An Adaptive Surrogate-Assisted Particle Swarm Optimization for Expensive Problems

Xuemei Li

East-China Institute of Technology: East China University of Science and Technology

Shaojun Li (lishaojun@ecust.edu.cn)

East China University of Science and Technology

Research Article

Keywords: Surrogate-assisted evolutionary algorithm,Ensemble model,Function evaluations,Particle swarm optimization

Posted Date: April 13th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-410841/v1

An Adaptive Surrogate-Assisted Particle Swarm Optimization for Expensive Problems

Xuemei Li, Shaojun Li

(Key Laboratory of Advanced Control and Optimization for Chemical Processes, East China University of Science and Technology, Ministry of Education, Shanghai 200237, China)

Abstract

To solve engineering problems with evolutionary algorithms, many expensive objective function evaluations (FEs) are required. To alleviate this difficulty, the surrogate-assisted evolutionary algorithm (SAEA) has attracted increasingly more attention in both academia and industry. The existing SAEAs depend on the quantity and quality of the original samples, and it is difficult for them to yield satisfactory solutions within the limited number of FEs. Moreover, these methods easily fall into local optima as the dimension increases. To address these problems, this paper proposes an adaptive surrogate-assisted particle swarm optimization (ASAPSO) algorithm. In the proposed algorithm, an adaptive surrogate selection method that depends on the comparison between the best existing solution and the latest obtained solution is suggested to ensure the effectiveness of the optimization operations and improve the computational efficiency. Additionally, a model output criterion based on the standard deviation is suggested to improve the robustness and stability 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 the limited number of FEs. The performance of the proposed algorithm in solving engineering problems is verified by applying the algorithm to the PX oxidation process.

Keyword Surrogate-assisted evolutionary algorithm·Ensemble model·Function evaluations·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) tend to fall into a local optimum; to solve this problem, evolutionary algorithms have emerged. As a powerful optimization tool, evolutionary algorithms can address problems that are difficult to be optimized by traditional mathematical programming techniques, and they have achieved great success in practical industrial problems. For instance, Zhu et al. (2019) employed adaptive particle swarm optimization (PSO) and a genetic algorithm to different constrained engineering design problems, and the results demonstrated the superiority of the application of evolutionary algorithms to engineering processes. Mohamed (2017) applied differential evolution to engineering optimization problems, and obtained efficient and robust solutions. However, in complex practical problems, fitness assessment often involves costly numerical simulations or expensive experiments, which is a substantial challenge for the promotion of evolutionary algorithms. To address this problem, many researchers have developed various surrogate-assisted evolutionary algorithms (SAEAs). The surrogate model was originally an analysis model (Alizadeh et al. 2020), and uses a limited number of samples and a given model structure to fit the model expression close to the input-output relationship. The principle of SAEAs is that some of the real fitness assessments in the evolution process are replaced by 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 accurate function evaluations (FEs).

Common surrogate models in SAEAs include radial basis functions (RBFs) (Liu et al. 2016; Sun et al. 2017), polynomial regression (PR) (Wu et al. 2018; Si et al. 2011), the Kriging model (Fu et al. 2020; Huang et al. 2018), artificial neural networks (ANNs) (Pan et al. 2019), and ensemble surrogate models (Wang et al. 2017; Li et al. 2019) composed of multiple independent surrogate models. Among these models, RBFs have high prediction accuracy and good global and local approximation capabilities when

the number of samples is not sufficient (Jia et al. 2020). They are especially suitable for solving high-dimensional nonlinear problems (Yu et al. 2018). A PR is simple in structure and easy to construct, and has low calculation costs. However, it is not suitable for complicated nonlinear problems, and its modeling accuracy is excessively dependent on the quality of samples (Wu et al. 2018). The principle of the Kriging model is similar to that of the RBF model, and it has high prediction accuracy when there are enough samples. It is suitable for dealing with low/medium-dimensional problems and low-order nonlinear problems in high-dimensional space (Fan et al. 2020). ANNs can obtain the relationship between input and output variables by learning from the known sample. Although they are characterized by high precision and good robustness, their learning speed is slow. Finally, the ensemble surrogate model can integrate the features of different single models, and is suitable for dealing with some complex or medium-high problems.

The key issue to be addressed in SAEAs is determining which individuals should be selected for evaluation or reevaluation using the real fitness function. The most direct method is to evaluate those individuals with a good fitness value or high approximation accuracy according to the results obtained from the surrogate model (Fan et al. 2020; Zhang et al. 2015). In addition, some representative individuals have been selected in different research. For example, Cheng et al. (2016) and Gong et al. (2017) selected the individuals closest to each reference vector or the best in each class via clear or fuzzy clustering. Liu et al. (2017) and Hüllen et al. (2017) posited that the individuals with high uncertainty may be good candidates for reassessment. Moreover, according to Li (2020), the large uncertainty in individual approximations indicates that the fitness landscape around these solutions has not been well exploited, and more information may be provided to find better solutions.

The process, in which appropriate points are selected for evaluation 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, and a variety of research on this process has been reported. Usually, the

combination of global and local model management is utilized, which is also referred to as exploration and exploitation. A reasonable exploration and exploitation design is helpful for improving the convergence rate of the evolutionary search process. For example, Liu et al. (2018) employed derivative-based optimization to first 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. Garud and Mariappan (2019) partitioned a training dataset into several Delaunay simplices and then selected the most promising region to place one new point. The partition emphasizes the ability to obtain a global optimum, while the operation of placing new points improves the quality of the local model. Wang et al. (2017) proposed a global model management method inspired by committee-based active learning, which searches the best and most uncertain solutions used to update the global surrogate model. Simultaneously, a local surrogate model is constructed around the currently available optimal solutions. When the evolutionary search using the global model management strategy is not further improved, the whole evolutionary search is switched to the local search and vice versa. This achieves a good balance between the global and local models. Sun et al. (2015) established a global surrogate model and multiple local surrogate models for the approximation of the fitness of the population. The global model guides the search for the optimum, and the local surrogate model improves the accuracy of fitness estimation.

The methods described previously can achieve good results; however, with the increase of the dimension, some of these models easily fall into local optima, or cause the algorithm to keep switching between global and local searching. This indicates that the existing relevant approaches cannot successfully find an optimal solution with real significance, which wastes time and increases the computational cost. In addition, the solution obtained via an SAEA defaults to the current optimal solution, which is no longer compared with the best solution in the current training dataset. The reliability of this operation largely depends on the accuracy of the surrogate model, which is mainly decided by the quantity and quality of the training dataset. When the training dataset is

sufficiently large and of good quality, the trained surrogate model has high prediction accuracy. However, in some actual industrial processes, it is difficult to obtain enough data to construct a surrogate model; moreover, the obtained data are always accompanied by noise, heterogeneity, and time variability (Dew et al. 2020). As a result, a surrogate model may mislead the evolutionary search process in the wrong direction and fail to yield a satisfactory optimal solution.

To effectively address these problems, in this study, an adaptive surrogate-assisted PSO (ASAPSO) algorithm is proposed by taking advantage of the ability of ensemble learning to improve the prediction accuracy of the ensemble model (Ye et al. 2018). In the ASAPSO algorithm, an adaptive surrogate selection method is executed between an ensemble model based on ensemble learning and a single RBF model, and the selected surrogate is used to assist the PSO algorithm to search for an optimal solution. The method relies on the comparison of the solution obtained by the surrogate-assisted PSO algorithm and the best solution in the current training dataset. Furthermore, by combining the limitation of the samples and the randomness of sampling, instead of simply calculating the average of the sub-models, the influence of anomalies in the modeling process is considered when calculating the output of the ensemble model. In this study, considering the characteristics of data in engineering processes, RBFs are selected for both the single surrogate model and the base model of ensemble learning.

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

2 Related techniques

2.1 RBF model

The basic concept of RBF model construction is to determine a set of reliable

sample points $x_i (i = 1,2,\Lambda\ ,n)$. The radial distance $\varphi_i (i = 1,2,\Lambda\ ,n)$ between variable x and each data point $x_i (i = 1,2,\Lambda\ ,n)$ can be expressed as follows:

$$\varphi_i = \left\| x - x_i \right\| \ (i = 1,2,\Lambda\ ,n), \tag{1}$$

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

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

where $\beta = [\beta_1, \beta_2, \Lambda\ , \beta_n]^T$ 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 points x_i. The substitution of Eq. (3) into Eq. (2) yields the following:

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

$\Phi = \begin{bmatrix} \phi(\left\| x_1 - x_1 \right\|), & \Lambda\ , & \phi(\left\| x_n - x_1 \right\|) \\ \phi(\left\| x_1 - x_2 \right\|), & \Lambda\ , & \phi(\left\| x_n - x_2 \right\|) \\ & \mathrm{M} & \\ \phi(\left\| x_1 - x_n \right\|), & \Lambda\ , & \phi(\left\| x_n - x_n \right\|) \end{bmatrix}$, $f = [f(x_1), f(x_2), \Lambda\ , f(x_n)]^T$. 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	$\phi(r) = e^{-\varepsilon r^2}$
Multiple quadratic(MQ)	$\phi(r) = \sqrt{\varepsilon^2 + r^2}$
Inverse quadratic	$\phi(r) = \frac{1}{\sqrt{\varepsilon^2 + r^2}}$
Thin-plate spline	$\phi(r) = r^2 \log(\varepsilon r^2 + 1)$
Cubic spline	$\phi(r) = (r + \varepsilon)^2$

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, and 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.

2.3 Particle swarm optimization algorithm

The PSO algorithm searches for the optimal solution of 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 swarms 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 $v_i(t) = (v_{i1}(t), v_{i2}(t), \Lambda\ , v_{iD}(t))$ and $x_i(t)$=$(x_{i1}(t), x_{i2}(t), \Lambda\ , x_{iD}(t))$ are respectively the velocity and position of particle i at the t-th iteration, $P_i(t)$=$(p_{i1}(t), p_{i2}(t), \Lambda\ , p_{iD}(t))$ is the best historical location for the discovery of particle i (called the individual best), $P_g(t)$=$(p_{g1}(t), p_{g2}(t), \Lambda\ , p_{gD}(t))$ is the best historical location for the swarm (called the global best), r_1 and r_2 are two random numbers uniformly generated 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.

The detailed process of the PSO algorithm is shown in Algorithm 1.

Algorithm 1: PSO algorithm

---

1. **Input:** t_{max}, the maximum number of iterations; the initial number of iterations $t = 0$

2. Initialize a population (*POP*) with n particles, and give each particle in *POP* a random velocity

3. Evaluate all particles in *POP*, and take the positions of all the particles in *POP* as the optimal solutions they find, which is denoted as P_i $(i = 1,2,\Lambda\ ,n)$; the solution

---

with the best fitness is denoted as P_g

4. **While** $t < t_{max}$ **do**

5. Update the velocity and positions of all particles in *POP* according to Eqs. (6) and (7)

6. Evaluate all particles in *POP*

7. Update P_i and P_g

8. $t = t + 1$

9. end **while**

10. **Output** P_g, the solution with the best fitness

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 algorithm. 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 dataset. When the latest obtained optimal solution is better than the already existing best solution, the RBF model is selected as the surrogate; otherwise, an ensemble model based on ensemble learning is selected as the surrogate. This method is used to ensure the effectiveness of the optimization operation and to reduce computational resources.

(2) An ensemble model output criterion is suggested. Its purpose is to enhance the stability and generalization ability of the 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 on the output of the ensemble model, the output of the ensemble model is obtained by the reasonable analysis and processing of the T sub-models. Two statistics, the average and median, are used in this work. 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 function evaluations (FE_{max}) is used as the termination condition of the ASAPSO algorithm. The detailed steps of ASAPSO are as follows.

Step 1: M samples $X = [x_1, x_2, \Lambda, x_M]^T$ is taken from the definition domain of the function by using Latin hypercube sampling (LHS) . j is the number of real function evaluations; set $j = 0$.

Step 2: Calculate the real fitness of X, obtain $Y = [Y_{x_1}, Y_{x_2}, \Lambda, Y_{x_M}]^T$, and construct the initial training dataset $Dt = [X,Y]$. Additionally, $j = M$, $Y^j_{best} = minimize(Y)$, $Y_{gbest} = Y^j_{best}$.

Step 3: If Y_{gbest} is less than or equal to Y^j_{best}, go to step 4; otherwise, go to step 6.

Step 4: Construct the RBF model using Dt.

Step 5: Obtain the optimal solution, *gbest*, of the RBF model by using the PSO algorithm, and go to step 8.

Step 6: Establish an ensemble model using Dt.

Step 7: Obtain the optimal solution, *gbest*, of the ensemble model by using the PSO algorithm.

Step 8: Evaluate *gbest* via a real fitness function, obtain Y_{gbest}, and add ($gbest$,Y_{gbest}) to Dt. Additionally, set $j = j + 1$ and $Y^j_{best} = minimize(Y)$.

Step 9: If j is less than FE_{max}, go to step 3; otherwise, go to step 10.

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.

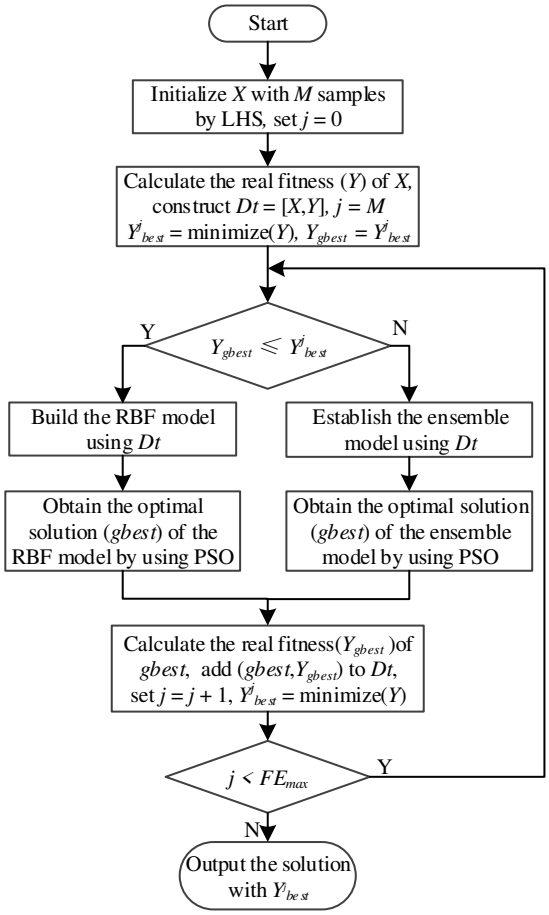

Fig. 1 The flowchart of the ASAPSO algorithm

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 initial 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 that each data point in Dt is 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.

Table 2 Information on the common benchmark functions

Function	Problem	Variable range	Optimum	Feature
F1	Ellipsoid	[-5.12,5.12]	0	Uni-modal
F2	Sum square	[-10,10]	0	Uni-modal
F3	Sphere	[-100,100]	0	Uni-modal
F4	Griewank	[-600,600]	0	Multi-modal
F5	Schwefel2.21	[-500,500]	0	Multi-modal
F6	Alpine	[-10,10]	0	Multi-modal
F7	Quartic	[-1.28,1.28]	0	Multi-modal
F8	Ackley	[-32,32]	0	Multi-modal
F9	Rastrigin	[-5,5]	0	Very complicated multi-modal
F10	Rosenbrock	[-2.048,2.048]	0	Multi-modal with narrow valley

4.1 Low-dimensional problems

To verify the performance of the proposed ASAPSO algorithm in low dimensions, it was applied to 10-*D* and 20-*D* problems, and its performance was compared with the performance 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 (RMSEs) of the PR, RBF, and Kriging models. In the experiment, WTA1 continued to evaluate the optimum of the surrogate ensemble and added it to the training dataset for model updating.

(3) MAES-ExI is a Kriging-based SAEA with 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 3, and the best fitness of the four algorithms for each problem is indicated in bold.

Table 3 Average best fitness and standard deviation of the four comparison algorithms

Function	D	ASAPSO	CALSAPSO	MAES-ExI	WTA1
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
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
F3	10	1.38E+02 ± 2.83E+02	**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
F4	10	5.00E+00 ± 7.87E+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
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
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
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
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
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
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
Average rank		1.4	2.3	3.7	2.6
Wilcoxon test		NA	0.023	0.000	0.004

As revealed by the statistical results presented in Table 3, 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 on the 10-*D* problem for a portion of the test functions, especially for uni-modal functions. The reason for this may be that only 60 new solutions were evaluated in the entire optimization process, and most of the solutions could be obtained by using the RBF-assisted PSO algorithm; the ensemble learning method did work significantly, which was not different from the other existing SAEAs. When the dimension increased, the ASAPSO algorithm exhibited a better optimization effect. In addition, the results reported in Table 3 were analyzed by a Friedman test (López-Vázquez et al. 2019) at the significance level of $\alpha = 0.05$, and p-value = 0.000 < 0.05 was obtained; this indicates that the four algorithms had significant differences. Therefore, a Wilcoxon test (Xu et al. 2020) was conducted at the significance level of $\alpha = 0.05$ to further compare the differences between the other three algorithms and ASAPSO. The last row of Table 3 lists the results of the Wilcoxon test, and the p-values were respectively 0.023, 0.000, and 0.004, which are all less than 0.05; this means that the ASAPSO algorithm performed significantly better than the other three algorithms.

To obtain a more comprehensive understanding of the ASAPSO algorithm, several

representative test functions were analyzed, and the convergence profiles of three representative benchmark functions on D = 10, 20 are presented in Fig. 2. 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 Figs. 2(a) and 2(b), it is evident that ASAPSO exhibited a strong optimization ability; in particular, on the 20-*D* function, it converged to the optimal solution when the number of FEs reached half the maximum value. 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 Figs. 2(c) and 2(d), whether on the 10-*D* or 20-*D* function, ASAPSO quickly converged to an optimal solution, and the optimum was significantly better than those obtained by the other three algorithms. Finally, Ratrigin is a very complicated multi-modal function 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 can be used to check the practicability of the algorithm. According to Figs. 2(e) and 2(f), whether on the 10-*D* or 20-*D* function, 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 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 CALSAPSO and WTA1, both of which use a heterogeneous ensemble, performed more robustly on various problems as compared to ASAPSO and MAES-ExI, which use one single model or homogeneous model as the surrogate. This observation suggests that a heterogeneous ensemble is more reliable for fitness estimation when little is known about the problem to be optimized.

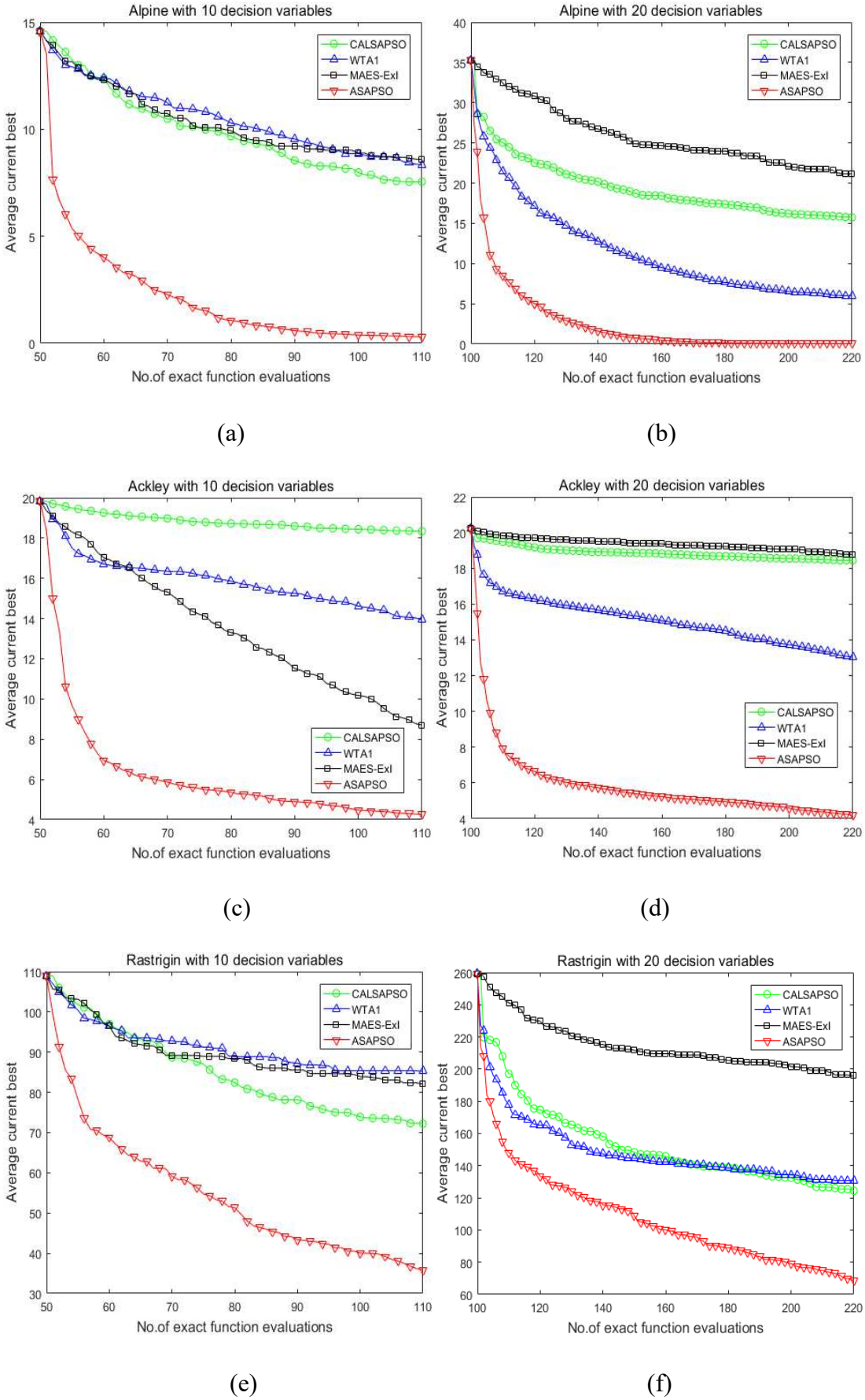

Fig. 2 The convergence profiles of three representative benchmark function on 10-D, 20-D

4.2 Scalability test

CALSAPSO, WTA1, and MAES-ExI were developed mainly to address low-dimensional problems (Wang et al. 2017). Therefore, the performance of ASAPSO in dealing with high-dimensional problems with dimensions over 30 (Fan et al. 2020) 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. Thus, to verify the scalability of the ASAPSO algorithm, it was compared with GPEME and iDEaSm in terms of five classical benchmark functions on a 50-*D* problem. 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 (Fan et al. 2020). For a fair comparison, the ASAPSO algorithm was also independently run 25 times. The average best fitness and standard deviation of the three comparison algorithms on the 50-*D* problem are reported in Table 4. The best fitness of the three algorithms for each problem is indicated in bold.

Table 4 Average best fitness and standard deviation of three comparison algorithms on 50-D

Function	D	ASAPSO	iDEaSm	GPEME
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
Wilcoxon test		NA	0.043	0.043

It can be seen from Table 4 that for the five classical test functions on the 50-*D* problem, 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. As determined by a Wilcoxon test at the significance level of $\alpha = 0.05$, the two *p*-values were 0.043, which are both less than 0.05; this demonstrates that GPEME and iDEaSm were significantly worse than ASAPSO. The reason for this is that ASAPSO selects RBFs as the surrogate models, which is suitable for overcoming the problem of inadequate samples. 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 uses 5000 sample points for 50-*D* problems. However, the original Kriging model is used in GPEME. Overall, the proposed ASAPSO algorithm exhibited excellent 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. 3.

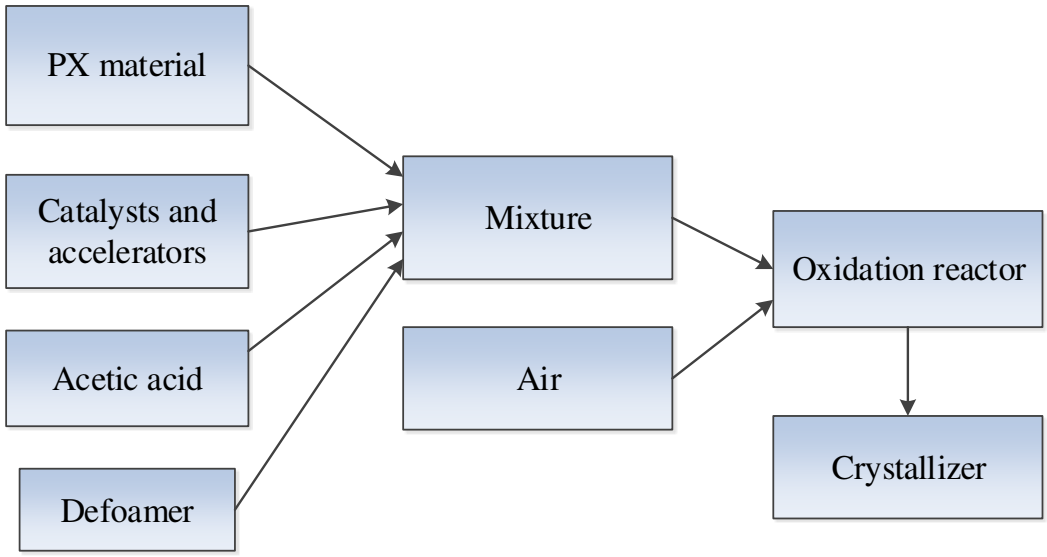

Fig. 3 The flowchart of PX oxidation reaction process

Under the conditions of a 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 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 2012). 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 samples was 30. The ASAPSO algorithm was terminated when the number of FEs reached 66. It should be noted that the real objective function was calculated by Aspen Plus 10 software during the entire optimization process. The comparison between the actual conditions and the results optimized by the ASAPSO algorithm is presented in Table 5.

Table 5 The results comparison between actual condition and the ASAPSO algorithm

Variable	x_1	x_2	x_3	x_4	x_5	x_6	PX combustion loss
Actual	242.99	124.67	187.28	3519.90	2.46	187.28	15.13
Optimization	216.73	140.69	183.00	4931.28	2.02	186.05	13.34

It can be seen from Table 5 that the ASAPSO algorithm 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 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, which 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 it is recognized that the RBF-assisted PSO algorithm is invalid; otherwise, the RBF-assisted PSO algorithm is executed to search for an optimal solution. For the ensemble model, a model output criterion was 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 the other five compared 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).

Compliance with ethical standards

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

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

Author's contribution **Xuemei Li**: Conceptualization, Methodology, Software, Writing-Reviewing and Editing. **Shaojun Li**: Resources, Supervision, Project administration, Funding acquisition.

References

Alizadeh R, Allen JK, Mistree F (2020) Managing computational complexity using surrogate models: a critical review. Res Eng Design 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

Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 20(5): 773-791. https://doi.org/10.1109/TEVC.2016.2519378

Dew R, Ansari A, Li Y (2020) Modeling dynamic heterogeneity using Gaussian processes. J Mark Res 57(1): 55-77. https://doi.org/10.1177/0022243719874047

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, vol 91. 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, vol 206. 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

Garud SS, Mariappan N, Karimi IA (2019) Surrogate-based black-box optimization via domain exploration and smart placement. Comput Chem Eng, vol 130. https://doi.org/10.1016/j.compchemeng.2019.106567

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

Gong W, Duan QY (2017) An adaptive surrogate modeling-based sampling strategy for parameter optimization and distribution estimation (ASMO-PODE). Environ Model Softw 95: 61-75. https://doi.org/10.1016/j.envsoft.2017.05.005

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, vol 136. 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, vol 45. https://doi.org/10.1016/j.aei.2020.101123

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 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

López-Vázquez C, Hochsztain E (2019) Extended and updated tables for the Friedman rank test. Comm. Statist. Theory Methods 48(2): 268-281. https://doi.org/10.1080/03610926.2017.1408829

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

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

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

Xu K, Zhou YQ (2020) Maximum-type tests for high-dimensional regression coefficients using Wilcoxon scores. J Stat Plan Infer 211: 221-240. https://doi.org/10.1016/j.jspi.2020.06.011

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

Yu HB, Tan Y, Zeng JC, Sun CL, Jin YC (2018) Surrogate-assisted hierarchical particle swarm optimization. Inf Sci 454-455: 59-72. https://doi.org/10.1016/j.ins.2018.04.062

Zhang XY, Tian Y, Cheng R, Jin YC (2015) An efficient approach to nondominated 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

Figures

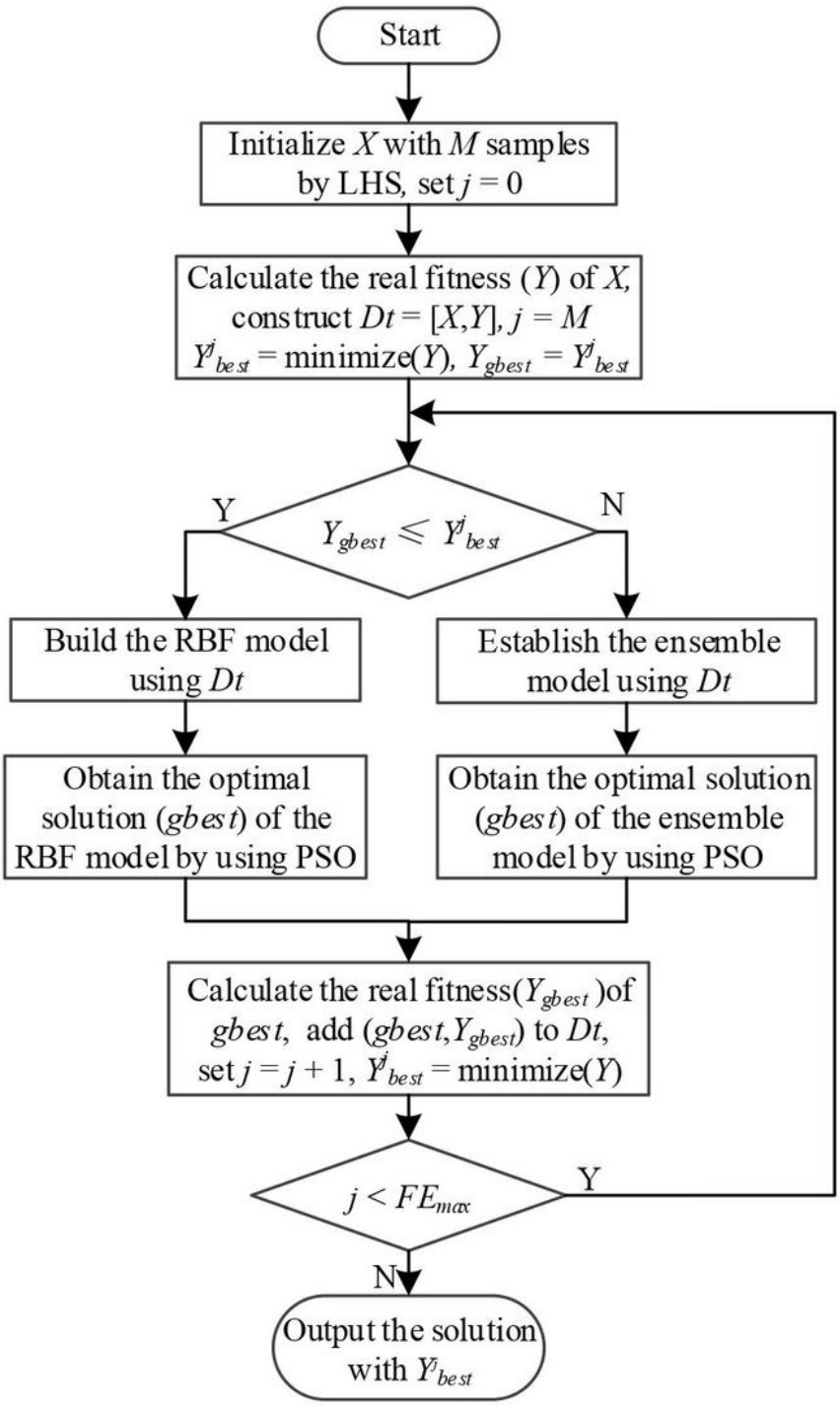

Figure 1

The flowchart of the ASAPSO algorithm

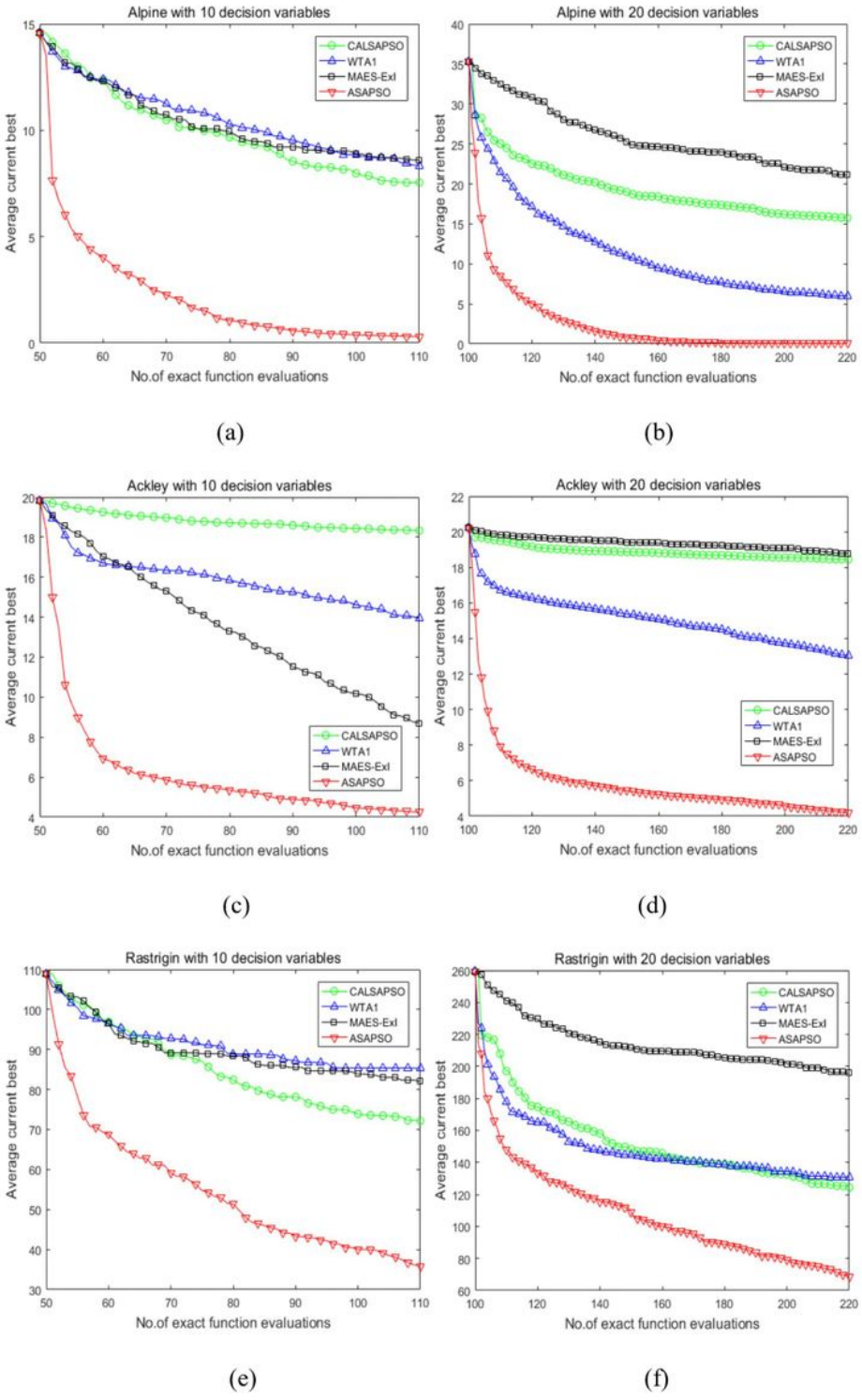

Figure 2

The convergence profiles of three representative benchmark function on 10-D, 20-D

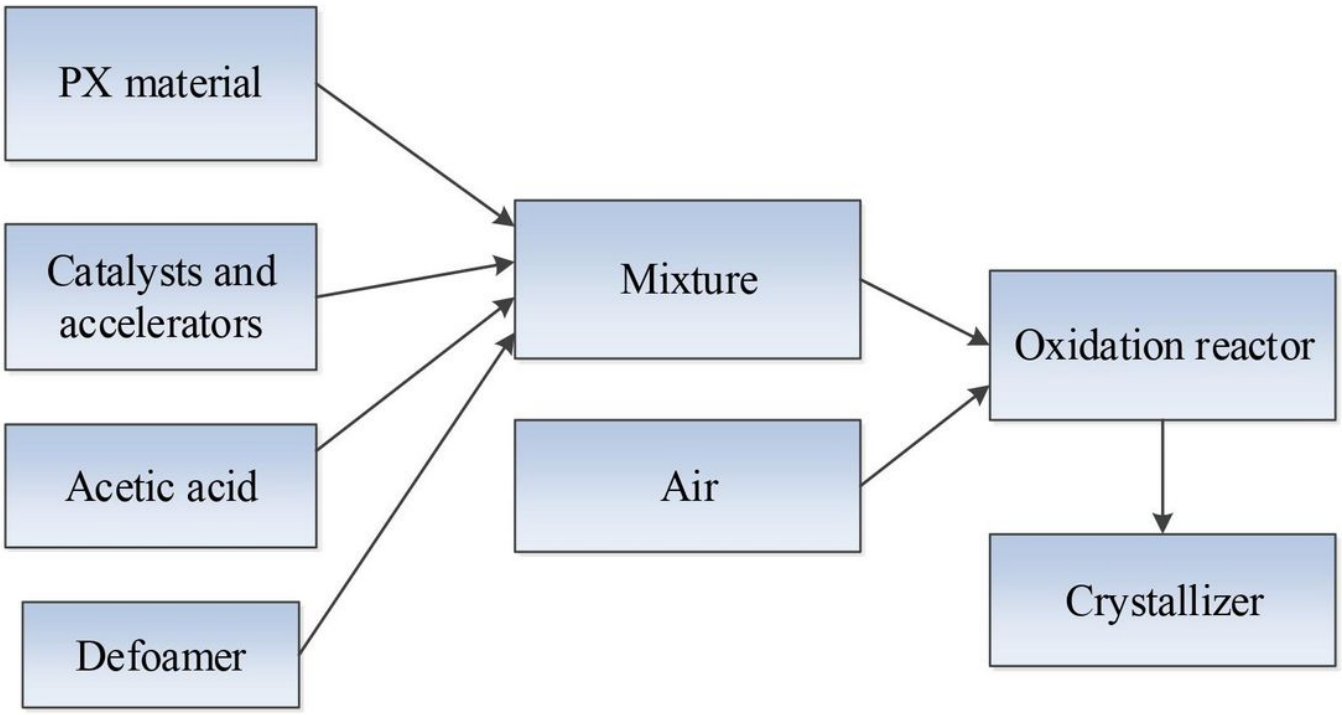

Figure 3

The flowchart of PX oxidation reaction process