Soft Computing (2021) 25:15051–15065
https://doi.org/10.1007/s00500-021-06348-2

OPTIMIZATION

An adaptive surrogate-assisted particle swarm optimization for expensive problems

Xuemei Li[1] · Shaojun Li[1]

Accepted: 24 September 2021 / Published online: 8 October 2021

Abstract
To solve engineering problems with evolutionary algorithms, many expensive function evaluations (FEs) are required. To alleviate this difficulty, surrogate-assisted evolutionary algorithms (SAEAs) have attracted increasingly more attention in both academia and industry. Most existing SAEAs either waste computational resources due to the lack of accuracy of the surrogate model or easily fall into the local optimum as the dimension increases. To address these problems, this paper proposes an adaptive surrogate-assisted particle swarm optimization algorithm. In the proposed algorithm, a surrogate model is adaptively selected from a single model and an ensemble model by comparing the best existing solution and the latest obtained solution. Additionally, a model output criterion based on the standard deviation is suggested to improve the stability and generalization ability of the ensemble model. To verify the performance of the proposed algorithm, 10 benchmark functions with different modalities from 10 to 50 dimensions are tested, and the results are compared with those of five state-of-the-art SAEAs. The experimental results indicate that the proposed algorithm performs well for most benchmark functions within a limited number of FEs. Moreover, the performance of the proposed algorithm in solving engineering problems is verified by applying the algorithm to the PX oxidation process.

Keywords Surrogate-assisted evolutionary algorithm · Ensemble model · Radial basis functions · Particle swarm optimization

1 Introduction

In recent years, due to the existence of complex nonlinearities in actual engineering processes, commonly used optimization algorithms (such as gradient descent) easily fall into a local optimum. To solve this problem, evolutionary algorithms have emerged, and have achieved great success in practical industrial problems. For instance, Zhu et al. (2019) employed adaptive particle swarm optimization (PSO) and the genetic algorithm to different constrained engineering design problems, and the results demonstrated the superiority of evolutionary algorithms for application in the engineering process. Mohamed (2017) applied differential evolution to engineering optimization problems and obtained efficient and robust solutions. However, in complex practical problems, function evaluations (FEs) often involve costly numerical simulations or expensive experiments, which is a great challenge for the promotion of evolutionary algorithms. To address this problem, many researchers have developed various surrogate-assisted evolutionary algorithms (SAEAs). The principle of SAEAs is to replace some of the real FEs in the evolution process with a surrogate model. The computational cost of constructing a surrogate model to approximate the fitness for candidate solutions is much lower than that of conducting real FEs.

The surrogate model is an analytical model (Alizadeh et al. 2020) that uses a limited number of samples and a given model structure to fit a model expression that approximates the input–output relationship. Common surrogate models used in SAEAs include radial basis functions (RBFs) (Liu et al. 2016; Sun et al. 2017), polynomial regression (Wu et al. 2018; Si et al. 2011), the Kriging model (Fu et al. 2020; Huang et al. 2018), artificial neural

✉ Shaojun Li
lishaojun@ecust.edu.cn

[1] Key Laboratory of Smart Manufacturing in Energy Chemical Processes, East China University of Science and Technology, Ministry of Education, Shanghai 200237, China

networks (Pan et al. 2019), support vector machine (SVM) (Zhai et al. 2019), and ensemble surrogate models (Wang et al. 2017; Li et al. 2019). Many experiments have been conducted to compare the performance of different surrogate models (Díaz-Manríquez et al. 2017; Jia et al. 2020), and the compared results have shown that the RBF models have high global and local approximation capabilities.

The process in which individuals are selected for evaluation using the real fitness function and then used to update the surrogate model is called the "infill criterion" or "model management" (Urquhart et al. 2020; Huang et al. 2018). Model management plays an important role in SAEAs. A variety of research on this process has been reported, and the most direct criterion is to evaluate those individuals that have the best fitness or high approximation accuracy based on the results obtained from the surrogate model (Fan et al. 2020; Zhang et al. 2015; Regis 2014); these solutions may improve the accuracy of the surrogate model in promising regions of search space (Wang et al. 2017). In addition, Liu et al. (2017) and Hüllen et al. (2020) posited that solutions with high uncertainty may be good candidates for evaluation, as the exact evaluation of these solutions can push the search to unexplored or not-well-explored regions (Li et al. 2020; Jin 2011). To further improve the accuracy of the surrogate model and obtain higher-quality optimal solutions, both exploration and exploitation should be simultaneously considered (Pan et al. 2021), which is also referred to as the balance between global and local model management. For example, Liu et al. (2018) employed derivative-based optimization to find a few local optimal solutions, and then made an SAEA work harmonically from the not-well-distributed local optimization results, which ultimately achieved a global optimization effect. Wang et al. (2017) proposed a global model management method inspired by committee-based active learning, which searches for the best and most uncertain solutions for updating the global surrogate model. Simultaneously, a local surrogate model is constructed around the currently available optimal solution. When the evolutionary search using the global model management strategy is not further improved, the entire evolutionary search is switched to the local search, and vice versa. Sun et al. (2015) constructed a global surrogate model and multiple local surrogate models for approximating the fitness of the population. The global model guides the search for the optimal solution, and the local surrogate model improves the accuracy of fitness estimation.

While the methods mentioned previously can achieve good results, with the increase of the dimension, some of these methods easily fall into the local optimum or continue to switch between global and local searches. This indicates that the existing relevant approaches cannot successfully find an optimal solution and also waste computational resources. In addition, the solution obtained via an SAEA defaults to the current optimal solution and is not compared with the best solution that already exists. The reliability of this operation largely depends on the surrogate model. Various strategies have been reported to improve the effectiveness of the surrogate models in SAEAs to the greatest extent. For instance, Liang and Rasheed (2008) proposed an adaptive surrogate-assisted genetic algorithm that adaptively selects the type of surrogate model and adjusts the complexity and frequency of use of the model as the search process proceeds. Nguyen and Long (2021) proposed the use of adaptive control to determine the effective proportion of evaluations conducted by the real fitness function and the surrogate model during the search process, which overcomes the imbalance between exploitation and exploration resulting from the use of a fixed proportion of the two evaluation methods. Loshchilov et al. (2012) used covariance matrix adaptation evolutionary strategies (CMA-ES) as the optimization algorithm and an SVM as the surrogate model and proposed an algorithm that adaptively adjusts the life length of the surrogate model (the number of CMA-ES generations before updating the surrogate model) and the hyper-parameters of the surrogate model. These methods adaptively adjust the type and frequency of surrogate models based on the optimization results during the search process, which helps to improve the quality of the surrogate models and obtain better solutions while reducing the computational cost.

Inspired by the idea of adaptive methods, and to avoid the invalid calculation caused by the insufficient prediction accuracy of the surrogate model to the greatest extent, an adaptive surrogate-assisted PSO (ASAPSO) algorithm is proposed in this study and takes advantage of the ability of ensemble learning to improve the prediction accuracy of the ensemble model (Ye et al. 2018). In the ASAPSO algorithm, a strategy of adaptively selecting a surrogate between an ensemble model based on RBF models and a single RBF model is executed to assist the PSO algorithm to search for an optimal solution. The strategy relies on the comparison of the solution obtained by the surrogate-assisted PSO algorithm and the best solution in the current training data set. Additionally, considering the limitation of samples and the randomness of sampling, rather than simply calculating the average of the sub-models, the influence of abnormal conditions in the process of modeling is taken into account when calculating the output of the ensemble model. The output criterion is used to enhance the stability and generalization of the ensemble model.

The remainder of this manuscript is organized as follows. Section 2 provides a brief overview of the related techniques. The proposed ASAPSO algorithm is elaborated

in Sect. 3. Section 4 introduces the comparative experimental studies on benchmark functions. In Sect. 5, the proposed algorithm is applied to a chemical engineering problem. Finally, Sect. 6 summarizes this paper.

2 Related techniques

2.1 RBF model

The basic concept of constructing an RBF model is to determine a set of reliable sample points $x_i(i = 1, 2, \cdots, n)$. The radial distance $\phi_i(i = 1, 2, \cdots, n)$ between variable x and each sample point $x_i(i = 1, 2, \cdots, n)$ can be expressed as follows:

$$\phi_i = \|x - x_i\|(i = 1, 2, \cdots, n) \tag{1}$$

where $\|x - x_i\|$ represents the Euclidean distance from unknown point x to sample point x_i. The original complex high-dimensional problem is transformed into a simple low-dimensional problem by introducing the radial distance. The values at unknown sample points are calculated by the linear superposition of the basis function φ_i. The RBF model is typically used to construct the functional approximation of the following expression:

$$\hat{f}(x) = \sum_{i=1}^{n} \beta_i \cdot \varphi_i(\|x - x_i\|) \tag{2}$$

where $\beta = [\beta_1, \beta_2, \cdots, \beta_n]^T$ is the weight coefficient. The use of Eq. (2) as the surrogate model should meet the following condition:

$$\hat{f}(x_i) \approx f(x_i) \tag{3}$$

where $\hat{f}(x_i)$ and $f(x_i)$,respectively, refer to the predicted and actual values of the sample point x_i. The substitution of Eq. (3) into Eq. (2) yields the following:

$$f = \beta^T \Phi \tag{4}$$

$$\Phi = \begin{bmatrix} \varphi(\|x_1 - x_1\|), & \cdots, & \varphi(\|x_n - x_1\|) \\ \varphi(\|x_1 - x_2\|), & \cdots, & \varphi(\|x_n - x_2\|) \\ & \vdots & \\ \varphi(\|x_1 - x_n\|), & \cdots, & \varphi(\|x_n - x_n\|) \end{bmatrix}, \qquad f = [f(x_1), f(x_2), \cdots, f(x_n)]^T.$$

The coefficient β can be obtained by Eq. (5).

$$\beta = \Phi^{-1} f \tag{5}$$

In the process of building the RBF model, the reasonable selection of the radial function has a great influence on the accuracy of the model. As shown in Table 1, the common radial functions are the Gaussian, multiple quadratic (MQ), inverse quadratic, thin-plate spline, and cubic spline functions. As presented in the table, ε is a constant greater than 0, and is used to control the scope of the radial function.

Table 1 The common used radial functions

Radial function	Expression ($r = \|x - x_i\|$)
Gaussian	$\varphi(r) = e^{-\varepsilon r^2}$
Multiple quadratic(MQ)	$\varphi(r) = \sqrt{\varepsilon^2 + r^2}$
Inverse quadratic	$\varphi(r) = \frac{1}{\sqrt{\varepsilon^2 + r^2}}$
Thin-plate spline	$\varphi(r) = r^2 \log(\varepsilon r^2 + 1)$
Cubic spline	$\varphi(r) = (r + \varepsilon)^2$

Among these radial functions, the MQ and Gaussian functions are the most widely used. The MQ function has the advantages of high calculation efficiency, good model fitting accuracy, and excellent stability. It is usually selected when the RBF is used for interpolation (Gao et al. 2020). The Gaussian function is selected when using a three-layer network to realize RBF approximation (Wang et al. 2019). The MQ function was selected for use in the present work.

2.2 Particle swarm optimization algorithm

The PSO algorithm searches for the optimal solution to the problem by simulating the behavioral characteristics of biological populations, such as birds and fish. It has been widely used in the field of engineering optimization design due to its simple algorithm and strong optimization ability (Sun et al. 2015). The algorithm starts by randomly locating a swarm of particles in the search space, each of which has its own position and velocity. During each iteration, the velocity and position of the particle are, respectively, updated by Eqs. (6) and (7):

$$v_i(t+1) = w(t) \cdot v_i(t) + c_1 r_1 (P_i(t) - x_i(t)) + c_2 r_2 (P_g(t) - x_i(t)) \tag{6}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{7}$$

where $v_i(t) = (v_{i1}(t), v_{i2}(t), \cdots, v_{iD}(t))$ and $x_i(t) = (x_{i1}(t), x_{i2}(t), \cdots, x_{iD}(t))$ are, respectively, the velocity and position of particle i at the t-th iteration, $P_i(t) = (p_{i1}(t), p_{i2}(t), \cdots, p_{iD}(t))$ is the best historical location for the discovery of particle i (called the individual best), $P_g(t) = (p_{g1}(t), p_{g2}(t), \cdots, p_{gD}(t))$ is the best historical location for the swarm (called the global best), r_1 and r_2 are two random numbers generated uniformly within the range [0,1], and c_1 and c_2 are positive constants called the coefficients of acceleration, which are usually set to 1.5.

Finally, $w(t)$ is the inertia weight factor, which is updated according to Eq. (8) as the number of iterations changes:

$$w(t) = 0.9 - 0.5 * \frac{t}{t_{\max}} \tag{8}$$

where t_{max} is the maximum number of iterations.

3 The proposed ASAPSO algorithm

In some existing SAEAs, the evolutionary search process easily falls into a local optimum due to the lack of accuracy of the surrogate model or the defects of the evolutionary algorithms. As a consequence, the SAEA cannot obtain a satisfactory optimal solution. To solve this problem, the ASAPSO algorithm is proposed in this paper. The main contributions of the proposed algorithm include the following two aspects.

(1) An adaptive surrogate selection method is suggested, which depends on the comparison between the solution obtained via the surrogate-assisted PSO algorithm and the best solution in the current training data set. This method is used to ensure the effectiveness of the optimization and to reduce the computational cost while simultaneously improving the quality of the optimal solution.

The prediction accuracy of the surrogate model will largely determine the search direction of the optimal solution. A single RBF model has low prediction accuracy when dealing with complex problems, which tends to result in premature convergence and invalid computations. While an ensemble model has high prediction accuracy in most cases, considering that it applies the bootstrap method to randomly select subsets for building sub-models, the corresponding sub-models may be over-fitted when the number of samples in the subsets is too small, which will lead to a decrease in the prediction accuracy of the ensemble model.

The comparison between the obtained optimal solution and the best solution in the current training data set is used to determine whether the currently selected surrogate model can assist the PSO algorithm in searching for a better solution. If a better solution is obtained, it indicates that the solution in the current training data set is converging toward the real optimal solution, and the selection of a single RBF model as the surrogate will help to exploit the region around the real optimal solution while reducing the time required to construct an ensemble model. If a better solution is not obtained, it indicates that the prediction accuracy of the current surrogate model is inadequate and misleads the PSO algorithm in the wrong direction, and the selection of an ensemble model as the surrogate will help to improve the quality of the optimal solution.

(2) An ensemble model output criterion is suggested. Its purpose is to enhance the stability and generalization ability of the ensemble model.

Considering the limitations of the data in practical engineering problems and the randomness of bootstrap sampling, the T sub-models, respectively, established by using different subsets have a certain degree of correlation. To reduce the influence of this correlation, the output of the ensemble model is obtained by reasonably analyzing and processing the output of the T sub-models. Two statistics, the average, and median, are used to obtain the output of the ensemble model. The average is susceptible to outliers; thus, when there are no outliers among the T predicted values, the generalization ability of the ensemble model can be improved by averaging over T sub-models. Additionally, the median is not easily affected by outliers; thus, when there are outliers among the T predicted values, the median of the T sub-models can be chosen to eliminate the influence of outliers on the ensemble model. Based on the preceding analysis, an approach based on the standard deviation is suggested in this paper. The calculation of the ensemble model output includes two steps.

Step 1 Calculate the standard deviation (σ) of all predicted values of the T sub-models via Eq. (9), where T is the number of sub-models and $\overline{Y}$ is the average of the T predicted values.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{T} (Y_i - \overline{Y})}{T}} \tag{9}$$

Step 2 If all the predicted values of the T sub-models are within the range $[\overline{Y} - 3\sigma, \overline{Y} + 3\sigma]$, the output of the ensemble model is equal to the average of the T sub-models; otherwise, the median of the T sub-models is used as the output of the ensemble model.

Consider a minimization of the function as an example. The input variable is the decision variable X with dimension D. The output variable is the function value of X. The maximum number of real FEs (FE_{max}) is used as the termination condition of the ASAPSO algorithm. The detailed steps of ASAPSO are as follows.

Step 1 M samples $X = [x_1, x_2, \cdots, x_M]^T$ are taken from the definition domain of the function by using Latin

hypercube sampling (LHS) (Yondo et al. 2018). j is the number of real FEs; set $j = 0$.

Step 2 Calculate the real fitness of X, obtain $Y = [Y_1, Y_2, \cdots, Y_M]^T$, and construct the initial training data set $Dt = [X,Y]$. Additionally, $j = M$, $Y^j_{best} = minimize(Y)$, and $Y_{gbest} = Y^j_{best}$.

Step 3 If j is less than FE_{max}, go to step 4; otherwise, go to step 10.

Step 4 If Y_{gbest} is less than or equal to Y^j_{best} , go to step 5; otherwise, go to step 7.

Step 5 Construct the RBF model using Dt.

Step 6 Obtain the optimal solution, *gbest*, of the RBF model by using the PSO algorithm, and go to step 9.

Step 7 Construct an ensemble model using Dt.

Step 8 Obtain the optimal solution, *gbest*, of the ensemble model by using the PSO algorithm.

Step 9 Evaluate *gbest* via a real fitness function, obtain Y_{gbest}, and add ($gbest$,Y_{gbest}) to Dt. Additionally, set $j = j + 1$ and $Y^j_{best} = minimize(Y)$, and return to step3.

Step 10 Output the solution with Y^j_{best} in Dt.

To provide a more intuitive understanding of the proposed ASAPSO algorithm, a flowchart is presented in Fig. 1. The pseudo-code of the ASAPSO algorithm is presented in Algorithm 1.

Algorithm 1 The ASAPSO algorithm

1. Initialize X with M samples using LHS, set $j = 0$.
2. Calculate the real fitness value (Y) of X, set $j = M$.
3. Construct the training data set $Dt = [X,Y]$, $Y^j_{best} = \text{minimize}(Y)$, $Y_{gbest} = Y^j_{best}$.
4. **while** $j < FE_{max}$ **do**
5. **if** $Y_{gbest} \le Y^j_{best}$
6. Construct the RBF model using Dt.
7. Obtain the optimal solution (*gbest*) of the RBF model by using PSO.
8. **else**
9. Construct the ensemble model using Dt.
10. Obtain the optimal solution (*gbest*) of the ensemble model by using PSO.
11. **end if**
12. Calculate the real fitness value (Y_{gbest}) of *gbest*, and then add (*gbest*, Y_{gbest}) to Dt. Set $j = j + 1$, $Y^j_{best} = \text{minimize}(Y)$.
13. **end while**
14. Output the solution with Y^j_{best}.

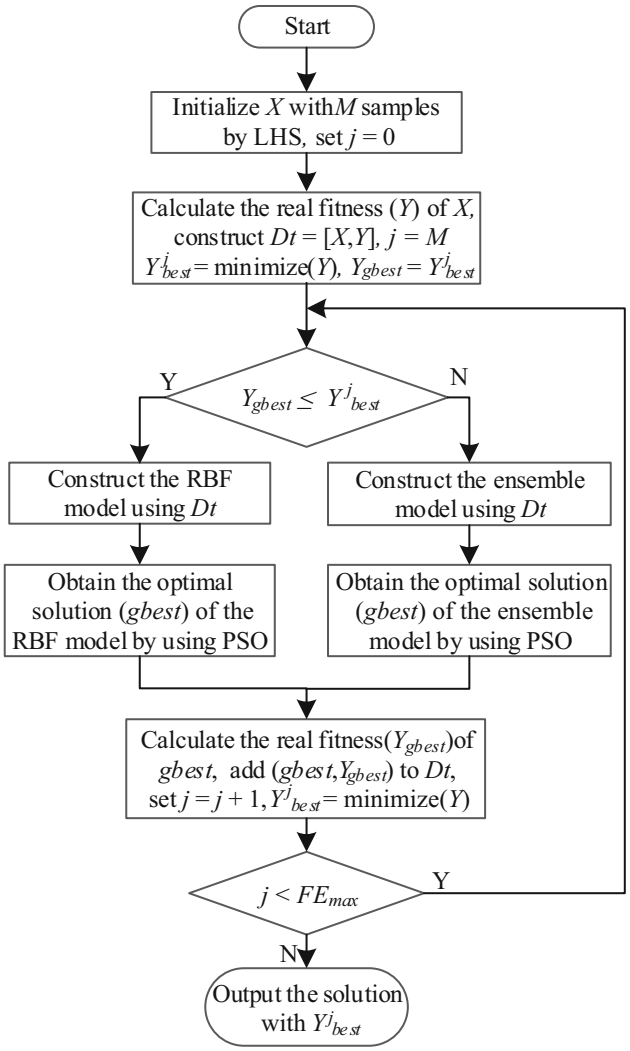

Fig. 1 The flowchart of the ASAPSO algorithm

4 Experimental studies

In the experiments, all algorithms under comparison began with 5D (where D is the dimension of the problem) exact FEs and were terminated after 11D exact FEs were exhausted. The parameters of the PSO algorithm were the same in all algorithms under comparison, the size of the population was 100, and the maximum number of iterations was 100. If the ensemble scale is too large, the calculation cost will grow linearly with the increase of the ensemble size (Wang et al. 2019), which will reduce the computational efficiency; thus, $T = 20$ was selected in all experiments. For bootstrap sampling, the probability of each sample in Dt being selected is 0.5 (Wang et al. 2019), which guarantees the diversity of the ensemble in principle. Information on the common benchmark functions is presented in Table 2.

To judge whether there is statistical significance between the ASAPSO algorithm and other algorithms, the Wilcoxon rank-sum test (Masato et al. 2020) at a significance level of $\alpha = 0.05$ was conducted. In Tables 3, 4, 5, "≈" indicates that there is no statistically significant difference between the results obtained by the ASAPSO algorithm and the compared algorithms, " + " indicates that the ASAPSO algorithm is significantly better than the compared algorithms, and " − " indicates that the ASAPSO algorithm is significantly outperformed by the compared algorithm. The last row of Tables 3, 4, 5 summarizes the results of the Wilcoxon rank-sum test as w/t/l (win/tie/lose).

4.1 Effects of the model-based criterion

To demonstrate the effects of the model-based criterion on the proposed ASAPSO algorithm, the ASAPSO algorithm was compared with two of its variants (ASAPSO-S and ASAPSO-E). ASAPSO-S is similar to ASAPSO except that it selects only a single RBF to assist the PSO algorithm. Moreover, ASAPSO-E is similar to ASAPSO except that it selects an ensemble model to assist the PSO algorithm. Ten benchmark functions with 10-D and 20-D were used to compare the three algorithms. The average best fitness, standard deviation, and running time of the three algorithms over 30 independent runs are reported in Table 3, and the individual instances with best fitness are highlighted.

The average ranks of the three algorithms reported in Table 3 demonstrate that the proposed ASAPSO algorithm is the best of the three algorithms. However, according to the results of the Wilcoxon rank-sum test, ASAPSO-E outperformed ASAPSO on the 10-D Quartic problem and performed similarly to ASAPSO on the 20-D Quartic problem. The reason for this may be that the Quartic problem has a small search space, and an ensemble model with high prediction accuracy can be obtained using a limited number of samples, which is conducive to obtaining higher-quality optimal solutions. The ASAPSO-E algorithm outperformed the ASAPSO algorithm on the 10-D Sphere, Griewank, and Schwefel2.21 problems. Additionally, ASAPSO-S slightly outperformed ASAPSO on the 10-D Ellipsoid problem and achieved similar performance on the 10-D Rosenbrock problem. These findings indicate that the ASAPSO algorithm is not suitable for solving low-dimensional problems with large design spaces. The reason for this may be that it cannot search the entire design space in detail within a limited number of FEs when the dimension is low. With the increase of the dimension, the number of real fitness evaluations increases; consequently, increasingly more optimal solutions are added to the training data set, and the prediction accuracy

Table 2 Information on the common benchmark functions

Function	Problem	Decision space	Optimum	Characteristics
F1	Ellipsoid	$[-5.12,5.12]^D$	0	Uni-modal
F2	Sum square	$[-10,10]^D$	0	Uni-modal
F3	Sphere	$[-100,100]^D$	0	Uni-modal
F4	Griewank	$[-600,600]^D$	0	Multi-modal
F5	Schwefel2.21	$[-100,100]^D$	0	Multi-modal
F6	Alpine	$[-10,10]^D$	0	Multi-modal
F7	Quartic	$[-1.28,1.28]^D$	0	Multi-modal
F8	Ackley	$[-32,32]^D$	0	Multi-modal
F9	Rastrigin	$[-5,5]^D$	0	Very complicated multi-modal
F10	Rosenbrock	$[-2.048,2.048]^D$	0	Multi-modal with narrow valley

of the surrogate model continually increases. The convergence profiles of the algorithms of the 10-D and 20-D Rastrigin problems are shown in Fig. 2. It can be concluded from the figure that in the ASAPSO-S algorithm, due to the low quality of the single RBF model, the search direction of the optimal solution may be easily misled, which leads to the problem of premature convergence. Moreover, due to the higher prediction accuracy of the ensemble model in the ASAPSO-E algorithm, better optimal solutions, and faster convergence were obtained. The ASAPSO algorithm can quickly mitigate the impact of incorrect optimal solutions on the search direction of the optimal solution by adaptively selecting a surrogate model, which improves the quality of the optimal solution while ensuring the convergence speed. Additionally, by comparing the running times of the three algorithms reported in Table 3, it is evident that the adaptive selection of the surrogate model can significantly reduce the computational cost while ensuring the quality of the optimal solution. Overall, the effectiveness of the model-based criterion on the ASAPSO algorithm was demonstrated.

4.2 Low-dimensional problems

To verify the performance of the proposed ASAPSO algorithm in low dimensions, it was applied to 10-D, 20-D, and 30-D problems, and its performance was compared with those of several existing algorithms, namely CALSAPSO (Wang et al. 2017), WTA1 (Goel et al. 2007), and MAES-ExI (Emmerich et al. 2006), the main characteristics of which are described as follows.

(1) CALSAPSO is an ensemble-assisted evolutionary algorithm aided by multiple surrogates, namely PR, RBF, and Kriging models, and an active learning-based surrogate management strategy is employed.
(2) WTA1 is an ensemble-based SAEA with weights assigned by the root-mean-square errors of the PR, RBF, and Kriging models. In the experiment, WTA1 continued to evaluate the optimal solution of the surrogate ensemble and added it to the training data set for model updating.
(3) MAES-ExI is a Kriging-based SAEA with expected improvement (ExI) as its criterion for selecting solutions to evaluate.

To prevent randomness, each algorithm was independently run 30 times in the experiment. The average best fitness and standard deviation obtained by the four compared algorithms are reported in Table 4, and the best average fitness of the four algorithms for each problem is indicated in bold.

As revealed by the statistical results presented in Table 4, when the termination condition was satisfied, the performance ranking of the four algorithms was as follows: ASAPSO was the best, CALSAPSO was the second-best, WTA1 was the third-best, and MAES-ExI was the worst. The performance of ASAPSO was less competitive than those of the CALSAPSO and WTA1 algorithms on a portion of 10-D problems. The reason for this may be that the CALSAPSO and WTA1 algorithms are both ensemble surrogate-assisted PSO algorithms, and the ensemble surrogate has high prediction accuracy. For the ASAPSO algorithm, only 60 new solutions were evaluated with the real fitness function throughout the optimization process, and most of the new solutions could be obtained by using the RBF-assisted PSO algorithm; thus the ensemble model did not work significantly well.

On the 30-D Rastrigin and Rosenbrock problems, the performance of the CALSAPSO algorithm was similar to that of the ASAPSO algorithm. The reason for this is that the combination of global and local search in the CALSAPSO algorithm is a useful way to locate the optimal solution region when dealing with complex problems. The CALSAPSO algorithm outperformed the ASAPSO algorithm on the Sphere and Griewank problems, possibly because they have a large search space, and the sparse location of the search space cannot be fully explored within a limited number of FEs. In addition, for most functions,

Table 3 Average best fitness, standard deviation and running time of the three algorithms

Function	D	ASAPSO			ASAPSO-S				ASAPSO-E			
		Avg	Std	Time (s)	Avg		Std	Time (s)	Avg		Std	Time (s)
F1	10	1.42E-02	2.05E-02	6.77E+01	**3.60E-03**	+	7.30E-03	4.45E+00	1.55E+00	−	5.63E-01	3.25E+02
F1	20	**3.83E-02**	2.25E-02	6.82E+02	2.91E-01	−	2.41E-01	1.71E+01	9.28E+00	−	2.49E+00	2.45E+03
F2	10	**2.75E-01**	2.79E-01	7.82E+01	4.74E-01	≈	7.33E-01	2.56E+00	7.60E+00	−	3.95E+00	3.06E+02
F2	20	**4.95E-01**	4.07E-01	4.44E+02	6.11E+00	−	4.23E+00	9.22E+00	4.06E+01	−	8.34E+00	1.62E+03
F3	10	1.38E+02	7.89E+01	4.74E+01	2.46E+02	−	1.67E+02	2.47E+00	**9.49E+01**	+	3.95E+01	3.22E+02
F3	20	**9.61E+01**	3.59E+01	6.04E+02	7.27E+02	−	3.49E+02	8.99E+00	2.98E+02	−	6.89E+01	1.57E+03
F4	10	5.00E+00	3.87E+00	1.39E+02	7.88E+00	−	3.10E+00	4.66E+00	**1.79E+00**	+	3.85E-01	3.31E+02
F4	20	**1.76E+00**	2.32E-01	1.02E+03	1.32E+01	−	4.84E+00	1.67E+01	3.58E+00	−	6.28E-01	2.26E+03
F5	10	1.47E+01	9.78E+00	9.28E+01	4.41E+01	−	1.09E+01	2.41E+00	**6.63E+00**	+	1.86E+00	3.08E+02
F5	20	**4.96E+00**	1.58E+00	4.49E+02	5.63E+01	−	9.85E+00	8.91E+00	5.03E+00	−	1.08E+00	1.45E+03
F6	10	**2.98E-01**	5.08E-01	9.63E+01	5.13E+00	−	3.76E+00	5.60E+00	2.85E+00	−	1.41E+00	3.96E+02
F6	20	**6.51E-02**	1.58E-01	7.58E+02	6.94E+00	−	3.23E+00	1.67E+01	1.35E+00	−	3.75E-01	1.99E+03
F7	10	5.79E-02	3.89E-02	2.01E+02	1.82E-01	−	1.07E-01	5.14E+00	**4.00E-02**	+	3.28E-02	3.88E+02
F7	20	1.21E-01	5.14E-02	1.04E+03	2.27E-01	−	1.07E-01	1.44E+01	**1.12E-01**	≈	6.57E-02	2.07E+03
F8	10	**4.26E+00**	1.16E+00	1.22E+02	1.49E+01	−	3.94E+00	4.90E+00	4.58E+00	≈	9.22E-01	3.27E+02
F8	20	**4.20E+00**	9.25E-01	1.05E+03	1.51E+01	−	2.08E+00	1.69E+01	4.47E+00	≈	5.87E-01	2.34E+03
F9	10	**3.59E+01**	1.47E+01	1.65E+02	6.88E+01	−	1.70E+01	2.39E+00	4.11E+01	≈	1.24E+01	3.18E+02
F9	20	**6.86E+01**	2.78E+01	7.50E+02	9.67E+01	−	3.81E+01	9.27E+00	9.93E+01	−	2.44E+01	1.62E+03
F10	10	1.15E+01	2.29E+00	1.64E+02	**1.07E+01**	≈	3.18E+00	5.66E+00	2.17E+01	−	4.94E+00	4.11E+02
F10	20	**2.32E+01**	2.05E+00	1.16E+03	2.60E+01	−	6.84E+00	1.70E+01	5.21E+01	−	6.34E+00	2.65E+03
Average rank		1.35			2.55				2.1			
w/t/l					1/2/17				4/4/12			

The bold values are the best fitness of the three algorithms for each problem

Table 4 Average best fitness and standard deviation of the four compared algorithms

Function	D	ASAPSO		CALSAPSO			MAES-ExI			WTA1		
		Avg	Std	Avg		Std	Avg		Std	Avg		Std
F1	10	1.42E-02	2.05E-02	6.93E-01	−	6.93E-01	1.95E+01	−	8.75E+00	**8.90E-03**	+	**1.28E-02**
F1	20	**3.83E-02**	**2.25E-02**	4.72E+00	−	2.13E+00	5.78E+02	−	1.28E+02	5.27E+00	−	3.88E+00
F1	30	**9.96E-01**	**3.32E-01**	4.53E+00	−	2.45E+00	1.91E+03	−	2.69E+02	6.43E+01	−	2.02E+01
F2	10	2.75E-01	2.79E-01	1.63E+00	−	1.52E+00	7.96E+01	−	3.75E+01	**1.43E-01**	+	**1.61E-01**
F2	20	**4.95E-01**	**4.07E-01**	1.63E+01	−	8.56E+00	2.25E+03	−	5.10E+02	1.84E+01	−	1.08E+01
F2	30	**8.19E+00**	**3.51E+00**	2.12E+01	−	1.74E+01	7.22E+03	−	1.19E+03	2.74E+02	−	5.57E+01
F3	10	1.38E+02	7.89E+01	**2.73E+00**	+	**4.90E+00**	1.16E+03	−	5.92E+02	2.97E+01	+	1.50E+01
F3	20	9.61E+01	**3.59E+01**	**6.84E+01**	≈	3.68E+01	2.65E+04	−	3.94E+03	1.95E+02	−	8.69E+01
F3	30	2.61E+02	4.22E+01	**7.15E+01**	+	**3.74E+01**	5.15E+04	−	8.48E+03	1.41E+03	−	4.92E+02
F4	10	5.00E+00	3.87E+00	**1.35E+00**	+	**1.77E-01**	1.37E+01	−	5.95E+00	1.83E+00	+	5.50E-01
F4	20	1.76E+00	2.32E-01	**1.54E+00**	+	**3.13E-01**	2.39E+02	−	4.03E+01	6.68E+00	−	2.63E+00
F4	30	3.26E+00	4.02E-01	**1.75E+00**	+	**2.93E-0**1	4.50E+02	−	6.88E+01	1.52E+01	−	4.45E+00
F5	10	**1.47E+01**	9.78E+00	5.26E+01	−	1.06E+01	1.85E+01	≈	**7.53E+00**	2.43E+01	−	8.46E+00
F5	20	**4.96E+00**	**1.58E+00**	6.96E+01	−	6.55E+00	5.19E+01	−	5.59E+00	4.29E+01	−	6.50E+00
F5	30	**3.11E+00**	**6.25E-01**	7.03E+01	−	6.68E+00	6.52E+01	−	8.32E+00	4.76E+01	−	9.22E+00
F6	10	**2.98E-01**	**5.08E-01**	7.54E+00	−	2.59E+00	8.59E+00	−	2.46E+00	8.33E+00	−	3.52E+00
F6	20	**6.51E-02**	**1.58E-01**	1.57E+01	−	5.38E+00	2.11E+01	−	4.60E+00	5.99E+00	−	3.98E+00
F6	30	**1.74E-01**	**4.42E-01**	1.55E+01	−	5.25E+00	4.31E+01	−	5.59E+00	8.27E+00	−	2.73E+00
F7	10	**5.79E-02**	**3.89E-02**	1.25E-01	−	7.81E-02	5.90E-01	−	2.73E-01	4.74E-01	−	3.30E-01
F7	20	**1.21E-01**	**5.14E-02**	1.87E-01	≈	1.44E-01	1.75E+01	−	4.43E+00	5.50E-01	−	3.54E-01
F7	30	2.66E-01	**7.68E-02**	**2.07E-01**	+	1.35E-01	6.25E+01	−	1.72E+01	1.30E+00	−	8.33E-01
F8	10	**4.26E+00**	1.16E+00	1.83E+01	−	**6.91E-01**	8.70E+00	−	4.93E+00	1.39E+01	−	4.44E+00
F8	20	**4.20E+00**	**9.25E-01**	1.85E+01	−	1.82E+00	1.88E+01	−	1.42E+00	1.30E+01	−	3.07E+00
F8	30	**2.99E+00**	4.73E-01	1.56E+01	−	3.07E+00	2.02E+01	−	**1.96E-01**	9.17E+00	−	4.14E+00
F9	10	**3.59E+01**	1.47E+01	7.22E+01	−	1.98E+01	8.20E+01	−	**1.31E+01**	8.52E+01	−	2.07E+01
F9	20	**6.86E+01**	**2.78E+01**	1.24E+02	−	4.16E+01	1.96E+02	−	1.83E+01	1.31E+02	−	3.27E+01
F9	30	1.64E+02	3.00E+01	**1.54E+02**	≈	6.40E+01	3.37E+02	−	**2.01E+01**	1.75E+02	≈	2.37E+01
F10	10	**1.15E+01**	**2.29E+00**	3.37E+01	−	1.40E+01	1.20E+02	−	4.62E+01	4.70E+01	−	1.75E+01
F10	20	**2.32E+01**	**2.05E+00**	4.94E+01	−	1.26E+01	1.87E+03	−	4.69E+02	1.29E+02	−	2.91E+01
F10	30	**5.68E+01**	**1.01E+01**	5.90E+01	≈	1.07E+01	4.41E+03	−	6.44E+02	2.45E+02	−	6.02E+01
Average rank		1.4		2.2			3.77			2.63		
w/t/l				6/4/20			0/1/29			4/1/25		

The bold values are the best fitness and standard deviation of the four algorithms for each problem

Table 5 Average best fitness and standard deviation of three compared algorithms on 50-D

Function	D	ASAPSO		iDEaSm			GPEME		
		Avg	Std	Avg		Std	Avg		Std
F1	50	4.11E+01	5.19E+00	7.66E+01	–	1.09E+01	2.02E+02	–	6.21E+01
F4	50	8.22E+00	6.88E-01	8.79E+01	–	3.41E+01	3.05E+02	–	5.58E+01
F8	50	4.87E+00	3.47E-01	1.77E+01	–	5.54E+00	1.98E+01	–	8.92E-01
F9	50	3.76E+02	1.53E+01	4.77E+02	–	3.33E+01	4.86E+02	–	3.67E+01
F10	50	2.10E+02	1.49E+01	6.37E+03	–	2.97E+02	2.03E+04	–	2.15E+03
Average rank		1		2			3		
w/t/l				0/0/5			0/0/5		

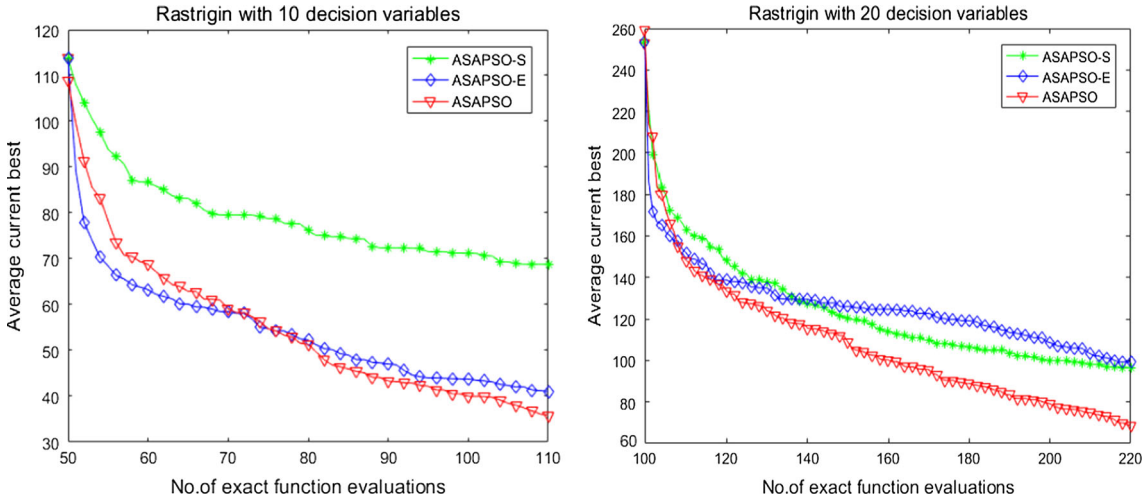

Fig. 2 The convergence profiles of the algorithms of the 10-D and 20-D Rastrigin problems

the ASAPSO algorithm obtained better optimal solutions with the increase of the dimension. The reason for this may be that the 11D−5D = 6D solutions evaluated with the real fitness function were added to *Dt* throughout the search process. As the dimension increases, the number of samples in *Dt* also increases, which improves the prediction accuracy of the surrogate model and contributes to the quality of the optimal solution.

To obtain a more comprehensive understanding of the performance of the ASAPSO algorithm, three representative test functions were analyzed, and the convergence profiles of three representative benchmark functions on $D = 10, 20$ are presented in Fig. 3. The Alpine function is a classical multi-modal minimization test function. When it tends to infinity in the definition domain, the function will produce a large number of differentiable local extrema along the direction of the independent variable, which is very difficult to optimize. The function is used to detect the optimization ability of the algorithm. From Fig. 3a and b, it is evident that ASAPSO exhibited a strong optimization ability; in particular, on the 20-D problem, it converged to the optimal solution when the number of FEs reached half the value of FE_{max}. Ackley is a multi-modal function; with the increase of the dimension, its direction gradient and forward direction are various, so the global convergence rate of the algorithm can be detected by this function. As revealed in Fig. 3c and d, both on the 10-D and 20-D problems, ASAPSO quickly converged to an optimal solution, which was significantly better than those obtained by the other three algorithms. Finally, Ratrigin is a very complicated multi-modal problem that has a large number of local optima; thus, it can be used to detect the abilities of the algorithm to jump out of the local optima and conduct a global search. Moreover, its local optima are regular, and it can be used to check the practicability of the algorithm. According to Fig. 3e and f, both on the 10-D and 20-D problems, ASAPSO found an optimal solution that was significantly better than those of the other three algorithms. This may be due to the premature convergence of the other three algorithms in the face of a large number of local

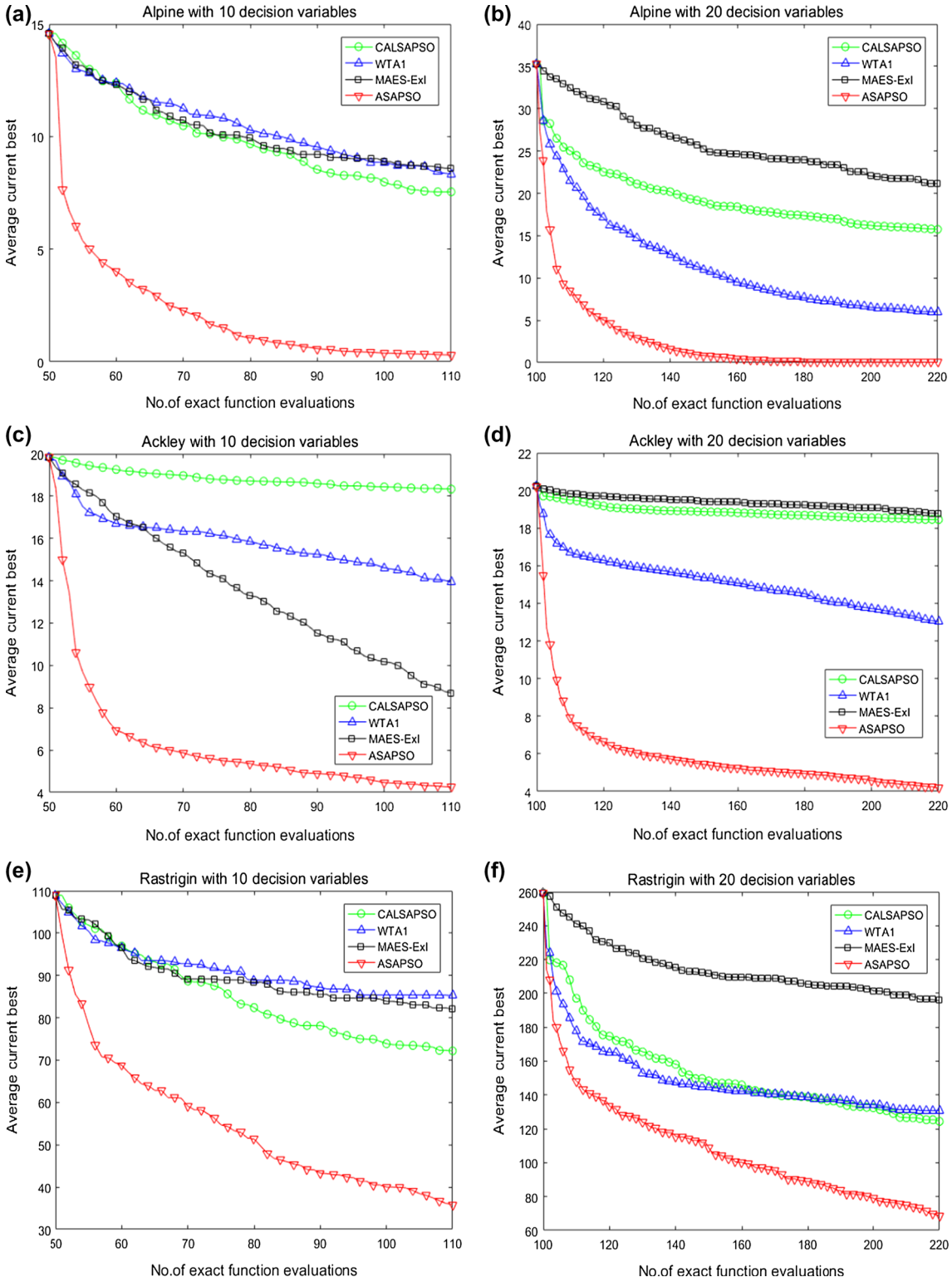

Fig. 3 The convergence profiles of three representative benchmark functions on 10-D, 20-D

extrema, and their inability to conduct a comprehensive search in the global range. Before the termination condition was satisfied, the quality of the optimal solutions obtained by ASAPSO continued to improve, which indicates that ASAPSO has a good global search ability.

Another observation is that the standard deviation of the ASAPSO algorithm was the smallest for most problems,

especially when the average fitness obtained by the ASAPSO algorithm was similar to or better than that of the algorithm being compared; this finding implies that the ASAPSO algorithm has higher stability than the compared algorithms.

4.3 Scalability test

CALSAPSO, WTA1, and MAES-ExI were developed primarily to address low-dimensional problems (Wang et al. 2017). Therefore, the performance of ASAPSO in dealing with high-dimensional problems with dimensions over 30 was assessed via comparison with different algorithms. Of the existing SAEAs, GPEME (Liu et al. 2014) and iDEaSm (Awad et al. 2018) perform well on high-dimensional problems. However, because the codes of the GPEME and iDEaSm algorithms are not available, their experimental results could only be copied from the existing literature (Fan et al. 2020). Thus, to verify the scalability of the ASAPSO algorithm, it was compared with the GPEME and iDEaSm algorithms on five common benchmark functions considered in the existing literature and this paper, and $D = 50$. The main characteristics of the two compared SAEAs are as follows.

(1) GPEME is an online single surrogate-assisted data-driven evolutionary algorithm assisted by a Kriging model, the surrogate management strategy of which is the LCB-based infill sampling criterion.
(2) iDEaSm is an SAEA dedicated to solving high-dimensional problems by optimizing the Kriging correlation parameter θ and using a differential evolution algorithm as the search engine.

The experimental results of the GPEME and iDEaSm algorithms were obtained by independently running each algorithm 25 times. For a fair comparison, the ASAPSO algorithm was also independently run 25 times. The average best fitness and standard deviation of the three compared algorithms on the 50-D problems are reported in Table 5. The best average fitness of the three algorithms for each problem is indicated in bold.

It can be seen from Table 5 that for the five common 50-D benchmark functions when the preset termination condition was met, the performance ranking of the three algorithms was as follows: ASAPSO was the best, iDEaSm was the second-best, and GPEME was the worst. The results of the Wilcoxon rank-sum test at the significance level of $\alpha = 0.05$ reported in the last row of Table 5 demonstrate that the GPEME and iDEaSm algorithms performed significantly worse than the ASAPSO algorithm. The reason for this is that ASAPSO selects the RBF model as the surrogate, which is suitable for overcoming the problem of inadequate samples. Furthermore, the adaptive selection of the single RBF model and the ensemble model can guarantee the quality of the current optimal solution and thus improve the accuracy of the surrogate model. The reason why iDEaSm is more outstanding than GPEME on high-dimensional problems is that it not only improves the Kriging model by optimizing the hyperparameter θ, but also updates the surrogate model with a large number of sample points. As mentioned in the original literature, iDEaSm used 5000 sample points for 50-D problems. However, the original Kriging model was used in GPEME. Overall, the proposed ASAPSO algorithm achieved good performance on high-dimensional problems as compared with the two compared SAEAs.

5 Case study

To verify the performance of the ASAPSO algorithm in dealing with real-world problems, the p-xylene (PX) oxidation process is analyzed. The flowchart of the PX oxidation reaction process is presented in Fig. 4.

Under high temperature and high pressure, terephthalic acid (TA) can be obtained by the oxidation of PX via the catalysis of cobalt acetate, manganese acetate, and bromine ions, and then crystallizes to obtain TA containing the impurities of 4-carboxybenzaldehyde (4-CBA) and p-toluic (PT) acid. In this oxidation process, the combustion losses of acetic acid (HAC) and PX in the reactor are two important economic indices. In addition, the combustion of HAC and PX will produce carbon monoxide (CO), a toxic gas, and carbon dioxide (CO_2), a greenhouse gas (Fan and Yan 2015). Therefore, to obtain greater economic benefits while reducing the impact of CO_2 and CO in the air, it is necessary to reduce the combustion losses of HAC and PX during the reaction process. Via the analysis of the actual working condition data, it was found that the change trends of the combustion losses of HAC and PX were the same. Thus, the reduction and optimization of PX combustion

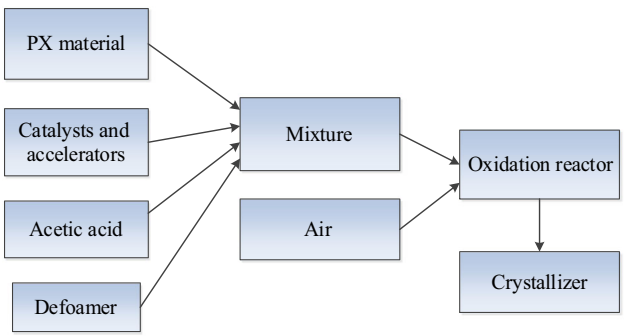

Fig. 4 The flowchart of PX oxidation reaction process

Table 6 The PX combustion loss under actual conditions and after optimization

	Actual conditions	ASAPSO	CALSAPSO	MAES-ExI	WTA1
PX combustion loss	15.13	13.34	13.61	13.59	13.64

loss during the reaction process were carried out by the ASAPSO algorithm as a case study.

The main factors that affect the reaction rate are the reaction temperature, solvent ratio, cobalt (Co) catalyst concentration, manganese (Mn) catalyst concentration, bromine (Br) catalyst concentration, residence time, and gas-phase oxygen concentration. Many research results have indicated that there exists a critical value of oxygen concentration (Liu and Li 2014). When the oxygen concentration is higher than the critical value, there is no effect on the oxidation reaction; only when the oxygen concentration is lower than this critical value will the oxidation reaction be affected. The present study only considers the case of a high gas-phase oxygen concentration, which means that the influence of the gas-phase oxygen concentration is removed. In addition, Co and Mn have the same effect on the reaction rate. Based on this knowledge, the decision variables in this case study were selected as the mass flow of Br (x_1, kg/hr), the mass flow of Co (x_2, kg/hr), the oxidation reactor temperature (x_3, °C), the residence time (x_4, s), the solvent ratio (x_5, mol/kg) and the crystallization temperature (x_6, °C). During the optimization process, depending on the actual working conditions, the upper limit of the decision variables is [300, 200, 210, 5983, 2.78, 187], and the lower limit of the decision variables is [200, 100, 183, 4135, 1.99, 185].

To ensure the quality of the product produced by the PX oxidation process, the optimization was carried out under the condition that the conversion rate of PX to TA was maintained at a high level (99.54% in the actual working conditions). The initial number of sample points was 30. The ASAPSO algorithm was terminated when the number of FEs reached 66. It should be noted that the real FEs were conducted by Aspen Plus 10 software during the entire optimization process. The comparison of PX combustion loss under actual conditions and after optimization by four algorithms is presented in Table 6.

It can be seen from Table 6 that the ASAPSO algorithm obtained the best result among the four optimization algorithms, and it reduced the PX combustion loss from 15.13 to 13.34, which proves that the proposed algorithm could greatly reduce the PX combustion loss in the engineering production process. Furthermore, due to the large scale of factory production and the continuous expansion of new production lines, the demand for raw materials is also further improved by the algorithm. The optimization scheme provides some effective measures for reducing the use of raw materials. Moreover, the contents of the toxic gas CO and the greenhouse gas CO_2 in the factory production process were reduced by the ASAPSO algorithm, which can therefore make some contributions to solving the problem of environmental pollution.

6 Conclusions

To overcome the problems of the existing SAEA methods relying too much on the original samples and falling into a local optimum, as well as the low accuracy of the surrogate model, the adaptive surrogate-assisted PSO (ASAPSO) algorithm was proposed in this paper; this algorithm tries to find a better solution within a limited number of function evaluations (FEs). In the proposed algorithm, an ensemble model based on ensemble learning is used to help the PSO algorithm search for an optimal solution when the currently obtained optimal solution is worse than the best solution that already exists; otherwise, the RBF-assisted PSO algorithm is executed to search for an optimal solution. For the ensemble model, a model output criterion is suggested to reduce the impact of the correlation among base models on the ensemble model output. To verify the good performance of the proposed ASAPSO algorithm, 10 benchmark functions were tested in different dimensions. The experimental results demonstrate that the ASAPSO algorithm found better solutions than five state-of-the-art algorithms for a majority of the problems. Moreover, to prove the effectiveness of the proposed algorithm in solving engineering problems, the algorithm was applied to the PX oxidation process, and satisfactory results were obtained.

Acknowledgments The authors of this paper appreciate the support from the National Natural Science Foundation of China (Project No. 21676086).

Author's contribution Xuemei Li contributed to conceptualization, methodology, software, writing-reviewing and editing. Shaojun Li contributed to conceptualization, writing-reviewing, resources, supervision, project administration and funding acquisition.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Alizadeh R, Allen JK, Mistree F (2020) Managing computational complexity using surrogate models: a critical review. Res Eng Des 31(3):275–298. https://doi.org/10.1007/s00163-020-00336-7

Awad NH, Ali MZ, Mallipeddi R, Suganthan PN (2018) An improved differential evolution algorithm using efficient adapted surrogate model for numerical optimization. Inf Sci 451:326–347. https://doi.org/10.1016/j.ins.2018.04.024

Díaz-Manríquez A, Toscano G, Coello Coello CA (2017) Comparison of metamodeling techniques in evolutionary algorithms. Soft Comput 21(19):5647–5663. https://doi.org/10.1007/s00500-016-2140-z

Emmerich MTM, Giannakoglou KC, Naujoks B (2006) Single- and multi-objective evolutionary optimization assisted by gaussian random field metamodels. IEEE Trans Evol Comput 10(4):421–439. https://doi.org/10.1109/TEVC.2005.859463

Fan CD, Hou B, Xiao LY, Yi LZ (2020) A surrogate-assisted particle swarm optimization using ensemble learning for expensive problems with small sample datasets. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106242

Fan QQ, Yan XF (2015) Differential evolution algorithm with self-adaptive strategy and control parameters for p-xylene oxidation process optimization. Soft Comput 19(5):1363–1391. https://doi.org/10.1007/s00500-014-1349-y

Fu CB, Wang P, Zhao L, Wang XJ (2020) A distance correlation-based Kriging modeling method for high-dimensional problems. Knowl-Based Syst. https://doi.org/10.1016/j.knosys.2020.106356

Gao KF, Mei G, Cuomo S, Piccialli F, Xu NX (2020) ARBF: adaptive radial basis function interpolation algorithm for irregularly scattered point sets. Soft Comput. https://doi.org/10.1007/s00500-020-05211-0

Goel T, Haftka RT, Shyy W, Queipo NV (2007) Ensemble of surrogates. Struct Multidiscip Optim 33(3):199–216. https://doi.org/10.1007/s00158-006-0051-9

Huang CW, Radi B, Hami AE, Bai H (2018) CMA evolution strategy assisted by Kriging model and approximate ranking. Appl Intell 48(11):4288–4304. https://doi.org/10.1007/s10489-018-1193-3

Hüllen G, Zhai JY, Kim SH et al (2020) Managing uncertainty in data-driven simulation-based optimization. Comput Chem Eng. https://doi.org/10.1016/j.compchemeng.2019.106519

Jia LY, Alizadeh R, Jia H et al (2020) A rule-based method for automated surrogate model selection. Adv Eng Inf. https://doi.org/10.1016/j.aei.2020.101123

Jin YC (2011) Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm Evol Comput 1(2):61–70. https://doi.org/10.1016/j.swevo.2011.05.001

Liang S, Rasheed K (2008) ASAGA: an adaptive surrogate-assisted genetic algorithm. In: Proceeding of the 2008 conference on genetic and evolutionary computation conference(GECCO). https://doi.org/10.1145/1389095.1389289

Li F, Cai XW, Gao L (2019) Ensemble of surrogates assisted particle swarm optimization of medium scale expensive problems. Appl Soft Comput 74:291–305. https://doi.org/10.1016/j.asoc.2018.10.037

Li F, Shen WM, Cai XW (2020) A fast surrogate-assisted particle swarm optimization algorithm for computationally expensive problems. Appl Soft Comput 92:106303. https://doi.org/10.1016/j.asoc.2020.106303

Li YH (2020) A Kriging-based multi-point sequential sampling optimization method for complex black-box problem. Evol Intell. https://doi.org/10.1007/s12065-020-00352-5

Liu B, Grout V, Nikolaeva A (2018) Efficient global optimization of actuator based on a surrogate model assisted hybrid algorithm. IEEE Trans Ind Electron 65(7):5712–5721. https://doi.org/10.1109/TIE.2017.2782203

Liu B, Zhang QF, Gielen GGE (2014) A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. IEEE Trans Evol Comput 18(2):180–192. https://doi.org/10.1109/TEVC.2013.2248012

Liu HT, Meng JG, Xu SL, Yang SH, Wang XF (2016) Optimal weighted pointwise ensemble of radial basis functions with different basis functions. AIAA J 54(10):3117–3133. https://doi.org/10.2514/1.J054664

Liu HT, Cai JF, Ong Y (2017) An adaptive sampling approach for Kriging metamodeling by maximizing expected prediction error. Comput Chem Eng 106:171–182. https://doi.org/10.1016/j.compchemeng.2017.05.025

Liu RR, Li ZM (2014) Soft sensor for CO_x content in tail gas of PX oxidation side reactions based on particle filters and EM algorithm. In: 2013 International Conference on Future Software Engineering and Multimedia Engineering 6: 63–71. https://doi.org/10.1016/j.ieri.2014.03.011

Loshchilov I, Schoenauer M, Sebag M (2012) Self-adaptive surrogate-assisted covariance matrix adaptation evolution strategy. In: Proceedings of the 2012 international conference on genetic and evolutionary computation. https://doi.org/10.1145/2330163.2330210

Masato K, Hidetoshi M (2020) The limiting distribution of combining the t and Wilcoxon rank sum tests. Stat 54(4):871–884. https://doi.org/10.1080/02331888.2020.1809662

Mohamed WA (2017) A novel differential evolution algorithm for solving constrained engineering optimization problems. J Intell Manuf 29(3):659–692. https://doi.org/10.1007/s10845-017-1294-6

Nguyen DD, Long N (2021) An adaptive control for surrogate assisted multi-objective evolutionary algorithms. pp: 123–132. https://doi.org/10.1007/978-981-15-8289-9_12

Pan JS, Liu NX, Chu SC, Lai TT (2021) An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems. Inf Sci 561:304–325. https://doi.org/10.1016/j.ins.2020.11.056

Pan LQ, He C, Tian Y, Wang HD, Zhang XY, Jin YC (2019) A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. IEEE Trans Evol Comput 23(1):74–88. https://doi.org/10.1109/TEVC.2018.2802784

Regis RG (2014) Particle swarm with radial basis function surrogates for expensive black-box optimization. J Comput Sci 5(1):12–23. https://doi.org/10.1016/j.jocs.2013.07.004

Si T, Jana ND, Sil J (2011) Constrained function optimization using PSO with polynomial mutation. In: International conference on swarm 7076: 209-216

Sun CL, Jin YC, Cheng R, Ding JL, Zeng JC (2017) Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. IEEE Trans Evol Comput 21(4):644–660. https://doi.org/10.1109/TEVC.2017.2675628

Sun CL, Jin YC, Zeng JC, Yu Y (2015) A two-layer surrogate-assisted particle swarm optimization algorithm. Soft Comput 19(6):1461–1475. https://doi.org/10.1007/s00500-014-1283-z

Urquhart M, Ljungskog E, Sebben S (2020) Surrogate-based optimization using adaptively scaled radial basis functions. Appl Soft Comput 88:106050. https://doi.org/10.1016/j.asoc.2019.106050

Wang HD, Jin YC, Sun CL, Doherty J (2019) Offline data-driven evolutionary optimization using selective surrogate ensembles. IEEE Trans Evol Comput 23(2):203–216. https://doi.org/10.1109/TEVC.2018.2834881

Wang HD, Jin YC, Doherty J (2017) Committee-based active learning for surrogate-assisted particle swarm optimization of expensive

problems. IEEE Trans Cybern 47(9):2664–2677. https://doi.org/10.1109/TCYB.2017.2710978

Wu JL, Luo Z, Zhang N, Gao W (2018) A new sequential sampling method for constructing the high-order polynomial surrogate models. Eng Comput 35(2):529–564. https://doi.org/10.1108/EC-05-2016-0160

Ye PC, Pan G, Dong ZM (2018) Ensemble of surrogate based global optimization methods using hierarchical design space reduction. Struct Multidiscip Optim 58(2):537–554. https://doi.org/10.1007/s00158-018-1906-6

Yondo R, Andrés E, Valero E (2018) A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses. Prog Aeosp Sci 96:23–61. https://doi.org/10.1016/j.paerosci.2017.11.003

Zhai JY, Boukouvala F (2019) Nonlinear variable selection algorithms for surrogate modeling. AIChE J 65(8):e16601. https://doi.org/10.1002/aic.16601

Zhang XY, Tian Y, Cheng R, Jin YC (2015) An efficient approach to non-dominated sorting for evolutionary multiobjective optimization. IEEE Trans Evol Comput 19(2):201–213. https://doi.org/10.1109/TEVC.2014.2308305

Zhu H, Hu YM, Zhu WD (2019) A dynamic adaptive particle swarm optimization and genetic algorithm for different constrained engineering design optimization problems. Adv Mech Eng 11(3):1–27. https://doi.org/10.1177/1687814018824930