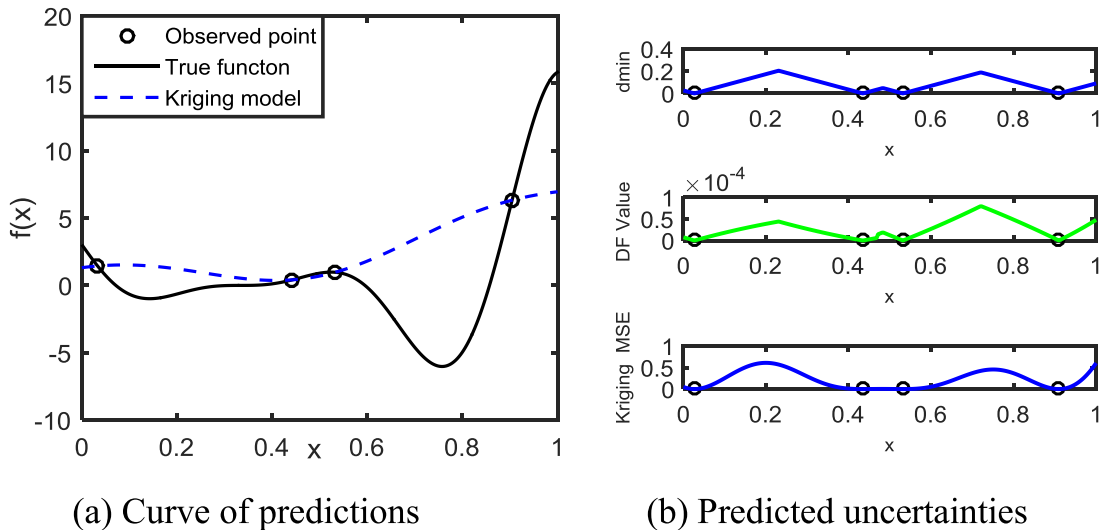(a) Curve of predictions   (b) Predicted uncertainties

Fig. 4. Forrester function with four non-uniformly distributed sample points.

Table 1
Benchmark functions used in the experimental study.

Benchmark Problem	Dimension	Characteristics	Global Optimum
Ellipsoid	10, 20, 30, 50	Unimodal	0
Rosenbrock	10, 20, 30, 50	Multimodal with narrow valley	0
Ackley	10, 20, 30, 50	Multimodal	0
Griewank	10, 20, 30, 50	Multimodal	0
Rastrigin	10, 20, 30, 50	Multimodal	0
F10 in CEC 2005 [47]	50	Shifted Rotated Rastrigin Very complicated multimodal	−330
F16 in CEC 2005 [47]	50	Rotated Hybrid composition function Very complicated multimodal	120
F19 in CEC 2005 [47]	50	Rotated Hybrid composition function Very complicated multimodal	10

4.2.2. Effects of the uncertainty-based criterion

To show the effects of the uncertainty-based criterion on the FSAPSO algorithm, we compare FSAPSO and FSAPSO without the uncertainty-based criterion (denoted as FSAPSO-WOU). The average best objective function values obtained by the three algorithms over 30 independent runs are shown in Table 3, and the best mean results of individual instances are highlighted. FSAPSO algorithm outperforms FSAPSO-WOU algorithm on seven out of fifteen problems and they show similar performance on eight problems. The two algorithms show similar performance on the 10-dimension Ellipsoid and two simple multi-modal functions (Ackley and Griewank). This may be that these problems are easy to approximate and the surrogate model can provide real promising information for the swarm even without the uncertainty-based criterion. However, the sample points are easy to gather in the neighbor of the current optimum without the uncertainty-based criterion, so the FSAPSO-WOU algorithm is easy to be trapped in local optima. The algorithm can explore more areas and the accuracy of the model is also improved with the uncertainty-based criterion. Besides, standard deviations achieved by FSAPSO algorithm are smaller than the algorithm
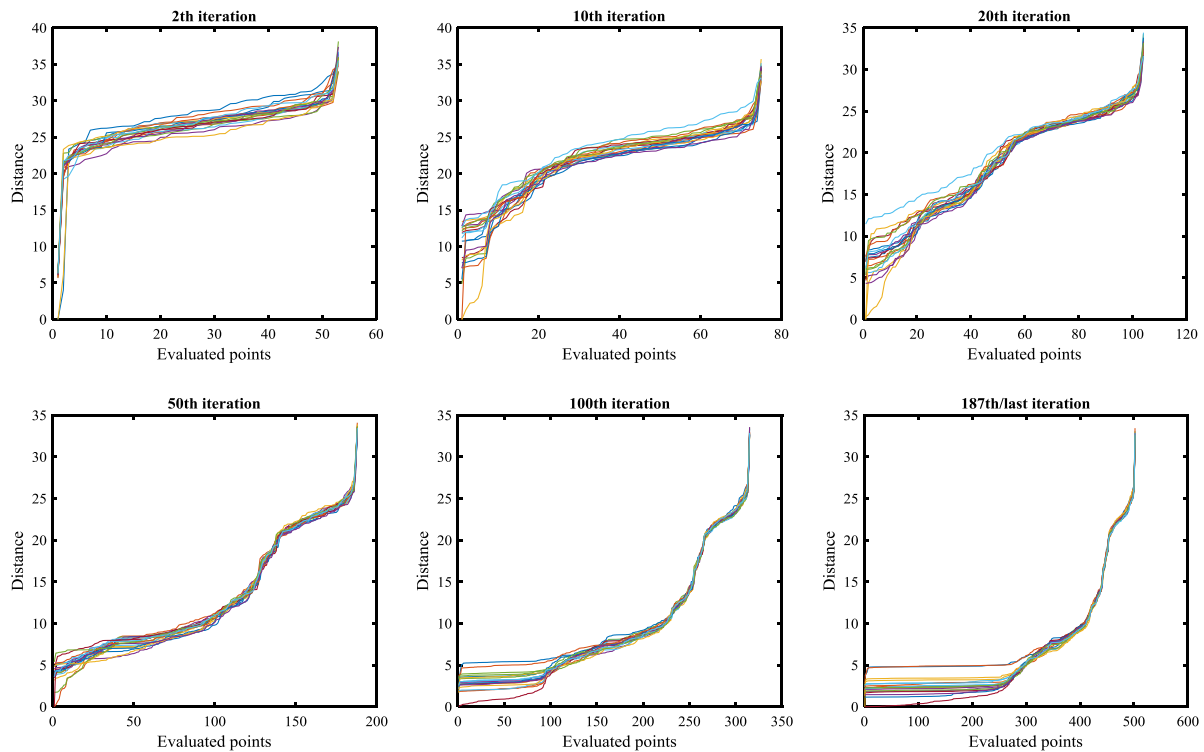
Fig. 5. Distances of particles to the evaluated points at different iterations.
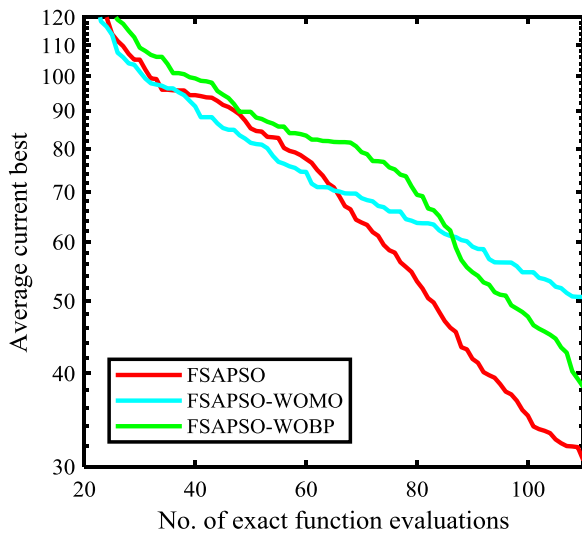

Fig. 6. Convergence profiles of the three algorithms on 10-dimension Rastrigin function.

without the uncertainty-based criterion on most problems. The results indicate that the algorithm with more exploration is more stable. Therefore, the effectiveness of the uncertainty-based criterion on FSAPSO algorithm is demonstrated.

4.2.3. Compare the DF criterion with other uncertainty-based criteria

Although the effectiveness of the DF criterion on the FSAPSO algorithm has been demonstrated, whether the DF criterion is a good measure of the uncertainty is unknown. Therefore, the DF criterion is also compared with other uncertainty-based criteria to further investigate its characteristics. Two variants of the FSAPSO algorithm, FSAPSO using MSE of Kriging model to replace the DF criterion and FSAPSO using a random criterion (randomly selecting a particle) to replace the DF criterion, are compared with FSAPSO algorithm with the DF criterion. The MSE of the Kriging model is a commonly used method for uncertainty. The random criterion is used to test whether it can play the same role as the DF criterion in FSAPSO. The outcome of this comparison will show whether other criteria are as effective as the DF criterion in FSAPSO. The average best objective function values obtained by the three algorithms over 30 independent runs are shown in Table 4, and the best mean values of individual instances are highlighted. DF criterion outperforms the random criterion on five out of fifteen problems, while their performances are very close on nine problems. Besides, the DF criterion leads to better mean values on twelve problems, so the DF criterion is different from the random criterion. These results also indicate that not all the criteria can show a similar performance as the DF criterion in FSAPSO. Furthermore, the DF criterion leads to better performance than the MSE criterion on ten out of fifteen problems. This indicates that the effectiveness of the DF criterion.

4.3. Comparative experiments on benchmark problems

4.3.1. Experiments on 10-, 20-, and 30-dimension benchmark problems

To further examine the performance of the proposed algorithm, we compare it with five state-of-art algorithms. SPSO algorithm [45] is adopted to test the effects of the surrogate model. Parameters of the SPSO are set as the same with FS-APSO. SAPSO-PBEST, an RBF model-assisted PSO algorithm, is used to investigate the effects of different pre-screening criteria on the algorithm. SAPSO-PBEST is similar to FSAPSO except that the PBEST-based criterion is used to select particles for exact evaluations. Three recently proposed surrogate-based optimization algorithms are also used. SHPSO [26] is a surrogate-assisted hierarchical particle swarm optimizer consisting of a PSO algorithm and a social learning-based PSO (SL-PSO) algorithm to

Table 2
Statistics results of FSAPSO, FSAPSO-WOMO and FSAPSO-WOBP algorithms.

Problem	D	Algorithm	Best	Worst	Median	Mean	Std	t-test
Ellipsoid	10	FSAPSO	6.94E−04	3.90E−01	1.75E−02	4.27E−02	8.48E−02	
		FSAPSO-WOMO	9.51E−01	1.74E+01	4.72E+00	6.80E+00	4.94E+00	+
		FSAPSO-WOBP	6.36E−04	1.28E−01	1.71E−02	**3.39E−02**	3.82E−02	≈
Ellipsoid	20	FSAPSO	2.30E−02	1.47E+00	3.48E−01	4.47E−01	3.82E−01	
		FSAPSO-WOMO	3.11E+00	3.32E+01	1.21E+01	1.42E+01	8.32E+00	+
		FSAPSO-WOBP	2.07E−02	1.25E+00	1.39E−01	**3.25E−01**	3.78E−01	≈
Ellipsoid	30	FSAPSO	1.24E−01	2.43E+00	1.06E+00	1.10E+00	6.79E−01	
		FSAPSO-WOMO	1.08E+01	5.19E+01	2.81E+01	2.95E+01	1.23E+01	+
		FSAPSO-WOBP	1.07E−01	1.54E+00	5.56E−01	**6.32E−01**	3.99E−01	−
Rosenbrock	10	FSAPSO	7.88E+00	4.06E+01	9.79E+00	**1.17E+01**	7.03E+00	
		FSAPSO-WOMO	1.07E+01	3.47E+01	1.58E+01	1.75E+01	6.38E+00	+
		FSAPSO-WOBP	8.15E+00	3.85E+01	1.06E+01	1.30E+01	6.69E+00	+
Rosenbrock	20	FSAPSO	1.50E+01	6.07E+01	1.99E+01	2.32E+01	1.17E+01	
		FSAPSO-WOMO	2.34E+01	6.31E+01	3.76E+01	3.85E+01	9.58E+00	+
		FSAPSO-WOBP	1.87E+01	3.25E+01	2.01E+01	**2.14E+01**	3.17E+00	≈
Rosenbrock	30	FSAPSO	2.92E+01	6.87E+01	3.85E+01	4.32E+01	1.22E+01	
		FSAPSO-WOMO	4.90E+01	1.56E+02	9.14E+01	9.38E+01	3.15E+01	+
		FSAPSO-WOBP	3.03E+01	5.42E+01	3.51E+01	**3.69E+01**	5.89E+00	≈
Ackley	10	FSAPSO	2.63E+00	1.15E+01	3.72E+00	4.48E+00	2.06E+00	
		FSAPSO-WOMO	5.92E+00	1.34E+01	9.60E+00	9.89E+00	2.20E+00	+
		FSAPSO-WOBP	2.05E+00	8.68E+00	3.63E+00	**4.41E+00**	2.19E+00	≈
Ackley	20	FSAPSO	2.90E+00	8.31E+00	6.05E+00	5.69E+00	1.74E+00	
		FSAPSO-WOMO	6.62E+00	1.21E+01	8.91E+00	9.07E+00	1.48E+00	+
		FSAPSO-WOBP	2.34E+00	7.10E+00	5.79E+00	**5.30E+00**	1.57E+00	≈
Ackley	30	FSAPSO	3.27E+00	1.32E+01	5.75E+00	6.21E+00	2.36E+00	
		FSAPSO-WOMO	7.65E+00	1.17E+01	9.98E+00	9.83E+00	1.38E+00	+
		FSAPSO-WOBP	3.47E+00	6.50E+00	5.05E+00	**5.00E+00**	7.92E−01	≈
Griewank	10	FSAPSO	3.35E−01	1.01E+00	8.46E−01	**7.84E−01**	2.11E−01	
		FSAPSO-WOMO	1.90E+00	7.12E+00	3.68E+00	4.04E+00	1.36E+00	+
		FSAPSO-WOBP	5.80E−01	1.20E+00	9.16E−01	8.81E−01	1.85E−01	≈
Griewank	20	FSAPSO	1.89E−01	6.89E−01	4.11E−01	**4.16E−01**	1.56E−01	
		FSAPSO-WOMO	3.81E+00	1.62E+01	7.12E+00	7.68E+00	3.20E+00	+
		FSAPSO-WOBP	2.13E−01	7.26E−01	5.23E−01	4.99E−01	1.54E−01	≈
Griewank	30	FSAPSO	1.79E−01	5.18E−01	3.20E−01	**3.33E−01**	8.56E−02	
		FSAPSO-WOMO	4.83E+00	2.88E+01	9.12E+00	9.95E+00	4.95E+00	+
		FSAPSO-WOBP	2.22E−01	5.58E−01	3.78E−01	3.93E−01	9.37E−02	≈
Rastrigin	10	FSAPSO	1.29E+01	5.54E+01	2.72E+01	**3.06E+01**	1.18E+01	
		FSAPSO-WOMO	3.10E+01	8.17E+01	5.19E+01	5.06E+01	1.18E+01	+
		FSAPSO-WOBP	1.20E+01	6.80E+01	3.49E+01	3.84E+01	1.70E+01	≈
Rastrigin	20	FSAPSO	3.08E+01	8.96E+01	4.85E+01	5.26E+01	1.55E+01	
		FSAPSO-WOMO	5.84E+01	1.32E+02	9.94E+01	9.30E+01	2.00E+01	+
		FSAPSO-WOBP	1.79E+01	8.19E+01	4.13E+01	**4.46E+01**	1.78E+01	≈
Rastrigin	30	FSAPSO	2.49E+01	9.95E+01	6.62E+01	6.80E+01	2.07E+01	
		FSAPSO-WOMO	9.79E+01	1.95E+02	1.29E+02	1.35E+02	2.86E+01	+
		FSAPSO-WOBP	3.91E+01	1.15E+02	5.54E+01	**6.15E+01**	2.07E+01	≈

explore and exploit search spaces. SHPSO is similar to SAPSO-PBEST except that the RBF model is built with the first P non-duplicated best samples in the archive, and an SL-PSO algorithm is used to search the optimum of the model. CAL-SAPSO [35], an ensemble-surrogate-based global optimization method, employs a PSO algorithm to find the best and the most uncertain points for exact evaluations. The uncertainty-based criterion of CAL-SAPSO consists of differences between outputs of the sub-models. GORS-SSLPSO [31] is a surrogate-assisted SL-PSO with a generation-based optimal restart strategy. SL-PSO restarts every few generations, and the best sample points archived in the database are employed to reinitialize the swarm at each restart. The individual with the best estimated fitness value is chosen for the exact evaluation before each restart of the SL-PSO. The parameters of the compared SAEAs are the same as those in the original papers. The average best objective function values obtained by the six algorithms over 30 independent runs are shown in Table 5 and the best mean results of individual instances are highlighted. Table 6 lists the results of Wilcoxon rank sum test calculated at a significance level of $\alpha = 0.05$, and results of CAL-SAPSO

are not tested as we cannot replicate the results of CAL-SAPSO. Convergence profiles of the algorithms are plotted in Fig. 7, and convergence profiles of CAL-SAPSO are extracted from [31].

From the results shown in Tables 5 and 6, we can observe that FSAPSO algorithm achieves the best mean values in eight out of fifteen problems, and GORS-SSLPSO algorithm obtains the best mean values on the remaining problems. In comparison with the non-surrogate assisted algorithm, FSAPSO algorithm significantly outperforms SPSO algorithms on all problems. This indicates that using the surrogates can significantly improve the final solutions. FSAPSO algorithm outperforms SAPSO-PBEST algorithm on eleven out of fifteen problems, while SAPSO-PBEST algorithm achieves slightly better results on Ackley function with $D = 20, 30$. Ackley function is a multimodal function with many shallow valleys. FSAPSO algorithm may be trapped in a local optimum. Besides, FSAPSO algorithm outperforms SHPSO algorithm on all problems. In SHSPO, the number of initial sample points is 100 when $D < 50$, and the number of samples to train the RBF surrogate model is also 100 for problems with $D < 50$, so many samples are used to construct the surrogate model. FSAPSO algorithm outperforms CAL-SAPSO algorithm on fourteen out of

Table 3
Statistics results of FSAPSO and FSAPSO-WOU algorithms.

Problem	D	Algorithm	Best	Worst	Median	Mean	Std	t-test
Ellipsoid	10	FSAPSO	6.94E−04	3.90E−01	1.75E−02	**4.27E−02**	8.48E−02	
		FSAPSO-WOU	6.42E−04	3.60E−01	2.71E−02	7.32E−02	9.63E−02	≈
Ellipsoid	20	FSAPSO	2.30E−02	1.47E+00	3.48E−01	**4.47E−01**	3.82E−01	
		FSAPSO-WOU	9.68E−02	2.10E+02	7.77E−01	1.90E+01	5.38E+01	+
Ellipsoid	30	FSAPSO	1.24E−01	2.43E+00	1.06E+00	**1.10E+00**	6.79E−01	
		FSAPSO-WOU	9.28E−01	5.66E+01	2.28E+00	9.05E+00	1.67E+01	+
Rosenbrock	10	FSAPSO	7.88E+00	4.06E+01	9.79E+00	**1.17E+01**	7.03E+00	
		FSAPSO-WOU	5.27E+00	1.27E+02	2.68E+01	3.68E+01	2.90E+01	+
Rosenbrock	20	FSAPSO	1.50E+01	6.07E+01	1.99E+01	**2.32E+01**	1.17E+01	
		FSAPSO-WOU	3.61E+01	1.12E+03	7.53E+01	1.39E+02	2.34E+02	+
Rosenbrock	30	FSAPSO	2.92E+01	6.87E+01	3.85E+01	**4.32E+01**	1.22E+01	
		FSAPSO-WOU	3.85E+01	1.45E+02	1.11E+02	1.04E+02	3.12E+01	+
Ackley	10	FSAPSO	2.63E+00	1.15E+01	3.72E+00	**4.48E+00**	2.06E+00	
		FSAPSO-WOU	7.06E−01	1.78E+01	5.01E+00	7.13E+00	5.32E+00	≈
Ackley	20	FSAPSO	2.90E+00	8.31E+00	6.05E+00	**5.69E+00**	1.74E+00	
		FSAPSO-WOU	2.39E+00	1.65E+01	5.57E+00	7.82E+00	4.98E+00	≈
Ackley	30	FSAPSO	3.27E+00	1.32E+01	5.75E+00	**6.21E+00**	2.36E+00	
		FSAPSO-WOU	3.38E+00	1.62E+01	7.16E+00	8.17E+00	3.82E+00	≈
Griewank	10	FSAPSO	3.35E−01	1.01E+00	8.46E−01	7.84E−01	2.11E−01	
		FSAPSO-WOU	3.80E−01	9.80E−01	7.29E−01	**7.17E−01**	1.57E−01	≈
Griewank	20	FSAPSO	1.89E−01	6.89E−01	4.11E−01	4.16E−01	1.56E−01	
		FSAPSO-WOU	1.93E−01	8.10E−01	3.07E−01	**3.66E−01**	1.81E−01	≈
Griewank	30	FSAPSO	1.79E−01	5.18E−01	3.20E−01	**3.33E−01**	8.56E−02	
		FSAPSO-WOU	8.89E−02	9.02E+01	2.84E−01	4.78E+00	2.01E+01	≈
Rastrigin	10	FSAPSO	1.29E+01	5.54E+01	2.72E+01	**3.06E+01**	1.18E+01	
		FSAPSO-WOU	9.96E+00	7.10E+01	2.94E+01	3.41E+01	1.72E+01	≈
Rastrigin	20	FSAPSO	3.08E+01	8.96E+01	4.85E+01	**5.26E+01**	1.55E+01	
		FSAPSO-WOU	3.59E+01	1.66E+02	7.26E+01	7.58E+01	3.20E+01	+
Rastrigin	30	FSAPSO	2.49E+01	9.95E+01	6.62E+01	**6.80E+01**	2.07E+01	
		FSAPSO-WOU	4.88E+01	1.43E+02	9.20E+01	9.51E+01	2.75E+01	+

fifteen problems, and they show comparable performance on Rosenbrock with $D = 30$. However, FSAPSO algorithm only outperforms GORS-SSLPSO algorithm on Rosenbrock function with $D = 10, 20, 30$, and GORS-SSLPSO algorithm achieves slightly better results on 30-dimension Ellipsoid function and 10-dimension Ackley function.

From the performance curves presented in Fig. 7, one can easily see that FSAPSO and GORS-SSLPSO algorithms converge faster than the other four algorithms on most of the problems. SPSO, CAL-SAPSO, and SHPSO converge slower than other algorithms on most of the problems. FSAPSO can achieve greater improvement than SPSO on current optimum with the same number of FEs. Moreover, the global optima obtained by SAPSO-PBEST algorithm are worse than those obtained by FSAPSO algorithm within the same number of FEs on most of the problems except Ackley function with $D = 20$ and 30. FSAPSO converges much slower than SAPSO-PBEST in the later stage for Ackley function with $D = 20$ and 30. However, FSAPSO converges much faster than SAPSO-PBEST on the other two multi-modal functions (Rastrigin and Griewank). The three functions are multi-modal functions with many local optima, but Ackley has much shallower local optima than the other two problems. The PBEST-based pre-screening criterion can find promising areas on Ackley function, while many FEs are consumed on exploration for the other two problems. The performance-based criterion and uncertainty-based criterion are used in tandem in FSAPSO. This can achieve a good balance of exploration and exploitation, and will not waste many unnecessary FEs on exploration. In addition, FSAPSO converges much faster than the other surrogate-assisted algorithm (CAL-SAPSO and SHPSO). This indicates the superiority of the two criteria in FSAPSO.

From the above results, we can make the following observations. First, surrogate models help improve the performance of PSO. Second, the collaboration between the performance-based criterion and uncertainty-based criterion can make FSAPSO achieve good performance on both uni-modal and multi-modal problems. FSAPSO can obtain good solutions for the five classical benchmark functions in the context of the limited computational budget scenario. In addition, the convergence rate of FSAPSO is much faster than the other algorithms on most problems. This may owe to the good collaboration between performance-based criterion and uncertainty-based criterion. The performance-based criterion can promote the exploitation and accelerate convergence rate, while the uncertainty-based criterion can enhance exploration and avoid unnecessary FEs.

4.3.2. Experiments on 50-dimension benchmark problems

To further test the performance of the FSAPSO algorithm on higher-dimensional problems, 50-dimension functions in Table 1 are used in the experimental study. SPSO, SAPSO-PBEST, SH-PSO [26], GORS-SSLPSO [31], a surrogate-assisted cooperative swarm optimization algorithm (SA-COSO) [27] and an evolutionary sampling assisted optimization (ESAO) algorithm [29], are also used for comparison. The average best objective function values obtained by the algorithms over 30 independent runs are shown in Table 7, and the best mean results of individual instances are highlighted. Table 8 lists the results of the Wilcoxon rank sum test calculated at a significance level of $\alpha = 0.05$. The results of ESAO [29] are copied from the original papers. Rastrigin and F16 are not tested in the original papers of ESAO, so the results of the two functions on the three algorithm are blank. Convergence profiles of the algorithms are plotted in Fig. 8. Convergence profiles of ESAO algorithms are copied from the original papers. There may be some deviations to the original data, but the overall trends can be reflected for a fair comparison.
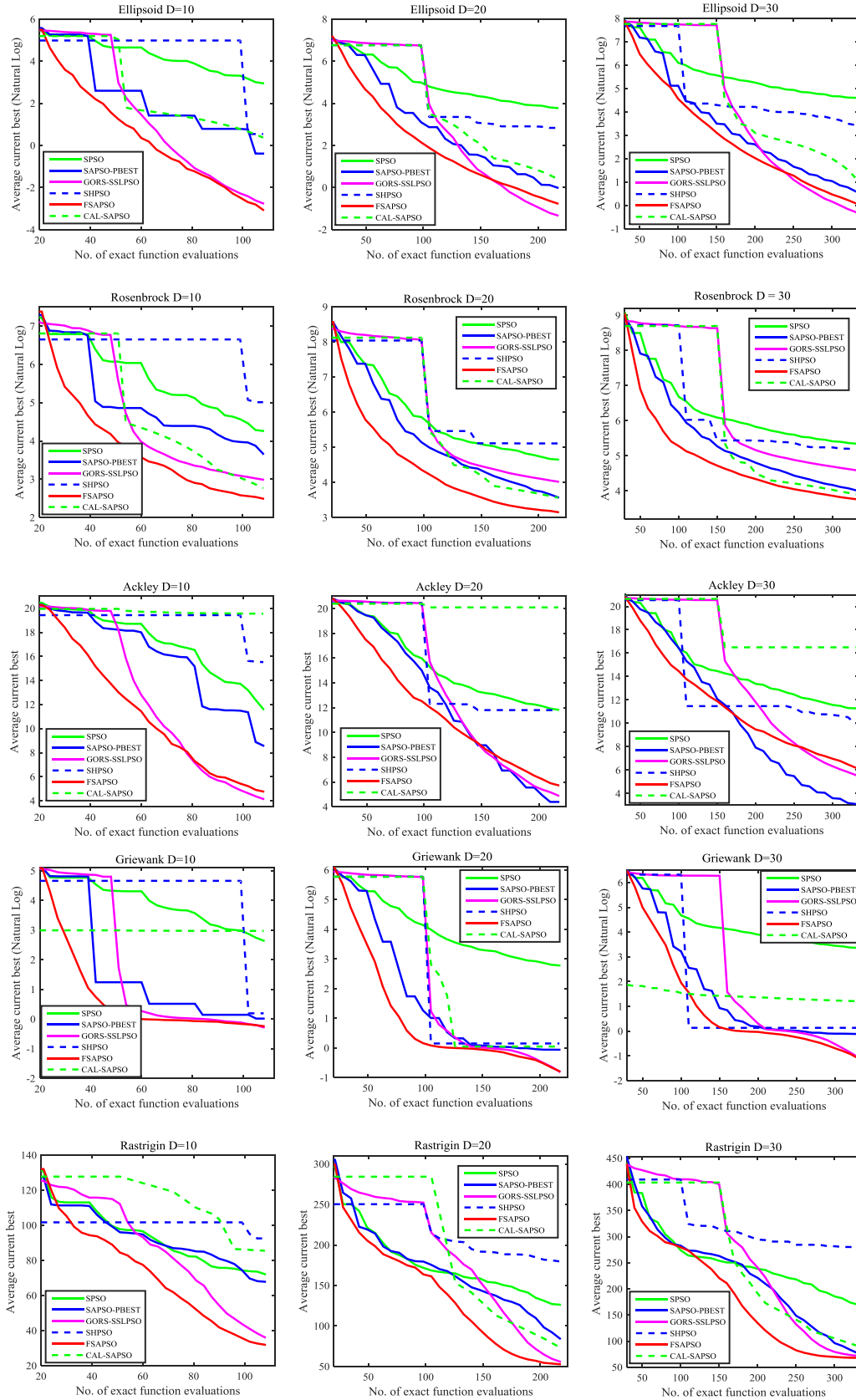
Fig. 7. Convergence profiles of the algorithms on the test problems.

Table 4

Statistics results of FSAPSO algorithm with three different criteria.

Problem	D	Criterion	Best	Worst	Median	Mean	Std	t-test
Ellipsoid	10	DF	6.94E−04	3.90E−01	1.75E−02	**4.27E−02**	8.48E−02	
		MSE	2.73E−03	2.05E−01	3.39E−02	5.64E−02	5.70E−02	≈
		Random	2.03E−03	8.13E−01	6.68E−02	1.11E−01	1.82E−01	+
Ellipsoid	20	DF	2.30E−02	1.47E+00	3.48E−01	**4.47E−01**	3.82E−01	
		MSE	7.05E−02	2.73E+00	4.10E−01	7.06E−01	7.32E−01	≈
		Random	1.19E−02	1.55E+00	3.40E−01	4.48E−01	3.91E−01	≈
Ellipsoid	30	DF	1.24E−01	2.43E+00	1.06E+00	**1.10E+00**	6.79E−01	
		MSE	4.92E−02	7.25E+00	1.71E+00	2.09E+00	1.73E+00	+
		Random	2.84E−02	3.45E+00	9.37E−01	1.26E+00	9.48E−01	≈
Rosenbrock	10	DF	7.88E+00	4.06E+01	9.79E+00	**1.17E+01**	7.03E+00	
		MSE	7.24E+00	4.47E+01	1.05E+01	1.39E+01	8.61E+00	≈
		Random	8.43E+00	4.12E+01	1.50E+01	1.71E+01	8.45E+00	+
Rosenbrock	20	DF	1.50E+01	6.07E+01	1.99E+01	**2.32E+01**	1.17E+01	
		MSE	1.90E+01	8.46E+01	2.25E+01	2.71E+01	1.47E+01	+
		Random	1.89E+01	6.89E+01	2.39E+01	2.67E+01	1.17E+01	+
Rosenbrock	30	DF	2.92E+01	6.87E+01	3.85E+01	**4.32E+01**	1.22E+01	
		MSE	2.98E+01	9.08E+01	3.99E+01	4.67E+01	1.81E+01	≈
		Random	2.94E+01	1.02E+02	4.19E+01	4.97E+01	2.00E+01	≈
Ackley	10	DF	2.63E+00	1.15E+01	3.72E+00	4.48E+00	2.06E+00	
		MSE	1.66E+00	1.33E+01	3.89E+00	**4.47E+00**	2.67E+00	≈
		Random	1.28E+00	1.15E+01	4.15E+00	4.98E+00	2.89E+00	≈
Ackley	20	DF	2.90E+00	8.31E+00	6.05E+00	5.69E+00	1.74E+00	
		MSE	2.90E+00	1.04E+01	5.98E+00	6.48E+00	2.22E+00	≈
		Random	1.34E+00	9.92E+00	4.40E+00	**4.74E+00**	2.20E+00	≈
Ackley	30	DF	3.27E+00	1.32E+01	5.75E+00	6.21E+00	2.36E+00	
		MSE	3.02E+00	9.55E+00	5.58E+00	5.79E+00	1.75E+00	≈
		Random	6.94E−01	1.10E+01	3.78E+00	**4.42E+00**	2.91E+00	−
Griewank	10	DF	3.35E−01	1.01E+00	8.46E−01	7.84E−01	2.11E−01	
		MSE	4.17E−01	1.01E+00	8.00E−01	**7.64E−01**	1.72E−01	≈
		Random	5.14E−01	1.16E+00	9.33E−01	8.79E−01	1.82E−01	≈
Griewank	20	DF	1.89E−01	6.89E−01	4.11E−01	**4.16E−01**	1.56E−01	
		MSE	1.02E−01	6.57E−01	5.01E−01	4.48E−01	1.47E−01	≈
		Random	1.41E−01	1.02E+00	5.91E−01	6.31E−01	2.67E−01	+
Griewank	30	DF	1.79E−01	5.18E−01	3.20E−01	3.33E−01	8.56E−02	
		MSE	1.85E−01	4.69E−01	3.06E−01	**3.20E−01**	8.14E−02	≈
		Random	1.30E−01	1.02E+00	6.94E−01	6.36E−01	2.82E−01	+
Rastrigin	10	DF	1.29E+01	5.54E+01	2.72E+01	**3.06E+01**	1.18E+01	
		MSE	1.41E+01	6.56E+01	3.23E+01	3.48E+01	1.27E+01	≈
		Random	1.00E+01	5.00E+01	3.38E+01	3.17E+01	1.07E+01	≈
Rastrigin	20	DF	3.08E+01	8.96E+01	4.85E+01	5.26E+01	1.55E+01	
		MSE	2.01E+01	8.17E+01	4.21E+01	**4.61E+01**	1.73E+01	≈
		Random	2.19E+01	1.25E+02	5.38E+01	6.52E+01	3.11E+01	≈
Rastrigin	30	DF	2.49E+01	9.95E+01	6.62E+01	**6.80E+01**	2.07E+01	
		MSE	3.50E+01	1.31E+02	6.97E+01	7.34E+01	2.02E+01	≈
		Random	3.45E+01	1.53E+02	8.35E+01	8.31E+01	3.10E+01	≈

Table 5

Statistical results of compared algorithms on the selected benchmarks, including the average fitness value and standard deviation shown as avg(std)

Problem	D	FSAPSO	GORS-SSLPSO	CAL-SAPSO	SHPSO	SAPSO-PBEST	SPSO
Ellipsoid	10	**4.27E−02(8.48E−02)**	5.59E−02(6.22E−02)	1.14E+00(9.95E−01)	1.72E+00(2.38E+00)	6.74E−01(3.03E−01)	1.37E+01(6.57E+00)
Ellipsoid	20	4.47E−01(3.82E−01)	**2.54E−01(2.27E−01)**	1.55E+00(5.15E−01)	1.50E+01(5.83E+00)	8.10E−01(7.09E−01)	4.28E+01(2.27E+01)
Ellipsoid	30	1.10E+00(6.79E−01)	**7.60E−01 (8.14E−01)**	3.29E+00(1.47E+00)	3.12E+01(1.19E+01)	1.83E+00(1.28E+00)	9.99E+01(3.06E+01)
Rosenbrock	10	**1.17E+01(7.03E+00)**	1.94E+01(8.52E+00)	1.59E+01(3.54E+00)	1.48E+02(8.69E+01)	3.40E+01(1.52E+01)	5.48E+01(2.19E+01)
Rosenbrock	20	**2.32E+01(1.17E+01)**	5.48E+01(1.56E+01)	3.53E+01(5.57E+00)	1.66E+02(5.09E+01)	3.44E+01(8.05E+00)	1.04E+02(4.15E+01)
Rosenbrock	30	**4.32E+01(1.22E+01)**	9.87E+01(2.33E+01)	4.94E+01(9.98E+00)	1.78E+02(4.07E+01)	5.52E+01(1.55E+01)	2.10E+02(6.42E+01)
Ackley	10	4.48E+00(2.06E+00)	**3.93E+00(2.88E+00)**	1.99E+01(4.43E−01)	1.55E+01(2.75E+00)	8.17E+00(1.12E+00)	1.11E+01(1.59E+00)
Ackley	20	5.69E+00(1.74E+00)	**4.81E+00(1.82E+00)**	2.01E+01(3.61E−15)	1.18E+01(1.80E+00)	4.37E+00(6.06E−01)	1.17E+01(1.29E+00)
Ackley	30	6.21E+00(2.36E+00)	**5.59E+00(1.41E+00)**	1.60E+01(5.62E−01)	9.96E+00(9.84E−01)	3.11E+00(3.76E−01)	1.13E+01(1.07E+00)
Griewank	10	7.84E−01(2.11E−01)	**7.40E−01(2.13E−01)**	1.13E+00(1.14E−01)	1.22E+00(1.39E−01)	1.02E+00(3.32E−02)	1.00E+01(3.66E+00)
Griewank	20	4.16E−01(1.56E−01)	**4.14E−01(1.41E−01)**	1.05E+00(3.57E−02)	1.16E+00(1.53E−01)	9.41E−01(1.12E−01)	1.61E+01(6.75E+00)
Griewank	30	**3.33E−01(8.56E−02)**	3.48E−01(1.04E−01)	9.92E−01(4.36E−02)	1.14E+00(8.22E−02)	8.90E−01(1.56E−01)	2.90E+01(8.62E+00)
Rastrigin	10	**3.06E+01(1.18E+01)**	3.46E+01(2.34E+01)	8.55E+01(2.37E+01)	8.82E+01(1.31E+01)	6.64E+01(1.04E+01)	6.58E+01(1.38E+01)
Rastrigin	20	**5.26E+01(1.55E+01)**	5.45E+01(2.79E+01)	7.35E+01(1.53E+01)	1.77E+02(2.53E+01)	8.07E+01(2.16E+01)	1.26E+02(2.03E+01)
Rastrigin	30	**6.80E+01(2.07E+01)**	7.17E+01(4.05E+01)	9.08E+01(1.93E+01)	2.78E+02(1.86E+01)	7.67E+01(1.64E+01)	1.70E+02(1.79E+01)

From the results shown in Tables 7 and 8, we can observe that FSAPSO, SAPSO-PBEST, and ESAO algorithms achieve the best mean values on four/one/three problems respectively. FS-APSO algorithm significantly outperforms the SPSO algorithm

Table 6
Wilcoxon rank sum test calculated at a significance level of $\alpha = 0.05$.

Problem	D	GORS-SSLPSO	SHPSO	SAPSO-PBEST	SPSO
Ellipsoid	10	≈	+	+	+
Ellipsoid	20	≈	+	+	+
Ellipsoid	30	−	+	≈	+
Rosenbrock	10	+	+	+	+
Rosenbrock	20	+	+	+	+
Rosenbrock	30	+	+	+	+
Ackley	10	−	+	+	+
Ackley	20	≈	+	−	+
Ackley	30	≈	+	−	+
Griewank	10	≈	+	+	+
Griewank	20	≈	+	+	+
Griewank	30	≈	+	+	+
Rastrigin	10	≈	+	+	+
Rastrigin	20	≈	+	+	+
Rastrigin	30	≈	+	≈	+
+ /≈/−		3/10/2	15/0/0	11/2/2	15/0/0

Table 8
Wilcoxon rank sum test calculated at a significance level of $\alpha = 0.05$.

Problem	GORS-SSLPSO	SHPSO	SA-COSO	SAPSO-PBEST	SPSO
Ellipsoid	+	+	+	+	+
Rosenbrock	+	+	+	+	+
Ackley	+	+	+	−	+
Griewank	+	+	+	+	+
Rastrigin	+	+	+	≈	+
F10	+	+	+	+	+
F16	+	≈	+	≈	+
F19	−	−	+	−	+
+ /≈/−	7/0/1	6/1/1	8/0/0	4/2/2	8/0/0

on all problems. FSAPSO algorithm significantly outperforms the SAPSO-PBEST algorithm on four problems, while it is outperformed by SAPSO-PBEST algorithm on two problems. The advantage of the FSAPSO algorithm is not so obvious on 50-dimension problems. The fact that FSAPSO only selects the best and most uncertain particles at each iteration may cause the algorithm to be trapped in a local optimum as the search space of high-dimensional problems is too large. Two candidates are exactly evaluated in ESAO [29]: the best offspring produced by differential evolution and the optimal solution of the local surrogate model. FSAPSO algorithm outperforms the ESAO algorithm on four out of six problems, while they obtain comparable results on Ackley. This indicates the two criteria used in FSAPSO can make FSAPSO achieve good performance. SA-COSO [27] uses two surrogate-assisted PSO algorithms collaboratively to search for the global optimum. FSAPSO algorithm significantly outperforms the SA-COSO algorithm on all eight problems. SHPSO [26] algorithm uses the same pre-screening strategy as the SAPSO-PBEST algorithm, but it uses a local surrogate model to pre-evaluate particles and SL-PSO is used to search the optimum of the model. SHPSO algorithm achieves worse performance than the SAPSO-PBEST algorithm on most problems, which may be that the use of the local model makes the algorithm trapped in a local optimum of the model. However, it shows better results on F16 and F19 which are very complex functions. GORS-SSLPSO [31] restarts SL-PSO after a few generations, and balances the exploration and exploitation by exactly evaluating the best particle in the SL-PSO. FSAPSO algorithm significantly outperforms GORS-SSLPSO algorithm on seven out of eight problems.

From the performance curves presented in Fig. 8, it can be seen that the FSAPSO algorithm converges much faster than other algorithms on most of the problems. The global optimum of the FSAPSO algorithm improves faster than the SPSO algorithm. It indicates that using the surrogate can improve the quality of final solutions within the same number of FEs on high-dimensional problems. FSAPSO converges a little slower than SAPSO-PBEST on Ackley, F16, and F19. The criterion in SAPSO-PBEST can select more particles to explore promising areas. This may explain that the SAPSO-PBEST algorithm still has more potential to improve the final solution in the later stage. In addition, FSAPSO converges faster than EASO on five out of six problems, while ESAO converges faster on F19. F19 is a complex multi-modal function, and ESAO uses differential evolution to evolve the population. The good exploration ability of the differential evolution algorithm may result in the good performance of ESAO on F19. FSAPSO can obtain a better optimum than SA-COSO within the same FEs on all problems. The strategies in SA-COSO distribute plenty of resource on exploration may induce its bad performance. FSAPSO converges faster than GORS-SSLPSO on six problems, while GORS-SSLPSO converges faster than FSAPSO on F19. They obtain a similar convergence rate on F10. GORS-SSLPSO uses SL-PSO to explore the search space. This may induce to better exploration ability of it than FSAPSO on F19.

From the above results, we can make the following observations. First, the surrogate model can significantly improve the performance of PSO on high-dimensional problems. Second, the FSAPSO algorithm significantly outperforms GORS-SSLPSO, SHPSO, SA-COSO, ESAO, SAPSO-PBEST and SPSO algorithms on most benchmark functions in the context of the limited computational budget scenario. In addition, the FSAPSO algorithm converges faster than other algorithms on most of the problems.
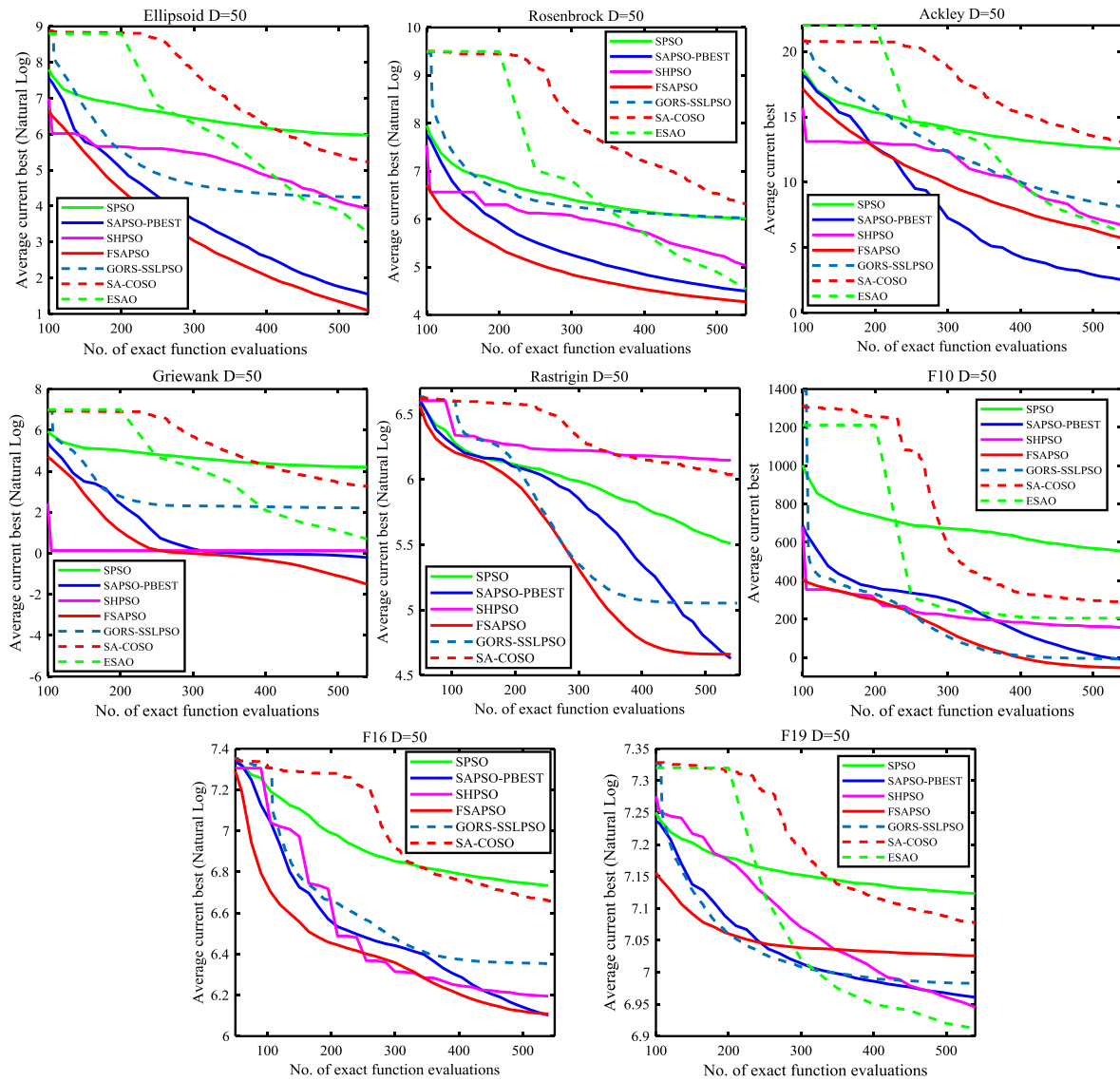
Table 7
Statistical results of compared algorithms on the selected benchmarks, including the average fitness value and standard deviation shown as avg(std).

Problem	FSAPSO	GORS-SSLPSO	SHPSO	SA-COSO	ESAO	SAPSO-PBEST	SPSO
Ellipsoid	**2.81E+00**	6.93E+01	4.97E+01	1.77E+02	2.22E+01	4.52E+00	3.94E+02
	(1.89E+00)	(1.56E+02)	(1.07E+01)	(5.29E+01)		(3.22E+00)	(7.81E+01)
Rosenbrock	**7.08E+01**	4.10E+02	1.47E+02	5.33E+02	8.56E+01	8.79E+01	4.07E+02
	(1.83E+01)	(4.69E+02)	(3.26E+01)	(9.41E+01)		(2.03E+01)	(7.02E+01)
Ackley	5.59E+00	8.04E+00	6.50E+00	1.29E+01	6.00E+00	**2.32E+00**	1.25E+01
	(1.07E+00)	(3.43E+00)	(5.00E−01)	(1.11E+00)		**(3.32E−01)**	(8.08E−01)
Griewank	**1.99E−01**	9.12E+00	1.15E+00	2.50E+01	1.82E+00	7.77E−01	6.65E+01
	(5.37E−02)	(2.75E+01)	(9.32E−02)	(4.67E+00)		(1.12E−01)	(1.33E+01)
Rastrigin	1.06E+02	1.57E+02	4.68E+02	4.15E+02	N/A	**9.77E+01**	2.44E+02
	(2.70E+01)	(4.24E+01)	(4.17E+01)	(3.50E+01)		**(1.98E+01)**	(2.48E+01)
F10	**−5.56E+01**	−8.80E+00	1.55E+02	2.87E+02	2.05E+02	−1.49E+01	5.51E+02
	(5.78E+01)	(3.90E+01)	(2.67E+01)	(5.01E+01)		(5.14E+01)	(9.15E+01)
F16	4.49E+02	5.72E+02	4.89E+02	7.76E+02	N/A	**4.44E+02**	8.39E+02
	(1.00E+02)	(1.29E+02)	(4.37E+01)	(8.97E+01)		**(9.24E+01)**	(1.03E+02)
F19	1.12E+03	1.08E+03	1.03E+03	1.18E+03	**1.00E+03**	1.05E+03	1.24E+03
	(6.59E+01)	(6.46E+01)	(2.02E+01)	(4.11E+01)		(3.46E+01)	(3.74E+01)

Fig. 8. Convergence profiles of the algorithms on 50-dimension functions.

4.4. Effects of number of nearest neighbors for the DF criterion

The uncertainty of a candidate is usually influenced by the nearest evaluated neighbors, and a remote observed point hardly has a significant influence on it. Moreover, the DF criterion is also devised to be applied in local areas as a large neighbor will make the variance of fitness values invalid. Therefore, the number of nearest neighbors (k) for the DF criterion could not be too large. Besides, k should be greater than 1 as only the distance information is considered when $k = 1$. The number of initial sample points for $D = 10, 20$ is 20, so the maximum value of k is set as 20. Five different values (1, 3, 5, 10, 20) are selected to test its influence on the final result, and five benchmark problems with $D = 10, 20, 30$ are used in the experimental study. The average best objective function values obtained by the algorithms over 30 independent runs are shown in Table 9, and the best mean results of individual instances are highlighted. For all tested problems, the worst results are obtained when $k = 1$, and similar results are obtained for the remains. The results indicate that it is better to use the distance and fitness value information to estimate uncertainty than just using the distance. There is no significant difference between $k = 3$ and the others, so the

parameter k hardly has any significant influence on the proposed algorithm

4.5. Effects of the used surrogate model

To test the effects of different surrogate models on the proposed FSAPSO algorithm, we compare FSAPSO with its variant FSAPSO1. FSAPSO1 is similar to FSAPSO except that it uses the Kriging model to predict the fitness values. For a second-order polynomial model, the number of terms in the quadratic model is $(D + 1)(D + 2)/2$ and the number of interaction terms is $D(D -1)/2$. D is the number of input variables. Many samples are required if the polynomial model is used. Therefore, we do not use the polynomial model. Four benchmark problems with dimensions $D = 10$ and 30 are used to test the two algorithms. The average best objective function values obtained by the two algorithms over 30 independent runs are shown in Table 10, and the best mean results of individual instances are highlighted. FSAPSO obtains better mean values than FSAPSO1 on seven out of eight problems, while FSAPSO1 obtains better mean values than FSAPSO on one problem. These results indicate the superiority of the RBF model over the Kriging model. In general, different types of models are suitable for their corresponding problems.

Table 9

Statistics results of FSAPSO with different k on selected benchmarks, including the average fitness value and standard deviation shown as avg(std)

Problem	D	1	3	5	10	20
Ellipsoid	10	1.21E+00(1.63E+00)	**4.27E−02(8.48E−02)**	7.06E−02(6.89E−02)	1.00E−01(1.47E−01)	7.96E−02(1.03E−01)
Ellipsoid	20	2.01E+00(3.09E+00)	4.47E−01(3.82E−01)	5.56E−01(4.97E−01)	4.02E−01(4.35E−01)	**3.86E−01(3.13E−01)**
Ellipsoid	30	4.20E+00(1.15E+01)	1.10E+00(6.79E−01)	**9.04E−01(5.53E−01)**	1.06E+00(7.91E−01)	1.21E+00(9.40E−01)
Rosenbrock	10	6.66E+01(9.93E+01)	1.17E+01(7.03E+00)	**1.11E+01(4.26E+00)**	1.58E+01(1.78E+01)	1.74E+01(2.00E+01)
Rosenbrock	20	1.82E+02(5.82E+02)	2.32E+01(1.17E+01)	2.38E+01(9.05E+00)	**2.13E+01(3.49E+00)**	2.31E+01(8.66E+00)
Rosenbrock	30	6.18E+01(3.49E+01)	4.32E+01(1.22E+01)	4.10E+01(1.31E+01)	**3.86E+01(1.34E+01)**	4.02E+01(1.27E+01)
Ackley	10	1.25E+01(5.66E+00)	**4.48E+00(2.06E+00)**	4.49E+00(3.75E+00)	5.14E+00(2.82E+00)	5.28E+00(3.24E+00)
Ackley	20	1.12E+01(2.98E+00)	5.69E+00(1.74E+00)	5.91E+00(2.26E+00)	5.51E+00(2.44E+00)	5.92E+00(2.41E+00)
Ackley	30	8.59E+00(4.19E+00)	6.21E+00 (2.36E+00)	**5.34E+00(1.80E+00)**	5.61E+00(1.50E+00)	5.85E+00(2.07E+00)
Griewank	10	9.91E−01(2.15E−01)	7.84E−01(2.11E−01)	8.89E−01(1.63E−01)	7.65E−01(2.05E−01)	**7.43E−01(1.90E−01)**
Griewank	20	1.30E+00(1.30E+00)	**4.16E−01(1.56E−01)**	5.25E−01(1.44E−01)	4.96E−01(1.66E−01)	4.63E−01(2.07E−01)
Griewank	30	2.06E+00(3.17E+00)	3.33E−01(8.56E−02)	**2.92E−01(1.12E−01)**	3.04E−01(1.19E−01)	3.26E−01(1.73E−01)
Rastrigin	10	6.81E+01(3.14E+01)	3.06E+01(1.18E+01)	**2.93E+01(1.34E+01)**	3.12E+01(1.44E+01)	2.95E+01(1.21E+01)
Rastrigin	20	1.03E+02(3.26E+01)	5.26E+01(1.55E+01)	4.43E+01(1.39E+01)	4.34E+01(9.72E+00)	**4.15E+01(1.50E+01)**
Rastrigin	30	1.19E+02(6.67E+01)	6.80E+01(2.07E+01)	7.08E+01(2.81E+01)	**6.53E+01(2.30E+01)**	6.56E+01(2.43E+01)

Table 10

Statistics results of FSAPSO and FSAPSO1 algorithms.

Problem	D	FSAPSO	FSAPSO1
Ellipsoid	10	4.27E−02(8.48E−02)	**2.71E−02 (4.91E−02)**
Ellipsoid	30	**1.10E+00(6.79E−01)**	1.96E+00(1.32E+00)
Rosenbrock	10	**1.17E+01(7.03E+00)**	5.02E+01(1.74E+01)
Rosenbrock	30	**4.32E+01(1.22E+01)**	9.91E+01(3.77E+01)
Ackley	10	**4.48E+00(2.06E+00)**	1.52E+01(2.11E+00)
Ackley	30	**6.21E+00(2.36E+00)**	1.12E+01(1.50E+00)
Griewank	10	**7.84E−01(2.11E−01)**	1.78E+00(1.85E+00)
Griewank	30	**3.33E−01(8.56E−02)**	3.68E+00(2.48E+00)

However, the computational complexity of the Kriging model is high when the dimension of the problem is bigger than 10. The RBF model has shown good performance in other surrogate-assisted evolutionary algorithms. Therefore, the RBF model is used in this paper.

4.6. Results analysis and discussion

All strategies used by SAPSO-PBEST and FSAPSO are the same except the pre-screening strategies. SAPSO-PBEST may select more candidates for exact evaluations at each iteration as the PBEST criterion is used. The better performance achieved by the FSAPSO algorithm indicates that evaluating fewer candidates to balance exploration and exploitation at each generation is better for problems with $D = 10, 20, 30$. As the total number of FEs is limit, selecting too many particles for exact evaluations at each iteration may make the algorithm stop without sufficient exploitation. However, FSAPSO and SAPSO-PBEST algorithms show comparable performance on 50-dimension problems. If enough FEs are provided, the PBEST-based criterion may be better as more exploration is allowed. Although CAL-SAPSO and FSAPSO algorithms both use the performance-based and uncertainty-based criteria to select candidates, the FSAPSO algorithm significantly outperforms the CAL-SAPSO algorithm on most problems. CAL-SAPSO algorithm is actually a model-based global optimization algorithm. It exactly evaluates the optimum of an ensemble at every iteration. In addition, the most uncertain point is exactly evaluated when the current global optimum is not improved to enhance the accuracy of the model and avoid the stagnation of the algorithm. There are many unexplored areas in the entail search space, and the most uncertain point is selected from the entail space in CAL-SAPSO. Although this is good to improve the model accuracy, its contribution to the improvement of the global optimum is limited. On the other hand, the exactly evaluated point in the FSAPSO algorithm is selected from the particle swarm. The whole swarm spreads in the entail space in the beginning, so evaluating the particle with the largest uncertainty is beneficial for exploration at this stage. The dispersion of the swarm grows smaller with the evolution of the swarm. Exactly evaluating the particle with great uncertainty is beneficial to explore the area near the current global optimum, which can improve the global optimum as the area is more promising and smaller than the entail space. In addition, if the uncertain particle is promising, the swarm might be attracted to the area. The GORS-SSLPSO algorithm uses an SL-PSO algorithm to solve the global optimum of a global RBF model, and the optimum will be exactly evaluated. However, the SL-PSO only evolves a certain generation, then an RBF model is updated and certain better points were selected to restart the swarm. The GORS-SSLPSO has a good exploratory ability in the beginning as the SL-PSO only evolves certain generations, which also avoids the stagnation of the algorithm in a local optimum of the model. It gradually varies from exploration to exploitation when the top-ranked points are near the current optimum. FSAPSO and GORS-SSLPSO algorithms show comparative performance on function with $D = 10, 20, 30$ in $11D$ FEs, which may be that they all have a good balance of exploration and exploitation with a limited number of FEs. In addition, FSAPSO can quickly obtain better solutions than other algorithms on most of the problems. The performance-based criterion selects the particle with the best fitness value and the optimum of the global surrogate model for exact evaluations. Evaluating the best particle can attract the swarm to exploit the promising areas once a promising area is found. In addition, the global surrogate model can roughly approximate the whole fitness landscape of the problem, and the optimum of the model may indicate the real promising area for the swarm. The swarm can quickly exploit new promising area by evaluating the optima of the surrogate model and use it to replace the global best if it is better. Therefore, the fast convergence rate of FSAPSO mainly owes to the performance-based criterion. However, the algorithm may quickly be trapped into the optimum of the surrogate model if only the performance-based criterion is used as the surrogate model is not accurate and the exploratory particle cannot be exactly evaluated. The uncertainty-based criterion selects the particles with the greatest uncertainty for the exact FEs. This can enhance the exploration in not-well-explored areas, enhance the accuracy of the surrogate model and relieve the premature stagnation of the algorithm. Overall, the collaboration between the performance-based criterion and uncertainty-based criterion accelerates the convergence rate of FSAPSO through the whole evolutionary process.

Although FSAPSO algorithm shows promising results on 50-dimension problems used in the paper, the strategy used in FSAPSO may not be suitable for higher-dimensional problems. The high-dimensional space is too large, more exploration is needed to get a better solution.
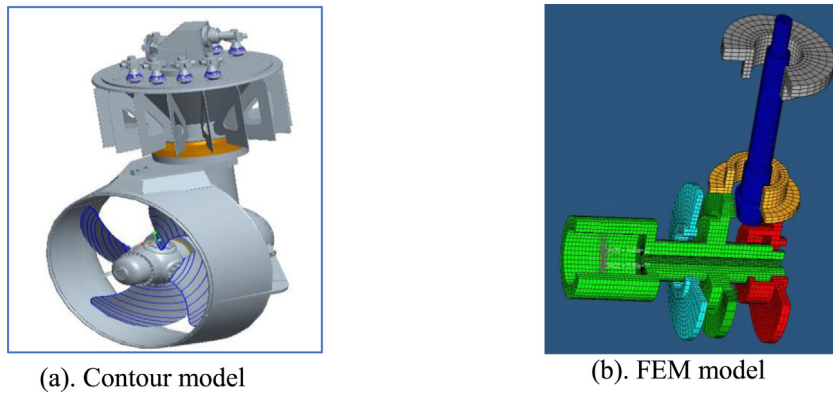
(a). Contour model

(b). FEM model

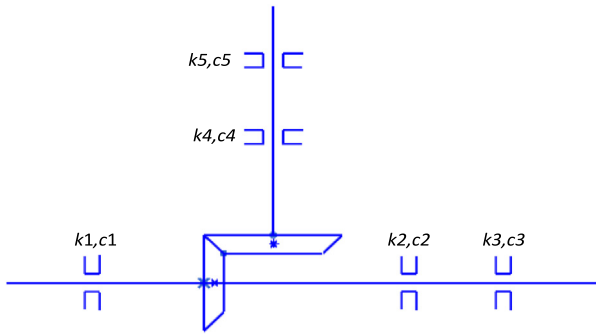**Fig. 9.** Different models of the propeller.

Fig. 10. Axis structure diagram of propeller.

4.7. Optimal design of bearings in an all-direction propeller

The all-direction propeller is widely used in ships and drilling platforms that require high accuracy of positioning. It is important to design an all-direction propeller with a good ability of resisting vibration to improve its performance. Besides, the power flow is a significant dynamic index to reflect the degree of vibration. The paper takes the power flow as an objective to obtain an optimal design of the all-direction propeller with a good dynamic performance. However, the calculation of the power flow needs computationally expensive finite element analysis simulations, which makes the acquisition of the power flow a black-box problem. Therefore, surrogates can be used to approximate the relationship between the power flow of the propeller and several key structural parameters including the stiffness and damping coefficients of the bearings in the propeller drive system to improve the design efficiency. To show the optimization problem, different models of the propeller are shown in Fig. 9. The finite element model (FEM) is used for power flow simulation.

The optimization problem can be formulated as:

$$
\begin{aligned}
&find \ \ K = [k_1, k_2, k_3, k_4, k_5, c_1, c_2, c_3, c_4, c_5]^T \\
&min \ \ \ \ \ P^* = \sqrt{\sum_{j=1}^{KP}(P_j)^2} \\
&P_j = \omega/2\pi \int_0^{\omega/2\pi} Re(f_i)Re(v_j)\mathrm{d}t \\
&s.t. \ \ \ \ k_i^L \leq k_i \leq k_i^U, c_i^L \leq c_i \leq c_i^U, \ \ \ \ i = 1, 2, 3, 4, 5
\end{aligned}
\tag{13}
$$

where k_i and c_i are the stiffness and damping coefficients of different bearings in the propeller, k_i^L and k_i^U are the lower and upper bounds of k_i, respectively; and c_i^L and c_i^U are the lower and upper bounds of c_i, respectively; P_j is the power flow at the selected evaluation point j in the propeller; and KP is the sum of evaluation points distributed in different bearings,
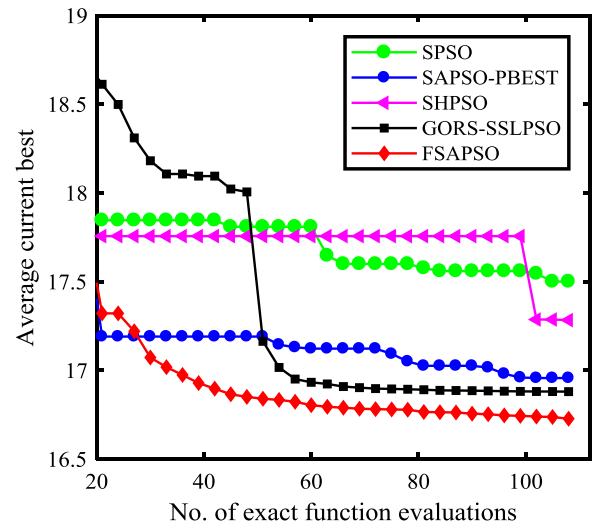

Fig. 11. Convergence curves of FSAPSO, GORS-SSLPSO, SHPSO, SAPSO-PBEST and SPSO algorithms for the propeller design problem.

which is set as 10 in the paper. f_j and v_j represents the force and velocity in the jth evaluation point. The power flow is the sum of the product of force and speed in a cycle as (13) defines. More details about power flow calculation and application for vibration control can be found in [48]. The initial values of design variables are given as: $[k_1, k_2, k_3, k_4, k_5, c_1, c_2, c_3, c_4, c_5,]_{\text{initial}} = [3, 4, 3, 3, 4, 1.5, 2, 1.5, 1.5, 2]$. The corresponding initial power flow of the propeller is $P^* = 22.58W$. The bounds are defined as $k_1, k_3, k_4 \in [2, 4] \times 10^7$ N/m,$k_2, k_5 \in [2, 6] \times 10^7$N/m, $c_1, c_3, c_4 \in [1, 2] \times 10^7$Ns/m, $c_2, c_5 \in [1, 3] \times 10^7$Ns/m. The axis structure diagram of the propeller is shown in Fig. 10.

The FSAPSO, GORS-SSLPSO, SHPSO, SAPSO-PBEST and SPSO algorithms are used to optimize the all-direction propeller. The optimization process is stopped when the number of all simulation points reaches 110 as it involves expensive power flow simulation. Each algorithm repeats ten times for robustness comparison as running a power flow simulation takes about 5 min. The statistics on the best objective function values obtained by the four algorithms over 10 independent runs are shown in Table 11 and the convergence curves of different algorithms for the propeller design problem are depicted in Fig. 11. It is evident that the proposed FSAPSO algorithm obtained the best mean value as compared with four other algorithms, and it achieves the fastest rate of convergence.

Table 11
Statistical results of compared algorithms on the propeller design problem.

Algorithm	Best	Worst	Median	Mean	Std
FSAPSO	1.6648E+01	1.6993E+01	1.6681E+01	**1.6725E+01**	**9.3700E−02**
GORS-SSLPSO	1.6648E+01	1.7244E+01	1.6879E+01	1.6880E+01	1.5002E−01
SHPSO	1.6916E+01	1.7668E+01	1.7192E+01	1.7263E+01	2.3529E−01
SAPSO-PBEST	1.6704E+01	1.7365E+01	1.6928E+01	1.6957E+01	1.6064E−01
SPSO	1.6950E+01	1.7783E+01	1.7580E+01	1.7501E+01	2.3217E−01

5. Conclusion and future work

This paper proposes a fast surrogate-assisted particle swarm optimization (FSAPSO) algorithm which requires only 11D FEs to solve medium scaled computationally expensive optimization problems. Particles with the best prediction and the greatest uncertainty are evaluated to explore and exploit the search space. An uncertainty-based criterion considering the distance and fitness value information is proposed to evaluate uncertainties of particles. Comprehensive analyses demonstrate the effectiveness of the uncertainty-based criterion. The proposed algorithm is compared with seven state-of-the-art algorithms on seven widely used benchmark problems of different dimensionalities ranging from 10 to 50. The experimental results indicate that the proposed algorithm can obtain significantly better results and achieve a faster convergence rate than the alternative competitors on most of the benchmark problems. The experimental results also indicate that exactly evaluating uncertain candidates in promising areas is better than in the whole search space with a limited number of FEs. Finally, the proposed FSAPSO algorithm is used to solve a 10-dimension propeller design problem, and the results confirm the good performance of the proposed algorithm.

The exploratory ability of the FSAPSO algorithm is limited and it may not very suitable for problems with more than 50 dimensions. This limitation is being addressed in our ongoing research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Fan Li: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Visualization. **Weiming Shen:** Resources, Writing - review & editing, Supervision. **Xiwen Cai:** Resources, Writing - review & editing. **Liang Gao:** Resources, Supervision, Project administration, Funding acquisition. **G. Gary Wang:** Validation, Writing - review & editing.

Acknowledgments

References

[1] Y. Zhang, X. Li, L. Gao, P. Li, A new subset based deep feature learning method for intelligent fault diagnosis of bearing, Expert Syst. Appl. 110 (2018) 125–142, http://dx.doi.org/10.1016/j.eswa.2018.05.032.

[2] B. Zhang, Q.-K. Pan, L. Gao, L.-L. Meng, X.-Y. Li, K.-K. Peng, A three-stage multiobjective approach based on decomposition for an energy-efficient hybrid flow shop scheduling problem, IEEE Trans. Syst. Man Cybern. Syst. (2019).

[3] A. Nickabadi, M.M. Ebadzadeh, R. Safabakhsh, A novel particle swarm optimization algorithm with adaptive inertia weight, Appl. Soft Comput. 11 (4) (2011) 3658–3670, http://dx.doi.org/10.1016/j.asoc.2011.01.037.

[4] W.H. Lim, N.A. Mat Isa, Teaching and peer-learning particle swarm optimization, Appl. Soft Comput. 18 (2014) 39–58, http://dx.doi.org/10.1016/j.asoc.2014.01.009.

[5] T. Goel, R.T. Hafkta, W. Shyy, Comparing error estimation measures for polynomial and kriging approximation of noise-free functions, Struct. Multidiscip. Optim. 38 (5) (2008) 429–442, http://dx.doi.org/10.1007/s00158-008-0290-z.

[6] K.H. Liang, Evolutionary search of approximated n-dimensional landscapes, Int. J. Knowl.-Based Intell. Eng. Syst. 4 (2000).

[7] W. Roux, N. Stander, R. Haftka, Response surface approximations for structural optimization, 1996, http://dx.doi.org/10.2514/6.1996-4042.

[8] R.L. Hardy, Multiquadric equations of topography and other irregular surfaces, J. Geophys. Res. 76 (8) (1971) 1905–1915, http://dx.doi.org/10.1029/JB076i008p01905.

[9] H.M. Gutmann, A radial basis function method for global optimization, J. Global Optim. 19 (3) (2001) 201–227.

[10] E. Acar, Optimizing the shape parameters of radial basis functions: An application to automobile crashworthiness, Proc. Inst. Mech. Eng. D 224 (12) (2010) 1541–1553, http://dx.doi.org/10.1243/09544070jauto1560.

[11] H. Dong, B. Song, P. Wang, Z. Dong, Hybrid surrogate-based optimization using space reduction (HSOSR) for expensive black-box functions, Appl. Soft Comput. 64 (2018) 641–655, http://dx.doi.org/10.1016/j.asoc.2017.12.046.

[12] D.R. Jones, M. Schonlau, W.J. Welch, J. Global Optim. 13 (4) (1998) 455–492, http://dx.doi.org/10.1023/a:1008306431147.

[13] M.D. Parno, T. Hemker, K.R. Fowler, Applicability of surrogates to improve efficiency of particle swarm optimization for simulation-based problems, Eng. Optim. 44 (5) (2012) 521–535, http://dx.doi.org/10.1080/0305215x.2011.598521.

[14] H. Dong, B. Song, Z. Dong, P. Wang, SCGOSR: Surrogate-based constrained global optimization using space reduction, Appl. Soft Comput. 65 (2018) 462–477, http://dx.doi.org/10.1016/j.asoc.2018.01.041.

[15] V. Volz, G. Rudolph, B. Naujoks, Investigating uncertainty propagation in surrogate-assisted evolutionary algorithms, 2017, pp. 881–888, http://dx.doi.org/10.1145/3071178.3071249.

[16] S.M. Clarke, J.H. Griebsch, T.W. Simpson, Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses, in: ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2005, pp. 1077–1087.

[17] D. Zhao, D. Xue, A comparative study of metamodeling methods considering sample quality merits, Struct. Multidiscip. Optim. 42 (6) (2010) 923–938, http://dx.doi.org/10.1007/s00158-010-0529-3.

[18] R. Jin, W. Chen, T.W. Simpson, Comparative studies of metamodelling techniques under multiple modelling criteria, Struct. Multidiscip. Optim. 23 (1) (2000) 1–13.

[19] R. Jin, W. Chen, T. Simpson, Comparative studies of metamodeling techniques under multiple modeling criteria, 2000, http://dx.doi.org/10.2514/6.2000-4801.

[20] J. Gu, G.Y. Li, Z. Dong, Hybrid and adaptive meta-model-based global optimization, Eng. Optim. 44 (1) (2012) 87–104, http://dx.doi.org/10.1080/0305215x.2011.564768.

[21] E. Acar, M. Rais-Rohani, Ensemble of metamodels with optimized weight factors, Struct. Multidiscip. Optim. 37 (3) (2008) 279–294, http://dx.doi.org/10.1007/s00158-008-0230-y.

[22] T. Goel, R.T. Haftka, W. Shyy, N.V. Queipo, Ensemble of surrogates, Struct. Multidiscip. Optim. 33 (3) (2006) 199–216, http://dx.doi.org/10.1007/s00158-006-0051-9.

[23] R.G. Regis, Particle swarm with radial basis function surrogates for expensive black-box optimization, J. Comput. Sci. 5 (1) (2014) 12–23, http://dx.doi.org/10.1016/j.jocs.2013.07.004.

[24] C. Sun, Y. Jin, J. Zeng, Y. Yu, A two-layer surrogate-assisted particle swarm optimization algorithm, Soft Comput. 19 (6) (2014) 1461–1475, http://dx.doi.org/10.1007/s00500-014-1283-z.

[25] J. Park, K.-Y. Kim, Meta-modeling using generalized regression neural network and particle swarm optimization, Appl. Soft Comput. 51 (2017) 354–369, http://dx.doi.org/10.1016/j.asoc.2016.11.029.

[26] H. Yu, Y. Tan, J. Zeng, C. Sun, Y. Jin, Surrogate-assisted hierarchical particle swarm optimization, Inform. Sci. 454–455 (2018) 59–72, http://dx.doi.org/10.1016/j.ins.2018.04.062.

[27] C. Sun, Y. Jin, R. Cheng, J. Ding, J. Zeng, Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems, IEEE Trans. Evol. Comput. 21 (4) (2017) 644–660, http://dx.doi.org/10.1109/tevc.2017.2675628.

[28] Z. Yang, H. Qiu, L. Gao, X. Cai, C. Jiang, L. Chen, A surrogate-assisted particle swarm optimization algorithm based on efficient global optimization for expensive black-box problems, Eng. Optim. (2018) 1–18, http://dx.doi.org/10.1080/0305215x.2018.1477940.

[29] X. Wang, G.G. Wang, B. Song, P. Wang, Y. Wang, A novel evolutionary sampling assisted optimization method for high dimensional expensive problems, IEEE Trans. Evol. Comput. (2019) 1, http://dx.doi.org/10.1109/tevc.2019.2890818.

[30] J. Tian, C. Sun, Y. Tan, J. Zeng, Granularity-based surrogate-assisted particle swarm optimization for high-dimensional expensive optimization, Knowl.-Based Syst. (2019) http://dx.doi.org/10.1016/j.knosys.2019.06.023.

[31] H. Yu, Y. Tan, C. Sun, J. Zeng, A generation-based optimal restart strategy for surrogate-assisted social learning particle swarm optimization, Knowl.-Based Syst. (2018) http://dx.doi.org/10.1016/j.knosys.2018.08.010.

[32] J. Tian, Y. Tan, J. Zeng, C. Sun, Y. Jin, Multi-objective infill criterion driven Gaussian process assisted particle swarm optimization of high-dimensional expensive problems, IEEE Trans. Evol. Comput. PP (99) (2018) 1.

[33] Y. Jin, H. Wang, T. Chugh, D. Guo, K. Miettinen, Data-driven evolutionary optimization: An overview and case studies, IEEE Trans. Evol. Comput. (2018) 1, http://dx.doi.org/10.1109/tevc.2018.2869001.

[34] J. Branke, C. Schmidt, Faster convergence by means of fitness estimation, Soft Comput. 9 (1) (2003) 13–20, http://dx.doi.org/10.1007/s00500-003-0329-4.

[35] H. Wang, Y. Jin, J. Doherty, Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems, IEEE Trans. Cybern. 47 (9) (2017) 2664–2677, http://dx.doi.org/10.1109/TCYB.2017.2710978.

[36] D. Guo, Y. Jin, J. Ding, T. Chai, Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems, IEEE Trans. Cybern. (2018) 1–14, http://dx.doi.org/10.1109/tcyb.2018.2794503.

[37] R.G. Regis, C.A. Shoemaker, Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization, Eng. Optim. 45 (5) (2013) 529–555, http://dx.doi.org/10.1080/0305215x.2012.687731.

[38] F. Li, X. Cai, L. Gao, Appl. Soft Comput. 74 (2019) 291–305, http://dx.doi.org/10.1016/j.asoc.2018.10.037,

[39] X. Cai, L. Gao, X. Li, H. Qiu, Surrogate-guided differential evolution algorithm for high dimensional expensive problems, Swarm Evol. Comput. 48 (2019) 288–311, http://dx.doi.org/10.1016/j.swevo.2019.04.009.

[40] B. Liu, Q. Zhang, G.G.E. Gielen, A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems, IEEE Trans. Evol. Comput. 18 (2) (2014) 180–192, http://dx.doi.org/10.1109/tevc.2013.2248012.

[41] X. Cai, L. Gao, X. Li, Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems, IEEE Trans. Evol. Comput. (2019).

[42] X. Cai, L. Gao, F. Li, Sequential approximation optimization assisted particle swarm optimization for expensive problems, Appl. Soft Comput. (2019) 105659.

[43] X. Cai, H. Qiu, L. Gao, C. Jiang, X. Shao, An efficient surrogate-assisted particle swarm optimization algorithm for high-dimensional expensive problems, Knowl.-Based Syst. (2019) 104901.

[44] A.I.J. Forrester, A.J. Keane, Recent advances in surrogate-based optimization, Prog. Aerosp. Sci. 45 (1–3) (2009) 50–79, http://dx.doi.org/10.1016/j.paerosci.2008.11.001.

[45] Y. Shi, R. Eberhart, A Modified Particle Swarm Optimizer, Springer, Berlin Heidelberg, 1998, p. 439.

[46] Y. Zhang, S. Wang, G. Ji, A comprehensive survey on particle swarm optimization algorithm and its applications, Math. Probl. Eng. 2015 (2015) 1–38, http://dx.doi.org/10.1155/2015/931256.

[47] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb Y. Chen, A. Auger, Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, 2005.

[48] Y.P. Xiong, J.T. Xing, W.G. Price, A general linear mathematical model of power flow analysis and control for integrated structure–control systems, J. Sound Vib. 267 (2) (2003) 301–334.