A Surrogate-Assisted Multiswarm Optimization Algorithm for High-Dimensional Computationally Expensive Problems

Fan Li, *Student Member, IEEE*, Xiwen Cai, *Member, IEEE*, Liang Gao, *Member, IEEE*, and Weiming Shen, *Fellow, IEEE*

***Abstract*—This article presents a surrogate-assisted multiswarm optimization (SAMSO) algorithm for high-dimensional computationally expensive problems. The proposed algorithm includes two swarms: the first one uses the learner phase of teaching-learning-based optimization (TLBO) to enhance exploration and the second one uses the particle swarm optimization (PSO) for faster convergence. These two swarms can learn from each other. A dynamic swarm size adjustment scheme is proposed to control the evolutionary progress. Two coordinate systems are used to generate promising positions for the PSO in order to further enhance its search efficiency on different function landscapes. Moreover, a novel prescreening criterion is proposed to select promising individuals for exact function evaluations. Several commonly used benchmark functions with their dimensions varying from 30 to 200 are adopted to evaluate the proposed algorithm. The experimental results demonstrate the superiority of the proposed algorithm over three state-of-the-art algorithms.**

***Index Terms*—Computationally expensive problems, multiswarm optimization, particle swarm optimization (PSO), surrogate model, teaching-learning-based optimization (TLBO).**

I. Introduction

EVOLUTIONARY algorithms (EAs) are widely used to solve complicated problems in various fields for their good global search capability [1]–[3]. Many EAs usually consume plenty of function evaluations (FEs) to obtain a satisfied solution in solving complex or high-dimensional problems. These algorithms face great challenges in solving computationally expensive problems as one FE of these expensive problems can cost plenty of time or material resources [4], [5]. To address this issue, surrogate-assisted EAs (SAEAs) have been widely used as surrogate models can approximate a real-simulation model [5]. The generally utilized surrogate models include radial basis function (RBF) models [6], [7]; polynomial regression models [8]; artificial neural networks [9]; Kriging models [10]; and support vector regression models [11]. Many studies have compared the performance of different surrogate models [8], [12], [13], and the results have shown that different surrogate models are suitable for their corresponding problems [14]. In addition, as the characteristics of some problems in engineering practice are unknown in *a priori*, hybrid surrogates and an ensemble of surrogates are often applied to obtain a robust and accurate approximation [15]–[17].

Commonly used SAEAs can be roughly classified into two categories: 1) global-surrogate assisted EAs and 2) local-surrogate assisted EAs. Global surrogate models are usually applied to approximate the entire landscape of a problem. For instance, Zhou *et al.* [18] utilized a global Kriging model to select promising individuals in genetic algorithms. Yang *et al.* [19] utilized a global Kriging model to accelerate the search progress of the particle swarm optimization (PSO). Local surrogate models are often utilized to improve the approximation accuracy in a small area. The accuracy of the models is gradually enhanced with more individuals being evaluated in the local area. For example, Lim *et al.* [20] used two kinds of surrogate models to improve the local search efficiency of a memetic algorithm.

After surrogate models are built, different ways of using them to prescreen individuals for exact FEs will result in different optimization efficiencies. Generally, three types of criteria are commonly used for prescreening, including performance-based criteria, uncertainty-based criteria, and their combinations. In the performance-based criterion, candidate individuals with smaller predicted fitness values are usually selected for exact FEs in minimization problems. For instance, Regis [21] designed multiple trial positions for each particle, then used an RBF model to select a position with the minimum predicted fitness value. Sun *et al.* [22] used a global and a local surrogate-assisted PSO algorithm (SAPSO) for computationally expensive problems. The particle with a smaller predicted fitness value than its personal historical best was exactly evaluated. The uncertainty-based criterion usually selects individuals with great uncertainties for exact FEs. This criterion can guide the search for some

Manuscript received September 3, 2019; revised November 14, 2019; accepted January 7, 2020. This work was supported in part by the National Natural Science Foundation for Distinguished Young Scholars of China under Grant 51825502, in part by the 111 Project under Grant B16019, and in part by the Program for HUST Academic Frontier Youth Team under Grant 2017QYTD04. This article was recommended by Associate Editor H. Ishibuchi. *(Corresponding author: Liang Gao.)*

The authors are with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: d201780171@hust.edu.cn; 864533724@qq.com; gaoliang@mail.hust.edu.cn; wshen@ieee.org).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2020.2967553

not-well-explored areas. Generally, the performance-based criterion is used to exploit current promising areas, while the uncertainty-based criterion is used to explore sparse areas. Many combinations of the above two criteria are used to keep a good balance of global exploration and local exploitation. For example, Liu *et al.* [23] used a dimension reduction method to construct a Kriging surrogate model in a lower dimensional space, then offspring with better lower confidence bound (LCB) values were selected for exact FEs. Guo *et al.* [24] proposed an LCB criterion consisting of predicted mean values and standard deviations of a heterogeneous ensemble surrogate model to solve expensive multiobjective problems. Li *et al.* [25] used an LCB criterion based on two different surrogate models to select a promising position for each particle, and the weight coefficient of the LCB criterion was changed to control the evolutionary progress. In addition, surrogate models can guide the search of EAs to promising directions by using optima of the models [21], [22], [26], [27]. The optima of surrogate models are commonly used in SAPSOs. The optimum of the surrogate model will replace the global best of the swarm if it is better [27], [28], which can efficiently improve the search efficiency of the standard PSO (SPSO) algorithm. More details about SAEAs can be found in [20] and [29]–[33].

Until now, significant efforts have been dedicated to solving high-dimensional expensive problems. For instance, two types of surrogate-assisted algorithms are cooperatively applied to solve 200-D problems in [26]. A surrogate-assisted SPSO algorithm was mainly used to exploit the current promising area and a surrogate-assisted social learning-based PSO (SL-PSO) algorithm was utilized to enhance global searchability. A surrogate-assisted hierarchical PSO algorithm was proposed for high-dimensional expensive problems in [34]. An SPSO was evolved with the help of a local surrogate model and an SL-PSO algorithm was used to find an optimum of the model. Recently, an evolutionary sampling-assisted optimization (ESAO) method was proposed in [35], which combined the global exploration ability of differential evolution and a local surrogate model to solve expensive problems with their dimensions up to 200. A Gaussian process surrogate model with a dimension reduction technique was used to prescreen the offspring produced by three different strategies in [36], and the local Gaussian process model was used to assist a local search strategy to extensively exploit the promising regions. Global and local surrogates were used to assist the mutation in differential evolution algorithms [37] and genetic algorithms [38] to guide the search of the populations. The population was granulated into coarse-grained individuals and fine-grained ones in [39]. Individuals with maximum uncertainty in the two subsets and minimal approximated fitness value were exactly evaluated. There is also some research about large-scale expensive problems. Sun *et al.* [40] proposed a fitness approximation-assisted competitive swarm optimizer for 500-D problems, and achieved competitive performance when the number of FEs (NFEs) is limited to 10 times of the dimension. Pang *et al.* [41] proposed an adaptive surrogate-assisted cooperative coevolution framework for 1000-D problems. This algorithm can adaptively construct surrogate models for different subproblems.

It is hard for EAs to search for global optima in high-dimensional spaces due to the curse of dimensionality. SAEAs also encounter the same challenge when the dimension of a problem is high. Enhancing population diversity is an alternative strategy to relieve this challenge as more exploratory ability can be obtained. The population diversity can be enhanced by using multiple swarms as different subswarms can explore different search spaces simultaneously to efficiently find promising areas. In addition, the multiswarm optimization algorithm can combine the advantage of different swarms if heterogeneous swarms are used. Multiswarm optimization has been efficiently used to solve high-dimensional computationally cheap problems [42]. However, most of the existing SAEAs used only one swarm during the optimization process except that SA-COSO [26] used two swarms. This article tries to develop a surrogate-assisted multiswarm optimization (SAMSO) algorithm to efficiently solve high-dimensional expensive problems. Taking advantage of the good global searchability of the teaching-learning-based optimization (TLBO) algorithm and the fast convergence ability of the SPSO algorithm, the two swarms are used in the proposed algorithm. They can communicate information in the optimization process, and their swarm sizes are changing with iterations to balance the global exploration and local exploitation. Currently, the investigation on the hybridization of the TLBO and the PSO mainly focuses on using the TLBO as a strategy to help the PSO obtain a better solution [43]. In [44] and [45], all particles first followed the updating rules of the SPSO, then the particle that failed to improve its fitness value would enter into the peer-learning phase of the TLBO. In [46], the learner phase of the TLBO was periodically invoked in the evolutionary progress of the PSO to avoid particles being trapped in local optima. The method of using the SPSO and the TLBO as subswarms has not been investigated. In addition, the commonly used coordinate system in the SPSO is fixed and thus not well suitable for different function landscapes. To further enhance the search efficiency of the SPSO on different fitness landscapes, original and eigencoordinate systems are collaboratively used to generate promising positions for the SPSO in the proposed algorithm. To ensure that the SAMSO algorithm can obtain a good solution with limited FEs, a novel prescreening criterion is proposed to select promising individuals for exact FEs.

The remainder of this article is arranged as follows. Section II provides the preliminaries of the work presented in this article. Section III describes the main components of the proposed SAMSO algorithm. Section IV presents a numerical study to demonstrate and validate the proposed algorithm. Section V concludes this article and discusses some future work.

II. Preliminaries

A. Standard PSO Algorithm

The PSO is commonly used in different fields for its simplicity and powerful search capability. Various approaches

have been proposed to improve the search efficiency of the original PSO [47]–[49]. For example, Shi and Eberhart [50] added an inertia weight to the updating rules of the PSO to control global exploration and local exploitation. In [50], particles in the swarm updated their velocities and positions according to (1) and (2). The updating rules are also used in this article. Hereinafter, we also use SPSO to denote the PSO with the inertia weight

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot R_1 \cdot (p_i(t) - x_i(t)) + c_2 \cdot R_2 \cdot \big(p_g(t) - x_i(t)\big) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

where v_i is the speed of the ith particle. c_1 and c_2 are learning factors. R_1 and R_2 are diagonal matrices whose elements are between 0 and 1. x_i is the current position of the ith particle. p_i is the personal historical best position of the ith particle and p_g is the global optimum position of the swarm.

B. Teaching-Learning-Based Optimization Algorithm

The TLBO was inspired by a teaching system consisting of a teacher and some learners [51]. The teacher shares his or her information with the learners to enhance the quality of the learners. The teacher phase and the learner phase are devised to enhance the performance of each learner [44].

During the teacher phase, each learner studies from the best individual (the teacher) in the population to improve him or herself. Consider that the teaching result of a teacher is also influenced by the mean capability of the class [51]. The learner x_i updates his or her position during the teacher phase as follows:

$$x_{\text{new},i} = x_i + r_i \cdot (x_{\text{teacher}} - (T_F \cdot x_{\text{mean}})) \quad (3)$$

where r_i is a random number in [0.1]. T_F, a teaching factor that decides the value of mean to be changed, can be either 1 or 2. $x_{\text{new},i}$ will replace x_i if $x_{\text{new},i}$ is better than x_i.

Each learner attempts to improve its knowledge through the interaction with other learners in the learner phase. Specifically, a learner x_i randomly selects a peer learner x_j, then it selects direction to move according to (4) and (5). $x_{\text{new},i}$ will replace x_i if $x_{\text{new},i}$ is better than x_i

$$x_{\text{new},i} = x_i + r_i \cdot \big(x_i - x_j\big) \quad \text{if} \quad f(x_i) < f\big(x_j\big) \quad (4)$$

$$x_{\text{new},i} = x_i + r_i \cdot \big(x_j - x_i\big) \quad \text{if} \quad f\big(x_j\big) \le f(x_i). \quad (5)$$

C. Eigencoordinate System

The eigencoordinate system was originally used in the covariance matrix adaptation evolution strategy to produce promising offspring by learning the landscape of a problem [52]. Later, it was used to address the challenge that the commonly used coordinate system in nature-inspired optimization algorithms is fixed and thus not well suitable for different fitness landscapes [53]. The performance of the SPSO is changed if the search space is rotated [54], and this is called the rotation variance issue. It was proven in [55] that the SPSO is rotationally variant. The performance of the SPSO is degraded on nonseparable and ill-conditioned problems for this issue [56]–[58]. More promising offspring can be produced by using the eigencoordinate system as it can indicate the real landscape of a problem to some degree. Its effectiveness has been validated in [53], [59], and [60]. The method of establishing the eigencoordinate system is introduced in [53]. A covariance matrix (C) of some selected samples is first established, then an orthogonal matrix B is obtained from the eigendecomposition of C. Finally, the eigencoordinate system is established by using columns of B. In the SPSO, the velocity of a particle in the original coordinate system is produced by using (6), while (7) is used to produce the velocity in the eigencoordinate system

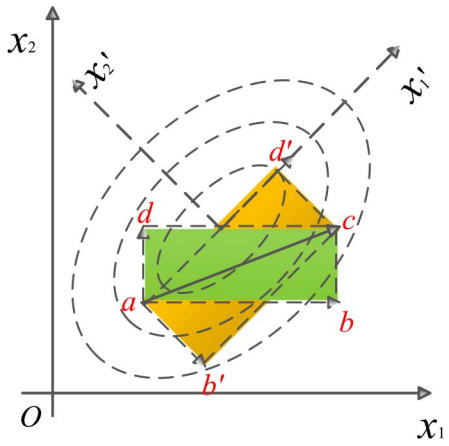

Fig. 1. Display the new position in the original coordinate system and eigencoordinate system.

$$\vec{v}_i(t+1) = w\vec{v}_i(t) + c_1R_1(\vec{p}_i(t) - \vec{x}_i(t)) + c_2R_2\big(\vec{p}_g(t) - \vec{x}_i(t)\big) \quad (6)$$

$$\vec{v}_i(t+1) = w\vec{v}_i(t) + c_1BR_1B^T(\vec{p}_i(t) - \vec{x}_i(t)) + c_2BR_2B^T\big(\vec{p}_g(t) - \vec{x}_i(t)\big) \quad (7)$$

where R_1 and R_2 are diagonal matrices whose elements are between 0 and 1. x_i is the current position of the ith particle. p_i is the personal best of the ith particle. p_g is the best particle in the swarm.

Take the example in [53] to analyze the effectiveness of the strategy. For a 2-D problem with variable correlation, the reachable location of a particle in both coordinate systems is shown in Fig. 1. If only one coordinate system is used, the particle will fall into the rectangular areas $abcd$ by using the original coordinate system x_1ox_2 or in $ab'cd'$ by using the eigencoordinate system $x_1'ox_2'$. It is evident that the $ab'cd'$ contains more promising areas, so the probability of obtaining a superior solution is bigger. In addition, if the new position is randomly produced in the original and eigencoordinate systems, the area covering $abcd$ and $ab'cd'$ may be reached. The entire search space is increased, which is good for exploration.

D. RBF Models

RBF models utilize a weighted sum of basic functions to roughly approximate the landscapes of problems [32]. For a data set consisting of the values of input variables and response values at M training points, the true function $y(x)$

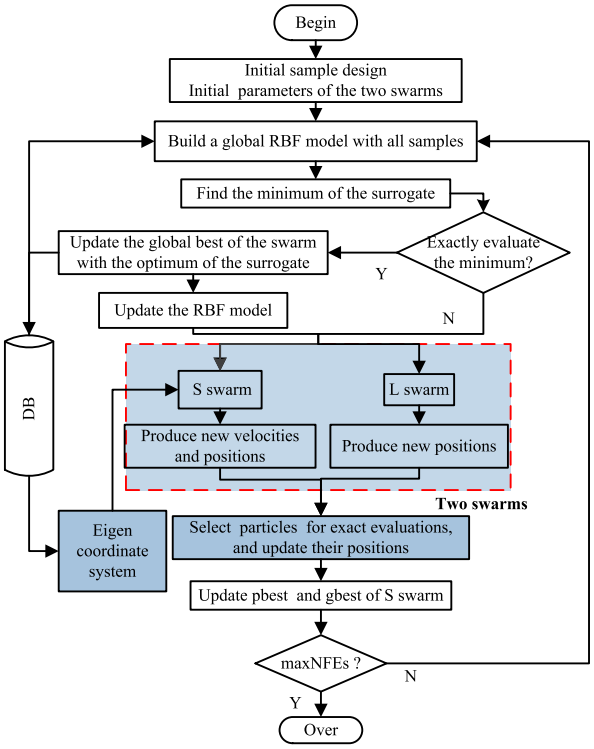

Fig. 2. Diagram of the overall algorithm.

can be approximated as

$$\hat{y}(x) = \sum_{i=1}^{M} \lambda_i \varphi(\|x - c_i\|) + p(x) \tag{8}$$

where λ are weight coefficients of basis functions. c_i is the ith center of the ith basis function. p is either a constant value or a polynomial model, and a linear polynomial is used in this article [21], [61]. φ is a basis function. As (8) is underdetermined, the orthogonality condition is further imposed on coefficients λ as

$$\sum_{i=1}^{M} \lambda_i p_j(x_i) = 0, \quad \text{for } j = 1, 2, \ldots, m \tag{9}$$

$$\begin{bmatrix} \Phi & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ b \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix} \tag{10}$$

where m is the number of terms of $p(x)$, $\Phi_{ij} = \varphi(\|x_i - x_j\|)$, $(i = 1, 2, \ldots, M)$, $(j = 1, 2, \ldots, M)$, $P_{ij} = p_j(x_i)$, $(i = 1, 2, \ldots, M)$, $(j = 1, 2, \ldots, m)$, $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_M]^T$, and $b = [b_1, b_2, \ldots, b_m]^T$. There are $(M + m)$ in (10) and its solution gives coefficients λ and b in (8).

III. Proposed SAMSO Algorithm

A. Overall Flowchart of the SAMSO

An SAMSO algorithm is described in this section. Fig. 2 shows the generic diagram of the proposed algorithm. Notable characteristics of the SAMSO algorithm include: two swarms are collaboratively used to enhance the population diversity of the algorithm. A dynamic swarm size adjustment strategy is proposed to balance global exploration and local exploitation. The commonly used fixed coordinate system is not well suitable for problems with different landscapes, so the eigencoordinate system is used to generate promising positions for the SPSO. In addition, a novel prescreening criterion is proposed to select promising particles for exact FEs.

Algorithm 1 Pseudocode of the SAMSO Algorithm

Input: Dimension of parameter space (d); real functions (f); the number of initial sample points (k); the number of maximum FEs ($maxNFEs$); the swarm size (N);
The threshold for determining whether the sample is exactly evaluated by f (η);
Output: The optimum of the swarm ($gbest$) and its fitness value (f_{gbest})

```
Initial sample design: the initial samples are generated by LHS; evaluate the initial samples with f; save the samples into the DB
Initial population design: Select N sample points from the DB; determine initial velocities v and initial positions and fitness values (pop, f_pop) for all particles; personal best of each particle (pbest, f_pbest); the global best of the swarm (gbest, f_gbest);
While NFEs<maxNFEs
    Use all samples to establish an RBF model
    Find the minimum of the surrogate: xmin_RBF;
    Calculate the minimum distance between xmin_RBF and other samples in the DB: dx_RBF
    if dx_RBF > η
        Exactly evaluate xmin_RBF and add it into the DB
        Update the RBF model if the DB is updated
        Update the global best of the whole swarm
    end if
    Using the two swarms to produce new particles
    Select particles for exact FEs and save new samples to the DB
    Update the L swarm
    Update the personal best of each particle in the S swarm
    Update the global best of the S swarm
End while
```

Algorithm 1 shows the pseudocode of the SAMSO. Initially, Latin hypercube sampling (LHS) is applied to initialize the database (DB) with k solutions, and N solutions with the best fitness values are selected as the initial positions of the whole swarm. Then, an RBF model is established by using all the samples. The Fmincon solver with the interior point method in MATLAB R2015a is used to find the optimum ($x\text{min}_{\text{RBF}}$) of the RBF model, and the global best of the swarm is selected as the start point. $x\text{min}_{\text{RBF}}$ will be exactly evaluated if the minimum distance between $x\text{min}_{\text{RBF}}$ and other evaluated samples in the DB is larger than the threshold η. The RBF model is also updated when a new sample is added to the DB. The exactly evaluated $x\text{min}_{\text{RBF}}$ will replace the global best of the swarm if it is better. Then, the two swarms will evolve sequentially. Next, a novel prescreening criterion is used to select particles for exact FEs, and the positions of these particles are updated. Finally, the personal best positions and the global best position of the S swarm are updated. Three kinds of solutions will be stored in the DB. These solutions include the initial sample points, the optimum of the RBF model and the evaluated particles at each iteration. The last two kinds of solutions will be cumulatively stored in DB at each iteration.

Although SAMSO and SA-COSO [26] algorithms both use the multiswarm strategy, there are great differences between the two algorithms. First, the types of swarms are different:

SA-COSO uses a PSO and an SL-PSO, while SAMSO uses an SPSO and a TLBO. Second, the collaboration between the two swarms is different: the better global best of the SL-PSO will update the global best of the PSO in SA-COSO, while the better global best of the TLBO will update the global best of the SPSO and the individual in the TLBO can also learn from particles from the SPSO in SAMSO. Third, methods of the model management are different: in SA-COSO, a fitness estimation strategy (FES) and an RBF network are used to pre-evaluate particles, and the particle will be exactly evaluated if its estimated fitness value is better than its personal best, or its estimated fitness value has a large degree of uncertainty. In the SAMSO, an RBF model is used to pre-evaluate particles, and a novel prescreening criterion is used to select particles for exact FEs.

B. Collaboration Between the Two Swarms

In the SPSO, the total exploratory space of the particle swarm is relatively small as velocities of the particles are only influenced by their personal best and the best particle in the swarm. This may cause the SPSO algorithm to converge prematurely in solving high-dimensional problems. Compared with the SPSO algorithm, the TLBO algorithm has a better global exploratory ability. The updating rules of the teacher phase are somewhat similar to the rules in the SPSO. All individuals learn from the teacher in the TLBO while all particles learn from the global best of the swarm and their own personal historical best in the SPSO. Both algorithms push all individuals toward a better position, which damages the exploration of the algorithms to some degree. However, a learner phase is followed in the TLBO so that individuals can learn from any other individuals in the population except themselves. The learner phase increases the exploratory space and slows down the convergence rate.

In this article, two swarms are used to efficiently solve high-dimensional computationally expensive problems. The first swarm evolves by following the rules of the learner phase in the TLBO (L swarm) and the second one evolves by following the rules of the SPSO (S swarm), inspired by the good global exploratory ability of the TLBO and the fast exploitative ability of the SPSO. These two swarms can learn from each other to promote information exchange. For each particle in the L swarm, a learning particle is randomly selected from the whole swarm ($L+S$ swarm), then it learns from the learning particle to produce a new offspring. Therefore, the particle in the L swarm not only can learn from particles in the L swarm but also learn from particles in the S swarm. The S swarm can also learn from the L swarm as the global best is taken as the best particle in the $S+L$ swarm. The L swarm can learn from any particle in the $S+L$ swarm, which increases the exploration of each individual and slows down the convergence rate of the whole swarm. The S swarm can guarantee the whole evolutionary speed.

The sizes of these two swarms can influence the global exploration and local exploitation of the algorithm. In this article, their swarm sizes are dynamically adjusted to control the evolutionary process. Generally, more exploration is needed in the initial stage of the evolution, then exploration gradually decreases and exploitation increases during the evolution. The swarm size of the L swarm decreases gradually with the iteration as the L swarm has greater exploratory ability. The size of the S swarm increases correspondingly as the total size of the $S+L$ swarm is fixed. The sizes of the two swarms are changing by following:

$$N_2 = \min(N, 2 + \lceil N((\max \text{NFEs-NFEs})/\max \text{NFEs})^{wn})\rceil) \quad (11)$$

$$N_1 + N_2 = N \quad (12)$$

where N_1 is the size of the S swarm. N_2 is the size of the L swarm. N is the total size of the two swarms. NFEs is the consumed NFEs. maxNFEs is the number of maximum allowed FEs. w_n is a parameter to control the rate of change.

The termination condition for the algorithm is that the number of consumed FEs reaches a specific value, so the number of consumed FEs is used to control the evolutionary progress. Considering that the increment of FEs is uncertain at each iteration, we study the variation curve of N_2 by assuming that the number of FEs increases only a unit value at each cycle. The real change of N_2 can be roughly reflected by the assumption. The relative curve of N_2 and the NFEs is shown in Fig. 3 by taking w_n as 0.2, 0.5, 1, 2, and 4, respectively.

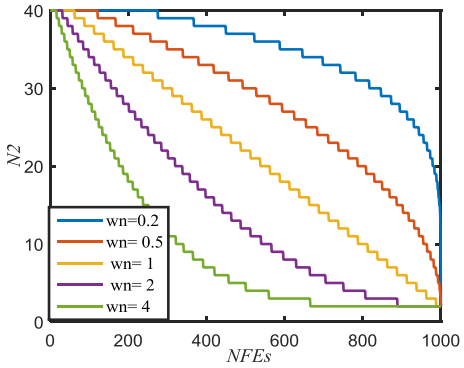

Fig. 3. Different relative curves of N_2 and NFEs in different control parameters w_n.

As shown in Fig. 3, N_2 decreases slowly in the initial stage and drops quickly in the later period when $w_n < 1$. The smaller w_n is, the slower the curve declines. The curve has a near uniform descent rate when $w_n = 1$, and it changes with the opposite trend when $w_n > 1$. More exploration and slower convergence rate are simultaneously obtained with a slower change of N_2. More exploration is better for complex problems when the total FEs are unlimited, so a smaller w_n is preferred in such cases. However, consuming fewer FEs is preferred in dealing with expensive problems, so how to select w_n is crucial to the performance of SAMSO. The performance of the proposed algorithm under different w_n is analyzed in the experimental part.

Fig. 4 shows the collaboration between the two swarms, and the pseudocode of producing offspring for the L and S swarms is shown in Algorithm 2. First, the sizes of the two swarms are dynamically adjusted according to (11) and (12). Then, the first N_1 particles evolve by following the rules of the SPSO, and the remaining particles evolve by following rules

Algorithm 2 Pseudocode of Producing Offspring for the L and S Swarm

```
Dynamically adjust the number of the two swarms N1, N2 according to (11) and (12).
for i = 1 : N do
  if i ≤ N1 (particle i in the S swarm)
    Produce new velocities by using Algorithm 3 and new positions by using (2): new_vi, new_popi;
    vi = new_vi, popi = new_popi;
  end if
  if i > N1 (particle i in the L swarm)
    Produce new positions by using (4) and (5): new_popi;
  end if
end for
```

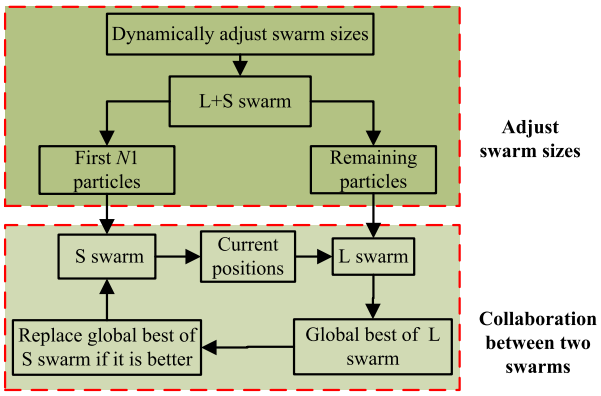

Fig. 4. Collaboration between the two swarms.

of the learner phase in the TLBO. In the (t)th iteration, particles in the L swarm not only can learn from current positions of their own swarm but can learn from the current positions of the S swarm to produce new positions in the $(t+1)$th iteration. In addition, the global best of the L swarm will replace the global best of the S swarm if it is better. Particles in the S swarm learn from the new global best to update their velocities.

C. Two Coordinate Systems for S Swarm

The commonly used coordinate system in the SPSO is fixed and not well suitable for various fitness landscapes. This seriously affects the generalization ability of the SPSO to different problems. However, if the information about a fitness landscape is known, the problem will be easier to be solved as the algorithm can adaptively adjust its searching direction. The population distribution information can indicate the features of a fitness landscape to some degree. In [53], this information is used to establish an eigencoordinate system, and new promising individuals are produced according to the new coordinate system. The original and eigencoordinate systems are both used to provide promising evolutionary directions for the population in [59], and this strategy can enhance exploratory space of the population as shown in Fig. 1. The same method is used in this article to produce new positions for the S swarm. If a random number between 0 and 1 is smaller than Pr, the eigencoordinate system is used, otherwise, the original coordinate system is used. Pr $= 0.5$ is adopted as recommended in [59].

Algorithm 3 Pseudocode of Producing New Positions for the S Swarm

```
Select 2d samples with the smallest function values from the DB
Construct a covariance matrix C by using the 2d samples
Eigen decomposition of C, and obtain orthogonal matrix B
For each particle in the S swarm do
  if rand< Pr
    Produce new velocity by using (6)
  Else
    Produce new velocity by using (7)
  End
End for
```

The surrogate model can approximate landscapes of the problems to some degree, and the exactly evaluated samples are selected by using the information of the model. Such samples may be able to indicate the fitness landscape. Therefore, some promising samples are used to establish the eigencoordinate system in this article. The covariance matrix is constructed by using $2d$ samples with the smallest function values from the DB as $2d$ samples are enough to build the covariance matrix, and selecting too many samples will increase the complexity of the algorithm. The pseudocode of producing new positions for the S swarm is described in Algorithm 3.

D. Model Management Strategy

For computationally expensive problems, one real FE may consume plenty of time. Therefore, algorithms should find good solutions with limited FEs in solving expensive problems. Some surrogate-assisted prescreening strategies have been proposed to select promising individuals for real FEs to save computational resources. However, how to design the prescreening strategy is difficult. If a prescreening strategy selects too many individuals or some unimportant individuals for FEs, it may make little contribution for reducing the overall FEs. The surrogate model can roughly approximate the fitness values of the original problem. By utilizing the predicted fitness values, we proposed the strategy in Algorithm 4 for selecting particles for exact FEs in this article. The strategy was designed to reduce some unimportant FEs, and leave more FEs for the particle to search the problem space.

In the S swarm, particles with worse fitness values than their historical optima will make little influence on the evolutionary progress as the evolutionary rules of the S swarm are influenced by the global best particle and historical optima of the particles. In the L swarm, particles with worse fitness values than their current fitness values will be discarded. In both S and L swarms, only particles with better fitness values than their current fitness values can influence their produced new positions in the next generation. Inspired by this, all particles are divided into two categories according to whether they have the capacity of self-improvement. The particle predicted to be with self-improvement is exactly evaluated as it has a bigger

Algorithm 4 Pseudocode of Selecting Particles for Exact FEs

```
Predict the fitness values of the new positions (new_pop) by using the RBF model: pre
for i = 1 : N do
Calculate the minimum distance between new_pop_i and other samples in the DB: dx
  if dx > η & pre_i < fpop_i
    Exactly evaluate new_pop_i: fnewpop_i = f(new_pop_i)
    Add the new sample to the DB; NFEs = NFEs + 1
    if fnewpop_i < fpop_i
      pop_i = new_pop_i; fpop_i = fnewpop_i
      Update the personal best of the particle
      Update the global best of the swarm
    end if
  end if
end for
if no new position is exactly evaluated
  Find the new_pop_i with the minimum pre
  Exactly Evaluate new_pop_i: fnewpop_i = f(new_pop_i)
  Add the new sample to DB; NFEs = NFEs + 1
    if fnewpop_i < fpop_i
      pop_i = new_pop_i; fpop_i = fnewpop_i
      Update the personal best of the particle i
      Update the global best of the swarm
    end if
End if
```

probability to improve its current fitness value. Contrarily, the particle predicted to be without self-improvement is not exactly evaluated as it has a small probability to improve its current fitness value.

In the t-th generation, an RBF surrogate model is first used to predict the fitness values of all particles, and the fitness value estimated by the RBF model is denoted as $\hat{f}$. If the predicted fitness value of the particle i is smaller than its fitness value in the $(t-1)$th generation and the minimum distance between the particle and other evaluated samples is bigger than the threshold η, the particle i will be exactly evaluated. All exactly evaluated particles will be put in set I^t as shown in (13). Meanwhile, the particle without exact FE will inherit its fitness value in the $(t-1)$th generation as shown in (14), which seems like a special fitness inheritance strategy in genetic algorithms. This strategy will promote the particle to the position better than its recently evaluated position. If no particle is exactly evaluated, the particle with the minimum prediction will be exactly evaluated to push the progression of evolution

$$I^t = \left\{ i : \hat{f}\left(x_i^t\right) < f\left(x_i^{t-1}\right) \wedge \min_{x \in DB} \operatorname{dist}\left(x_i^t, x\right) > \eta \right\} \quad (13)$$

$$f\left(x_i^t\right) = f\left(x_i^{t-1}\right) \text{ if } i \notin I^t. \quad (14)$$

Moreover, the optimum of the surrogate model is also exactly evaluated, and it will replace the global best of the S swarm if it is better. This method has been efficiently used to guide the searching direction of the particle swarm [21], [27], [28], but it may attract the swarm to the optimum of the surrogate. The optimum of the surrogate is usually not the real optimum of the problem. However, the L swarm does not learn from the global optimum of the surrogate, so the exploratory ability of the algorithm will not be affected by this strategy.

In [19], [22], and [34], the particle whose predicted value is better than its historical optimum will be exactly evaluated. The distinction between the two methods is that more exploration is allowed in the proposed strategy. The particle inferior to its personal best may be located in the not-well-explored area, so evaluating it may find a good solution [62] and can also improve the accuracy of the surrogate model.

IV. Numerical Experiment Study

In this section, a set of commonly used benchmark problems [23], [34], [63], [64] shown in Table I is used to validate the proposed algorithm. First, the behavior study of the proposed algorithm is conducted. Then, the proposed SAMSO algorithm is compared with three state-of-the-art SAEAs. Finally, the effects of some important parameters on the SAMSO algorithm are studied in the supplemental material.

A. Parameter Settings

During the experiment, 20 independent runs are performed for each algorithm. For all algorithms tested in this article, the termination condition is that the number of consumed FEs is less than maxNFEs which is set at 1000. For the SAMSO, the parameters are set as: $w = 0.792$, $c_1 = c_2 = 1.491$ and $V\max = 0.5(ub - lb)$, $V\min = -0.5V\max$. lb is the lower boundary of variables and ub is the upper boundary. These settings refer to [1] and [66]. η is min[sqrt($0.001^2 d$), 5.0e-$4d \times \min(ub - lb)$] to prevent the samples being too close. d is the dimension of the problem. The searching space is increasing with the increase of the dimension of a problem, so the swarm size N is set as 40 when $d \leq 50$, and N is 80 when $d > 50$. We investigate the effects of the total swarm size on the proposed algorithm in the supplementary material. The experimental results show the rationality of the setup of the parameters. The number of initial samples k is N when $d \leq 50$, and k is $2d$ when $d > 50$. The high-dimensional space is hard to be approximated, so the number of initial samples is bigger. $w_{nc} = 1$ and $\Pr = 0.5$ are used in the proposed SAMSO algorithm. We investigate the effects of the parameter w_{nc} and Pr on the proposed algorithm in the supplementary material. The experimental results show the rationality of the setup of the two parameters. In addition, we also exchange the setting of the size of the two swarms in the supplementary material and name this algorithm as SAMSO-Exc. The experimental results indicate that SAMSO-Exc shows worse performance than SAMSO on most of the problems. In the proposed SAMSO algorithm, the RBF interpolation model with a commonly used cubic basis function is used to approximate the global fitness landscape as the basis function shows good performance in other SAEAs [21], [61].

B. Behavior Study of the SAMSO

1) Effects of the Multiswarm Strategy: In this section, two 50-D multimodal functions (Rastrigin and F10) are used to study the effects of the multiswarm strategy. We name the SAMSO algorithm without using the eigencoordinate system as SAMSO1. Then, we compare the SAMSO1

TABLE I
BENCHMARK FUNCTIONS USED IN THE EXPERIMENT STUDY

Benchmark Problem	d	Global Optimum	Characteristics
Ellipsoid	30, 50, 100, 200	0	Unimodal
Rosenbrock	30, 50, 100, 200	0	Multimodal with narrow valley
Rastrigin	30, 50, 100,	0	Multimodal
Ackley	30, 50, 100, 200	0	Multimodal
Griewank	30, 50, 100, 200	0	Multimodal
F10 in CEC 2005 [65]	30, 50, 100, 200	-330	Shifted Rotated Rastrigin Very complicated multimodal
F16 in CEC 2005 [65]	30, 50, 100	120	Rotated Hybrid composition function Very complicated multimodal
F19 in CEC 2005 [65]	30, 50, 100, 200	10	Rotated Hybrid composition function Very complicated multimodal

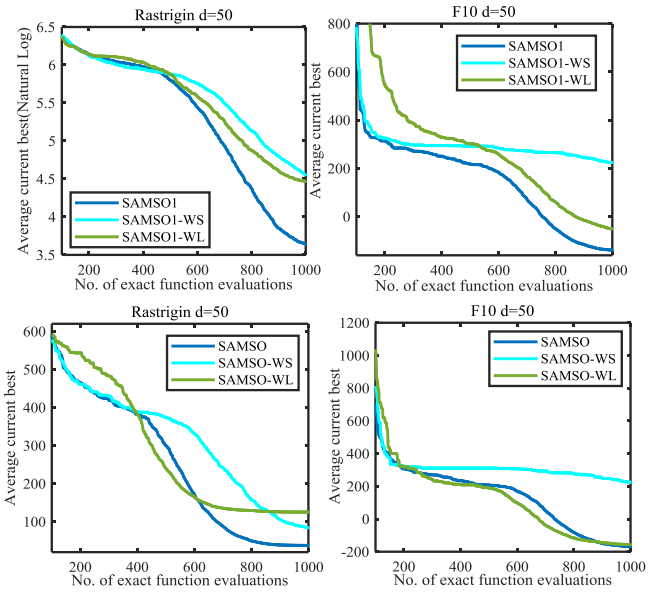

Fig. 5. Effects of the multiswarm strategy.

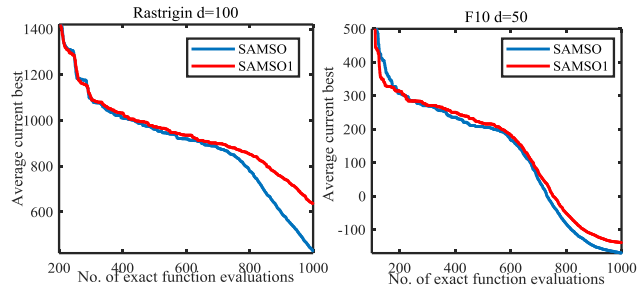

Fig. 6. Effects of the eigencoordinate system.

algorithm with SAMSO1-WS (SAMSO1 without S swarm) and SAMSO1-WL (SAMSO1 without L swarm) to investigate whether the strategy of using multiswarm is superior to the strategy of using one swarm. We also compare SAMSO algorithm with SAMSO-WS (SAMSO without S swarm) and SAMSO-WL (SAMSO without L swarm) to investigate whether the strategy of using multiswarm is superior to the strategy of using one swarm. Convergence profiles of the six algorithms on the two functions are shown in Fig. 5. The L swarm has better exploration ability than the S swarm. Algorithms with using the L swarm can obtain better results than algorithms which only use the S swarm in the early stage. Therefore, SAMSO and SAMSO1 can obtain good results in the early stage. The S swarm is superior in exploiting the promising area, so algorithms with using the S swarm can obtain good results in the later stage. The worse performance of SAMSO-WL than SAMSO-WS maybe that SAMSO-WL is trapped in local optima. The strategy of using two swarms shows superior performance than the strategies of only using one swarm on Rastrigin no matter whether the eigencoordinate system is used. For F10, SAMSO1 obtains significantly better results than SAMSO1 with only one swarm, and SAMSO obtains significantly better results than SAMSO-WS and similar results with SAMSO-WL. The strategy of using two swarms combines the advantage of the two swarm as SAMSO has good exploration ability in the early stage and good exploitation ability in the later stage. These findings all indicate the effectiveness of the multiswarm strategy.

In the proposed algorithm, the L swarm learns from the S swarm by randomly selecting a particle in the whole swarm to learn. The strategy can enhance the exploratory ability of the L swarm compared with the method that the particle in the L swarm only learns from particles in its own swarm. To validate this hypothesis, we changed SAMSO by making particles in the L swarm only learn from particles in its own swarm, and called the algorithm as SAMSO-L1. We compare the two algorithms in the supplemental material. The experimental results indicate that the strategy of making the L swarm learn from the S swarm is better.

2) Effects of the Eigencoordinate System: In this section, two functions are used to study effects of the eigencoordinate system. Using the original coordinate system and the eigencoordinate system can improve the performance of the algorithm to different problems and enhances the exploration ability. SAMSO shows better performance on a rotational function (50-D F10) and a high-dimensional function (100-D Rastrigin) as shown in Fig. 6. The eigencoordinate system is used in the S swarm, and the S swarm plays the main role in the later stage. The convergence curves of SAMSO and SAMSO1 are nearly similar in the early stage. However, the SAMSO algorithm can find better solutions in the later stage. These indicate the effectiveness of using the eigencoordinate system.

In the proposed algorithm, the S swarm evolves with two coordinate systems (original and eigencoordinate systems) in

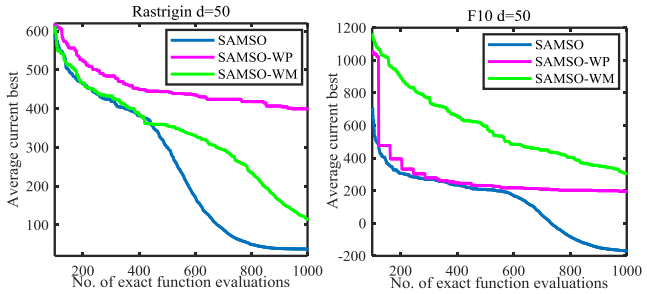

Fig. 7. Effects of the prescreening strategy and the optimum of the model.

order to enhance the generalization ability of the S swarm to different problems. This strategy can also enhance the exploratory ability and provide promising directions for the S swarm. The L swarm has the good exploratory ability, so the eigencoordinate system is not used with the L swarm. To validate this hypothesis, we change SAMSO by making the L swarm also evolve with two coordinate systems, and we name the algorithm as SAMSO-L2. We compare the two algorithms in the supplemental material. The similar performance of the two algorithms on most of the problems indicates that the two coordinate systems do not have significant effects on the L swarm. The L swarm is mainly used to explore the searching space. The searchability of the L swarm is already good for exploration. Therefore, the performance of the algorithm is not significantly improved even that the two coordinate systems are used on the L swarm

3) Effects of the Prescreening Strategy and the Optimum of the Model: In this section, two 50-D multimodal functions (Rastrigin and F10) are used to study the prescreening strategy and the optimum of the model. We name SAMSO without the prescreening strategy as SAMSO-WP, SAMSO without using the optimum of the model as SAMSO-WM. Convergence profiles of the three algorithms on the two functions are shown in Fig. 7. SAMSO obtains the best results on the two functions. SAMSO-WP converges slower on the two functions as many FEs are consumed on some unimportant individuals at each iteration. These results indicate the importance of using the prescreening criterion. SAMSO-WM also converges slower on the two functions, and it achieves worse results on 50-D F10. F10 with a flatter basin than Rastrigin may be easier to be approximated, and the optimum of the surrogate model can provide real promising directions for the swarm, so the optimum of the surrogate has a greater impact on F10.

Overall, the strategy of using both the L swarm and the S swarm performs better than the strategy of using only one swarm. It combines the good exploration ability of the L swarm and the good exploitation ability of the S swarm. The strategy of using the eigencoordinate system can further improve the performance of the S swarm. In addition, the prescreening strategy can significantly reduce some unimportant FEs, and the optimum of the model can guide the swarm to promising areas. All of the four strategies play a positive role in the proposed SAMSO algorithm. The multiswarm strategy, the prescreening strategy, and the strategy of using the optima of the surrogate paly the main role in the proposed algorithm as shown in Figs. 5 and 7. The strategy of using the eigencoordinate system plays a slightly weaker role.

C. Comparative Experiments on Benchmark Problems

To further test the efficiency of the proposed algorithm, the proposed algorithm is compared with three state-of-the-art SAPSOs: SHPSO [34], SA-COSO [26], and EASO [35] on 30-, 50-, 100-, and 200-D problems as shown in Table I. SHPSO uses an SPSO algorithm and an SL-PSO algorithm to explore and exploit the search space, and it uses the strategy that the particle with a better predicted fitness value than its historical personal best will be selected for exact FEs. SA-COSO uses two SAPSOs collaboratively to search for the global optimum, and two swarms are used. In ESAO, global and local surrogate models are collaboratively used to select the best offspring produced by the differential evolution and the optimal solution of the local surrogate model for exact FEs. The three algorithms are recently published promising algorithms for high-dimensional expensive problems.

1) Experimental Results on 30-, 50- and 100-D Problems: The average best fitness values obtained by the four algorithms over 20 independent runs are shown in Table II. The results of EASO are copied from the original paper. The best mean results of individual test instances are highlighted. Results of Wilcoxon rank-sum test calculated at a significance level of $\alpha = 0.05$ are also listed in Table II, where "$\approx$" indicates that there is no statistically significant difference between the results obtained by SAMSO and the compared algorithm, "$+$" indicates that SAMSO is significantly better than the compared algorithm, while "$-$" means that SAMSO is significantly outperformed by the compared algorithms. As shown in Table II, SAMSO algorithm has achieved the best mean values in 17 out of 24 problems and achieved the second-best mean values in the remaining seven problems. EASO and SHPSO achieve the best mean values in four and three problems respectively.

SAMSO obtains better mean values than EASO on 14 out of 18 problems. EASO uses a global surrogate assisted differential evolution for exploration and a local surrogate model for exploitation. The best individual in the differential evolution and the optimum of the local surrogate model are exactly evaluated. Generally, the differential evolution has a better global exploratory ability than the SPSO. EASO uses the differential evolution to produce candidates, which could promote the global search toward unexplored regions. The better results achieved by SAMSO on the complex multimodal problems, such as F10 and F19 may be that the strategy of using two swarms enhances the exploratory ability of SAMSO. However, EASO achieves slightly better mean values on 30- and 50-D Rosenbrock, 50-D Ackley, and 100-D F10. The local surrogate model in EASO can exploit the promising area well, which may be the reason that it shows slightly better performance on the four problems.

Both SA-COSO and SAMSO use two swarms to explore and exploit the landscape of problems. In SA-COSO, an RBF-assisted SL-PSO is used to explore the global optimum, and an FES-assisted PSO is used to enhance local searches. In SAMSO, a learner phase of TLBO is used to enhance

TABLE II
AVERAGE BEST FITNESS VALUES OBTAINED BY THE COMPARED ALGORITHMS ON BENCHMARK FUNCTIONS, INCLUDING THE AVERAGE FITNESS VALUE AND STANDARD DEVIATION SHOWN AS AVG (STD)

Functions	d	EASO	SA-COSO	Wilcoxon test	SHPSO	Wilcoxon test	SAMSO
Ellipsoid	30	2.75E-02(6.96E-02)	3.85E+00(1.19E+00)	+	4.38E-01(2.02E-01)	+	**5.30E-03**(5.76E-03)
	50	7.40E-01(5.55E-01)	4.66E+01(1.74E+01)	+	7.17E+00 (2.42E+00)	+	**5.13E-01**(2.85E-01)
	100	1.28E+03(1.34E+02)	9.85E+02(2.14E+02)	+	1.23E+02(2.36E+01)	+	**7.21E+01**(1.78E+01)
Rosenbrock	30	**2.50E+01**(1.57E+00)	5.99E+01 (2.43E+01)	+	2.86E+01(4.85E-01)	≈	2.83E+01(8.54E-01)
	50	**4.74E+01**(1.71E+00)	2.53E+02(5.67E+01)	+	5.13E+01 (2.00E+00)	≈	5.01E+01(7.68E-01)
	100	5.79E+02(4.48E+01)	2.50E+03(9.74E+02)	+	**2.09E+02(**5.36E+01**)**	**-**	2.86E+02(5.25E+01)
Rastrigin	30	N/A	1.38E+02(3.62E+01)	+	1.82E+02(3.30E+01)	+	**2.62E+01**(1.33E+01)
	50	N/A	3.22E+02(3.83E+01)	+	3.89E+02(6.27E+01)	+	**3.77E+01**(7.82E+00)
	100	N/A	8.81E+02(7.01E+01)	+	8.78E+02 (8.93E+01)	+	**4.29E+02**(5.72E+01)
Ackley	30	2.52E+00(8.40E-01)	5.01E+00(1.22E+00)	+	1.81E+00(3.75E-01)	+	**6.28E-01**(5.42E-01)
	50	**1.43E+00**(2.49E-01)	8.86E+00(1.10E+00)	+	2.60E+00(2.48E-01)	+	1.53E+00(4.36E-01)
	100	1.04E+01(2.11E-01)	1.59E+01(5.14E-01)	+	**5.48E+00** (8.99E-01)	**-**	6.12E+00(4.09E-01)
Griewank	30	9.53E-01(5.04E-02)	1.44E+00(1.80E-01)	+	9.17E-01(8.73E-02)	+	**5.38E-01**(1.44E-01)
	50	9.40E-01(4.21E-02)	5.63E+00(8.92E-01)	+	9.45E-01(5.39E-02)	+	**6.66E-01**(1.07E-01)
	100	5.73E+01(5.84E+00)	6.35E+01(1.49E+01)	+	1.12E+00(3.67E-02)	+	**1.06E+00**(2.64E-02)
F10	30	6.33E+00(2.65E+01)	-5.74E+01(1.75E+01)	+	-9.13E+01(3.28E+01)	+	**-2.39E+02(**2.43E+01**)**
	50	1.99E+02(4.58E+01)	2.35E+02(4.09E+01)	+	1.22E+02(2.59E+01)	+	**-1.69E+02**(3.17E+01)
	100	**7.13E+02**(2.65E+01)	1.42E+03(1.23E+02)	+	7.78E+02(7.17E+01)	+	7.37E+02(4.20E+01)
F16	30	N/A	5.28E+02(9.48E+01)	+	4.58E+02(6.49E+01)	+	**3.72E+02**(1.47E+02)
	50	N/A	6.13E+02(3.74E+01)	+	4.78E+02(3.96E+01)	+	**3.26E+02**(9.86E+01)
	100	N/A	8.07E+02(6.57E+01)	+	**5.08E+02**(2.15E+01)	≈	5.13E+02(1.85E+01)
F19	30	9.32E+02(8.94E+00)	9.69E+02(2.43E+01)	+	9.43E+02(9.60E+00)	+	**9.22E+02**(3.66E+00)
	50	9.75E+02(3.71E+01)	1.08E+03(3.66E+01)	+	1.00E+03(2.12E+01)	+	**9.70E+02**(2.92E+01)
	100	1.37E+03(2.75E+01)	1.41E+03(2.28E+01)	+	1.43E+03(3.57E+01)	+	**1.29E+03**(3.34E+01)
+/≈/-		N/A	24/0/0		19/3/2		**N/A**

exploration. SAMSO significantly outperforms SA-COSO on all 24 problems. Both the SL-PSO and the TLBO are powerful global search algorithms. The good results obtained by SAMSO may be that the strategy of dynamically adjusting swarm sizes can balance the exploration and exploitation of SAMSO.

SAMSO significantly outperforms SHPSO on 19 out of 24 problems, and they achieve similar results on 30- and 50-D Rosenbrock, and 100-D F16. SHPSO significantly outperforms SAMSO on 100-D Rosenbrock and Ackley. In SHPSO, the RBF model is built by using some samples with the best fitness values in the archive so that it can achieve good local exploitation. This strategy may make SHPSO show good results on Rosenbrock and Ackley. However, SAMSO can also obtain good results on unimodal or simple multimodal functions, such as Ellipsoid, Ackley, and Griewank. Although the SPSO is used both in SHPSO and SAMSO, the multiswarm strategy in SAMSO can significantly enhance the exploration of it, so SAMSO significantly outperforms SHPSO on complex multimodal functions, such as Rastrigin, F10, F16, and F19.

Overall, SAMSO algorithm can find better solutions on both unimodal and multimodal problems compared with three state-of-the-art algorithms. All of those indicate SAMSO shows good generalization ability to different types of problems.

The comparison of the convergence history of the compared algorithms is shown in the supplemental material. SAMSO can find better solutions in a short time on most of the functions compared with other algorithms, and it continuously improves the optima in the later stages of the iteration.

2) Experimental Results on 200-D Problems: To further test the performance of SAMSO on higher dimensional problems, we compare SAMSO with EASO and SA-COSO on six 200-D problems in the supplemental material. These results confirm the good performance of the proposed method for solving high-dimensional problems.

D. Computational Complexity of SAMSO Algorithm

In this section, we present an analytical study on the computational complexity of the SAMSO algorithm. We refer to the method in [20] to do the analysis. The computational complexity of SAMSO is determined by the computational time for fitness evaluations, for training the RBF model, for finding

the optima of RBF model, and other additional costs, such as for producing new population, selecting promising particles for real FEs and updating the personal best and global best of the swarm, which are often negligible. The computational effort (T) of SAMSO is formulated as follows:

$$T = 1000 \cdot F + \sum_{i=1}^{m_1} \left(2 \bullet T^i_{\text{RBF}} + T^i_{\text{sRBF}} + T^i_{\text{ovwehead}}\right)$$

where:

F	original FE cost;
T^i_{RBF}	time to build the RBF model;
T^i_{sRBF}	time to find the optimum of the RBF model;
T^i_{overhead}	other additional costs, such as for producing new population, selecting promising particles for real FEs and updating the personal best and global best of the swarm, which are often negligible;
m_1	total iterations of the algorithm.

The computational complexity of training the RBF model is $O(n^3)$. The computational complexity of the algorithm to find the optimum of the model is $O(d^3)$. n is the number of samples used to construct the models, and d is the dimension of the problems. The computational complexity of SAMSO algorithm is $O(n^3)$ as n is usually bigger than d. It is worth noting that FEs of expensive problems are quite time consuming, so the most significant part contributing to the total computational effort incurred is F. Hence, T_{RBF}, T_{sRBF}, and T_{overhead} are generally considered to be negligible, otherwise, such algorithm should never be used.

V. Conclusion

In this article, a SAMSO algorithm was proposed to efficiently solve high-dimensional computationally expensive optimization problems. Inspired by the fact that multiswarm can efficiently improve the global exploratory ability of algorithms, two swarms are collaboratively used to explore and exploit the search space. A dynamic swarm size adjustment scheme was proposed to balance the sizes of the two swarms during the evolutionary progress. The strategy of using two swarms has more potential to explore the search space, while the strategy of using one swarm is easy to prematurely converge on some problems. Two coordinate systems are used to promote the SPSO toward promising areas, which can enhance the quality of solutions on some problems. A new prescreening strategy based on self-improvement is proposed to select promising individuals for exact FEs, which can significantly improve solutions in the same number of FEs. Our empirical results confirm that the proposed algorithm is comparable to or better than the three state-of-the-art algorithms compared in this article, and it has a good self-adaption to problems with different dimensions.

Although the proposed SAMSO has shown superior performance on the eight high-dimensional test problems with their dimensions up to 200, some potential directions are worthy of further investigation. First, the sizes of the two swarms are dynamically adjusted according to the number of consumed FEs. However, it is better to adaptively adjust the parameter according to the evolutionary state and the number of allowed FEs. Second, only two swarms are used in this article. It would be valuable to investigate strategies of using more swarms to explore and exploit the search space. Third, it would also be very interesting to test the proposed algorithm in real-world application problems. In addition, the RBF model is used to approximate the fitness values of problems in this article. There might be a situation where the RBF model provides a bad estimation of the real fitness value. This may cause a bad influence on the algorithm as the prescreening strategy does not work. Therefore, finding some surrogate models which are suitable for high-dimensional expensive problems or have a good adaptation to different problems is also meaningful.

References

[1] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Math. Probl. Eng.*, vol. 2015, pp. 1–38, Oct. 2015.

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

[3] B. Zhang, Q. Pan, L. Gao, X.-Y. Li, L.-L. Meng, and K.-K. Peng, "A multiobjective evolutionary algorithm based on decomposition for hybrid flowshop green scheduling problem," *Comput. Ind. Eng.*, vol. 136, pp. 325–344, Oct. 2019.

[4] C. He, Y. Tian, H. Wang, and Y. Jin, "A repository of real-world datasets for data-driven evolutionary multiobjective optimization," *Complex Intell. Syst.*, pp. 1–9, Nov. 2019.

[5] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 442–458, Jun. 2019.

[6] R. L. Hardy, "Multiquadric equations of topography and other irregular surfaces," *J. Geophys. Res.*, vol. 76, no. 8, pp. 1905–1915, 1971.

[7] H.-M. Gutmann, "A radial basis function method for global optimization," *J. Glob. Optim.*, vol. 19, no. 3, pp. 201–227, 2001.

[8] T. Goel, R. T. Hafkta, and W. Shyy, "Comparing error estimation measures for polynomial and Kriging approximation of noise-free functions," *Struct. Multidiscipl. Optim.*, vol. 38, no. 5, pp. 429–442, 2008.

[9] S.-C. Horng and S.-Y. Lin, "Evolutionary algorithm assisted by surrogate model in the framework of ordinal optimization and optimal computing budget allocation," *Inf. Sci.*, vol. 233, pp. 214–229, Jun. 2013.

[10] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Glob. Optim.*, vol. 13, no. 4, pp. 455–492, 1998.

[11] S. M. Clarke, J. H. Griebsch, and T. W. Simpson, "Analysis of support vector regression for approximation of complex engineering analyses," *J. Mech. Design*, vol. 127, no. 6, pp. 1077–1087, 2005.

[12] D. Zhao and D. Xue, "A comparative study of metamodeling methods considering sample quality merits," *Struct. Multidiscipl. Optim.*, vol. 42, no. 6, pp. 923–938, 2010.

[13] R. Jin, W. Chen, and T. W. Simpson, "Comparative studies of metamodelling techniques under multiple modelling criteria," *Struct. Multidiscipl. Optim.*, vol. 23, no. 1, pp. 1–13, 2000.

[14] T. Goel, R. T. Haftka, W. Shyy, and N. V. Queipo, "Ensemble of surrogates," *Struct. Multidiscipl. Optim.*, vol. 33, no. 3, pp. 199–216, 2006.

[15] J. Gu, G. Li, and N. Gan, "Hybrid metamodel-based design space management method for expensive problems," *Eng. Optim.*, vol. 49, no. 9, pp. 1573–1588, 2016.

[16] H. Dong, B. Song, P. Wang, and Z. Dong, "Hybrid surrogate-based optimization using space reduction (HSOSR) for expensive black-box functions," *Appl. Soft Comput.*, vol. 64, pp. 641–655, Mar. 2018.

[17] P. Ye, G. Pan, and Z. Dong, "Ensemble of surrogate based global optimization methods using hierarchical design space reduction," *Struct. Multidiscipl. Optim.*, vol. 58, no. 2, pp. 537–554, 2018.

[18] Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum, "Combining global and local surrogate models to accelerate evolutionary optimization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 1, pp. 66–76, Jan. 2007.

[19] Z. Yang, H. Qiu, L. Gao, X. Cai, C. Jiang, and L. Chen, "A surrogate-assisted particle swarm optimization algorithm based on efficient global optimization for expensive black-box problems," *Eng. Optim.*, vol. 51, no. 4, pp. 549–566, 2019.

[20] D. Lim, J. Yaochu, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 329–355, Jun. 2010.

[21] R. G. Regis, "Particle swarm with radial basis function surrogates for expensive black-box optimization," *J. Comput. Sci.*, vol. 5, no. 1, pp. 12–23, 2014.

[22] C. Sun, Y. Jin, J. Zeng, and Y. Yu, "A two-layer surrogate-assisted particle swarm optimization algorithm," *Soft Comput.*, vol. 19, no. 6, pp. 1461–1475, 2014.

[23] B. Liu, Q. Zhang, and G. G. E. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 180–192, Apr. 2014.

[24] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1012–1025, Mar. 2019.

[25] F. Li, X. Cai, and L. Gao, "Ensemble of surrogates assisted particle swarm optimization of medium scale expensive problems," *Appl. Soft Comput.*, vol. 74, pp. 291–305, Jan. 2019.

[26] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 644–660, Aug. 2017.

[27] Y. Tang, J. Chen, and J. Wei, "A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions," *Eng. Optim.*, vol. 45, no. 5, pp. 557–576, 2013.

[28] M. D. Parno, T. Hemker, and K. R. Fowler, "Applicability of surrogates to improve efficiency of particle swarm optimization for simulation-based problems," *Eng. Optim.*, vol. 44, no. 5, pp. 521–535, 2012.

[29] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 61–70, 2011.

[30] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Comput.*, vol. 9, no. 1, pp. 3–12, Jan. 2005.

[31] M. Tabatabaei, J. Hakanen, M. Hartikainen, K. Miettinen, and K. Sindhya, "A survey on handling computationally expensive multiobjective optimization problems using surrogates: Non-nature inspired methods," *Struct. Multidiscipl. Optim.*, vol. 52, no. 1, pp. 1–25, 2015.

[32] A. I. J. Forrester and A. J. Keane, "Recent advances in surrogate-based optimization," *Progr. Aerosp. Sci.*, vol. 45, nos. 1–3, pp. 50–79, 2009.

[33] A. Diaz-Manriquez, G. Toscano, J. H. Barron-Zambrano, and E. Tello-Leal, "A review of surrogate assisted multiobjective evolutionary algorithms," *Comput. Intell. Neurosci.*, vol. 2016, Jun. 2016, Art. no. 9420460.

[34] H. Yu, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Surrogate-assisted hierarchical particle swarm optimization," *Inf. Sci.*, vols. 454–455, pp. 59–72, Jul. 2018.

[35] X. Wang, G. G. Wang, B. Song, P. Wang, and Y. Wang, "A novel evolutionary sampling assisted optimization method for high dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 815–827, Oct. 2019.

[36] Z. Yang, H. Qiu, L. Gao, C. Jiang, and J. Zhang, "Two-layer adaptive surrogate-assisted evolutionary algorithm for high-dimensional computationally expensive problems," *J. Glob. Optim.*, vol. 74, no. 2, pp. 327–359, 2019.

[37] X. Cai, L. Gao, X. Li, and H. Qiu, "Surrogate-guided differential evolution algorithm for high dimensional expensive problems," *Swarm Evol. Comput.*, vol. 48, pp. 288–311, Aug. 2019.

[38] X. Cai, L. Gao, and X. Li, "Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, to be published.

[39] J. Tian, C. Sun, Y. Tan, and J. Zeng, "Granularity-based surrogate-assisted particle swarm optimization for high-dimensional expensive optimization," *Knowl. Based Syst.*, vol. 187, Jan. 2020, Art. no. 104815.

[40] C. Sun, J. Ding, J. Zeng, and Y. Jin, "A fitness approximation assisted competitive swarm optimizer for large scale expensive optimization problems," *Memetic Comput.*, vol. 10, no. 2, pp. 123–134, 2016.

[41] B. Pang, Z. Ren, Y. Liang, and A. Chen, "Enhancing cooperative coevolution for large scale optimization by adaptively constructing surrogate models," in *Proc. Genet. Evol. Comput. Conf. Companion*, Kyoto, Japan, 2018, pp. 221–222.

[42] H. Ma, S. Shen, M. Yu, Z. Yang, M. Fei, and H. Zhou, "Multi-population techniques in nature inspired optimization algorithms: A comprehensive survey," *Swarm Evol. Comput.*, vol. 44, pp. 365–387, Feb. 2019.

[43] F. Zou, D. Chen, and Q. Xu, "A survey of teaching–learning-based optimization," *Neurocomputing*, vol. 335, pp. 366–383, Mar. 2019.

[44] W. H. Lim and N. A. M. Isa, "Teaching and peer-learning particle swarm optimization," *Appl. Soft Comput.*, vol. 18, pp. 39–58, May 2014.

[45] W. H. Lim and N. A. M. Isa, "Bidirectional teaching and peer-learning particle swarm optimization," *Inf. Sci.*, vol. 280, pp. 111–134, Oct. 2014.

[46] T. Cheng, M. Chen, P. J. Fleming, Z. Yang, and S. Gan, "A novel hybrid teaching learning based multi-objective particle swarm optimization," *Neurocomputing*, vol. 222, pp. 11–25, Jan. 2017.

[47] W. H. Lim and N. A. M. Isa, "An adaptive two-layer particle swarm optimization with elitist learning strategy," *Inf. Sci.*, vol. 273, pp. 49–72, Jul. 2014.

[48] Y. Wang, B. Li, T. Weise, J. Wang, B. Yuan, and Q. Tian, "Self-adaptive learning based particle swarm optimization," *Inf. Sci.*, vol. 181, no. 20, pp. 4515–4538, 2011.

[49] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, "Enhancing particle swarm optimization using generalized opposition-based learning," *Inf. Sci.*, vol. 181, no. 20, pp. 4699–4714, 2011.

[50] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput. World Congr. Comput. Intell.*, Anchorage, AK, USA, 1998, pp. 69–73.

[51] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput.-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[52] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evol. Comput.*, vol. 9, no. 2, pp. 159–195, 2001.

[53] Z.-Z. Liu, Y. Wang, S. Yang, and K. Tang, "An adaptive framework to tune the coordinate systems in nature-inspired optimization algorithms," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1403–1416, Apr. 2019.

[54] M. R. Bonyadi and Z. Michalewicz, "A locally convergent rotationally invariant particle swarm optimization algorithm," *Swarm Intell.*, vol. 8, no. 3, pp. 159–198, 2014.

[55] D. N. Wilke, S. Kok, and A. A. Groenwold, "Comparison of linear and classical velocity update rules in particle swarm optimization: Notes on scale and frame invariance," *Int. J. Numer. Methods Eng.*, vol. 70, no. 8, pp. 985–1008, 2007.

[56] W. Kumagai and K. Yasuda, "Particle swarm optimization with rotational invariance using correlativity," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Miyazaki, Japan, 2018, pp. 1201–1208.

[57] Y. Hariya, T. Shindo, and K. Jin'no, "A novel particle swarm optimization algorithm for non-separable and ill-conditioned problems," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Budapest, Hungary, 2016, pp. 2110–2115.

[58] M. R. Bonyadi and Z. Michalewicz, "Analysis of stability, local convergence, and transformation sensitivity of a variant of the particle swarm optimization algorithm," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 370–385, Jun. 2016.

[59] Y. Wang, H.-X. Li, T. Huang, and L. Li, "Differential evolution based on covariance matrix learning and bimodal distribution parameter setting," *Appl. Soft Comput.*, vol. 18, pp. 232–247, May 2014.

[60] Y. Wang, Z.-Z. Liu, J. Li, H.-X. Li, and G. G. Yen, "Utilizing cumulative population distribution information in differential evolution," *Appl. Soft Comput.*, vol. 48, pp. 329–346, Nov. 2016.

[61] R. G. Regis and C. A. Shoemaker, "Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization," *Eng. Optim.*, vol. 45, no. 5, pp. 529–555, 2013.

[62] J. Tian, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Multi-objective infill criterion driven Gaussian process assisted particle swarm optimization of high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 459–472, Jun. 2019.

[63] H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2664–2677, Sep, 2017.

[64] H. Yu, Y. Tan, C. Sun, and J. Zeng, "A generation-based optimal restart strategy for surrogate-assisted social learning particle swarm optimization," *Knowl. Based Syst.*, vol. 163, pp. 14–25, Jan. 2019.

[65] P. N. Suganthan *et al.*, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Nanyang Technol. Univ., Singapore, and KanGAL, IIT Kanpur, Kanpur, India, Rep. 2005005, May 2005.

[66] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Appl. Soft Comput.*, vol. 11, no. 4, pp. 3658–3670, 2011.

Fan Li (Student Member, IEEE) received the B.Eng. degree in mechanical design manufacturing and automation from the School of Electromechanical and Architectural Engineering, Jianghan University, Wuhan, China, in 2015. She is currently pursuing the Ph.D. degree in industrial engineering with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan.

Her current research interests include evolutionary optimization, multiobjective optimization, surrogate-assisted evolutionary optimization, and their applications to real-world engineering optimization problems.

Xiwen Cai (Member, IEEE) received the B.Eng. degree in mechanical design and theory from the Wuhan Institute of Technology, Wuhan, China, in 2012, and the Ph.D. degree in mechanical design and theory from the Huazhong University of Science and Technology, Wuhan, in 2017.

He is currently a Postdoctoral Fellow with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology. His current research interests include surrogates, surrogate-based optimization, surrogate-assisted evolutionary computation, and machine learning.

Liang Gao (Member, IEEE) received the B.Sc. degree in mechatronic engineering from Xidian University, Xi'an, China, in 1996, and the Ph.D. degree in mechatronic engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2002.

He is a Professor with the Department of Industrial and Manufacturing Systems Engineering, the Deputy Director of the State Key Laboratory of Digital Manufacturing Equipment and Technology, and the Vice Dean of the Research and Development Office, HUST. He was supported by the Program for New Century Excellent Talents in University in 2008 and National Science Fund for Distinguished Young Scholars of China in 2018. He published more than 200 papers indexed by SCIE, and has authored 7 monographs. His research interests include intelligent optimization algorithms, big data, and deep learning with their applications in design and manufacturing.

Prof. Gao currently serves as the Co-Editor-in-Chief for *IET Collaborative Intelligent Manufacturing* and an Associate Editor for *Swarm and Evolutionary Computation* and the *Journal of Industrial and Production Engineering*.

Weiming Shen (Fellow, IEEE) received the bachelor's and master's degrees from Northern (Beijing) Jiaotong University, Beijing, China, in 1983 and 1986, respectively, and the Ph.D. degree from the University of Technology of Compiègne, Compiègne, France, in 1996.

He is currently a Professor with the Huazhong University of Science and Technology, Wuhan, China, and an Adjunct Professor with the University of Western Ontario, London, ON, Canada. His research interest includes intelligent software agents, wireless sensor networks, IoT, big data, and their applications in industry.

Prof. Shen is a fellow of the Canadian Academy of Engineering and the Engineering Institute of Canada, and a Licensed Professional Engineer in Ontario, Canada.