Information Sciences 454–455 (2018) 59–72

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Surrogate-assisted hierarchical particle swarm optimization

Haibo Yu[a], Ying Tan[b], Jianchao Zeng[a,c], Chaoli Sun[b,*], Yaochu Jin[d,*]

[a] Department of Mechanical Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China
[b] Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China
[c] Department of Computer Science and Control Engineering, North University of China, Taiyuan 030051, China
[d] Department of Computer Science, University of Surrey, Guildford GU2 7XH, UK

ARTICLE INFO

Article history:
Received 27 November 2017
Revised 18 April 2018
Accepted 21 April 2018
Available online 24 April 2018

Keywords:
Computationally expensive problems
Surrogate model
Radial basis function
Particle swarm optimization

ABSTRACT

Meta-heuristic algorithms, which require a large number of fitness evaluations before locating the global optimum, are often prevented from being applied to computationally expensive real-world problems where one fitness evaluation may take from minutes to hours, or even days. Although many surrogate-assisted meta-heuristic optimization algorithms have been proposed, most of them were developed for solving expensive problems up to 30 dimensions. In this paper, we propose a surrogate-assisted hierarchical particle swarm optimizer for high-dimensional problems consisting of a standard particle swarm optimization (PSO) algorithm and a social learning particle swarm optimization algorithm (SL-PSO), where the PSO and SL-PSO work together to explore and exploit the search space, and simultaneously enhance the global and local performance of the surrogate model. Our experimental results on seven benchmark functions of dimensions 30, 50 and 100 demonstrate that the proposed method is competitive compared with the state-of-the-art algorithms under a limited computational budget.

1. Introduction

For solving complex real-world optimization problems, computationally efficient algorithms are in high demand. Over the past decades, a variety of metaheuristic optimization methods, such as genetic algorithms, particle swarm optimization (PSO) algorithms and differential evolution algorithms have been proposed and successfully applied to many engineering optimization problems. Their success can partly be attributed to the fact that metaheuristic algorithms do not require that the objective functions be analytical and differentiable and have better global search capability. However, a large number of fitness evaluations are usually required for a metaheuristic algorithm to locate a near-optimal solution, which poses a grand challenge for them to be applied to computationally expensive problems, which are commonly seen in the real-world, such as structural optimization design of truss topology [42], aerodynamic optimization of airfoil shape [23], streamline optimization of vehicles [19], reliability optimization of complex systems [45], crashworthiness analysis of vehicles [43], and car engine management systems [39]. To address this challenge, surrogate models, also known as meta-models, are proposed to be used in lieu of the expensive performance evaluation to reduce the computational cost. Till now, a number of surrogate assisted metaheuristic optimization algorithms have been proposed and successfully applied in practice [10,13,14]. Models, widely been used as surrogates, include polynomial regression (PR) [20], radial basis function network (RBFN) [16,36], artificial neural network (ANN) [7,15,26], Kriging or Gaussian process (GP) [1,9,31] and support vector machines (SVM) [29,41]. Comparative studies have been conducted on the performance of modeling the problems with different fitness landscape

* Corresponding author.
E-mail addresses: chaoli.sun.cn@gmail.com (C. Sun), yaochu.jin@surrey.ac.uk (Y. Jin).

https://doi.org/10.1016/j.ins.2018.04.062
0020-0255/

characteristics [6,12,49], and the empirical results showed that the RBFN performs best for problems with different degrees of nonlinearity on small size of training data and scales relatively well with the increase in search dimension [12,47], while the polynomial regression model is easy to be constructed and more convenient for analysis compared to other meta-models [12,48]. As a class of statistical learning models, GP is well suited for capturing the global landscape of complex optimization problems, and can obtain results comparable with the RBFN and PR. Different from deterministic models, GP can provide an approximated fitness value together with a confidence level of the fitness approximation, which is particularly helpful in managing surrogates. However, optimizing the hyperparameters of the GP can become very time-consuming, which is a major impediment for GP to be widely employed, especially when the dimension of the decision space of the optimization problem is high. In this work, we decided to adopt the RBFN as the surrogate model due to its satisfactory performance on both low- and high-dimensional problems.

Over the past decades, various surrogate-assisted evolutionary algorithms have been reported in the literature. In general, existing frameworks can be divided into two categories according to the number of surrogates used, i.e., single-surrogate-assisted evolutionary algorithms and multi-surrogate-assisted ones. Jin et al. [15] adopted a neural network to be a global surrogate model to assist a covariance matrix adaptation evolution strategy and investigated the effectiveness of the individual-based and generation-based model management strategies. Praveen et al. [28] developed a surrogate-assisted particle swarm optimization by employing the radial-basis-function model to construct a global surrogate for prescreening the promising solutions to save the computational resource. In [30], Regis et al. proposed a framework for particle swarm optimization with an RBF global surrogate. They first generated multiple candidate solutions for each particle in each generation, then the surrogate was employed to select the promising positions to form the new population. Chugh et al. [4] proposed a Kriging-assisted reference vector guided evolutionary algorithm for optimization of computationally expensive many-objective optimization problems. This approach constructs a local Kriging model for each objective function, where the training samples were carefully selected for reducing the computation time by taking into consideration their relationships to the reference vectors. Liu et al. [22] adopted the Gaussian process model with the lower confidence bound to prescreen solutions in a differential evolution algorithm and a dimensional reduction technique was proposed to be utilized to enhance the accuracy of the GP model. The maximum dimension of the test problems used in [22] is 50 and the dimension is reduced to 4 before the surrogate is constructed. Moreover, Wang et al. [41] suggested a support vector regression (SVR) model assisted multi-objective evolutionary algorithm for proactive scheduling in the presence of stochastic machine breakdown and deterioration effect, where a SVR was employed in the evaluation of rescheduling cost to prescreen promising individuals to be reevaluated using the time-consuming simulation. A similar framework was also reported in [29] for appointment scheduling with uncertain examination time.

Multiple surrogates have been shown to perform better than single ones in assisting evolutionary algorithms. Zhou et al. [50] suggested to combine a global surrogate with a local surrogate in a hierarchical way to accelerate the evolutionary search, in which the main idea is using a global GP model to prescreen the offspring and an RBF based trust-region algorithm for local search. Sun et al. [36] introduced the global RBFN into the fitness inheritance based evolutionary framework [37] and proposed a two-layer surrogate based PSO for computationally expensive problems. Tenne et al. [40] developed an improved version of hierarchical surrogate-assisted memetic algorithm by using variable global and local RBF models. During the optimization, the optimal global and local RBF models were adaptively determined based on the leave-one-out cross validation. Tang et al. [38] presented a hybrid surrogate-assisted PSO, in which an RBF model constructed by interpolating the residual errors was added to a low order polynomial regression model to form the final hybrid surrogate model. Lim et al. [21] proposed a generalized framework for surrogate-assisted single- and multi-objective evolutionary optimization employing an ensemble-based local surrogate and a low-order polynomial based global surrogate in the local search. Inspired by ideas in active learning, Wang et al. [44] proposed an ensemble surrogate based model management method for surrogate-assisted PSO which searches for the promising and most uncertain candidate solutions to be evaluated using the expensive fitness function.

In most existing surrogate-assisted optimization algorithms using multiple surrogate models, the global surrogate model typically targets to smoothen out the local optima, while the local ones aim to capture the local details of the fitness function around the neighborhood of the current individuals. However, performing local search around each individual may be inefficient in helping find the global optimum and it might be more helpful to find the optimum of the search space that the population currently covers. Therefore, in this work, we propose a surrogate-assisted hierarchical PSO for high-dimensional bound-constrained optimization problems by combining SL-PSO with PSO, in which the SL-PSO algorithm aims to find the optimum of the current search region so that the surrogate is able to accurately learn the local fitness landscape around the optimum, whereas the PSO algorithm helps explore the search space gradually so that the surrogate can approximate the global profile of the fitness landscape. This way, the RBF surrogate is able to learn both the local details and the relatively global features of the fitness landscape as the search proceeds, enabling the search algorithms to find the optimum more efficiently. As the search carried out by SL-PSO is performed within each generation of the PSO, we term the proposed algorithm a surrogate-assisted hierarchical particle swarm optimization (SHPSO) algorithm.

The rest of the paper is organized as follows: Section 2 briefly reviews the two particle swarm optimization algorithms and the radial basis function network. The surrogate-assisted hierarchical particle swarm optimization is detailed in Section 3. Section 4 presents the experimental results with discussions. Section 5 concludes the paper with a summary and suggestions of future research.

2. Related work

2.1. Particle swarm optimization algorithms

The canonical particle swarm optimization algorithm that simulates swarm behaviors of social animals such as the bird flocking or fish schooling was proposed by Kennedy and Eberhart in 1995 [17]. PSO has attracted wide attention in engineering design optimization owing to its algorithmic simplicity and powerful search performance. Different to genetic algorithms and differential evolution, which generate offspring through crossover and mutation operations, PSO maintains a number of good solutions as the leaders to guide the swarm towards the optimum. Various PSO algorithms have been proposed in order to improve the performance of the canonical PSO algorithm, for instance, a PSO algorithm with a constriction factor [5] is a minor variant of the canonical PSO, which has been showed the good convergence capability [8].

The SL-PSO was proposed very recently, which removed the need for maintaining a global best and personal best solutions by learning from randomly chosen better particles in the swarm [3]. SL-PSO was shown to perform very well on both low- and high-dimensional problems, which can mainly be attributed to its capability of escaping from local optima [25,46].

In both PSO variants, an initial swarm (population) is randomly generated and each particle in the swarm is assigned with a certain velocity. The velocity of each particle in the PSO with the constriction factor is updated according to Eq. (1):

$$v_{ij}^{(t+1)} = \chi\left(v_{ij}^{(t)} + c_1 r_1\left(pbest_{ij}^{(t)} - x_{ij}^{(t)}\right) + c_2 r_2\left(gbest_j^{(t)} - x_{ij}^{(t)}\right)\right), \tag{1}$$

where $1 < i \le N$, N is the swarm size, $\boldsymbol{x}_i^{(t)} = (x_{i1}^{(t)}, x_{i2}^{(t)}, \ldots, x_{id}^{(t)})$ and $\boldsymbol{v}_i^{(t)} = (v_{i1}^{(t)}, v_{i2}^{(t)}, \ldots, v_{id}^{(t)})$ are the position and velocity of the i^{th} particle at generation t, respectively. In Eq. (1), $\boldsymbol{pbest}_i^{(t)} = (pbest_{i1}^{(t)}, pbest_{i2}^{(t)}, \ldots, pbest_{id}^{(t)})$ is the i^{th} particle's personal best position found up to generation t, and $\boldsymbol{gbest}^{(t)} = (gbest_1^{(t)}, gbest_2^{(t)}, \ldots, gbest_d^{(t)})$ is the current global best position found by the whole swarm. Both c_1 and c_2 are acceleration coefficients, and χ is the constriction factor with the form $\chi = 2/(2 - \phi - \sqrt{(\phi^2 - 4\phi)})$, where $\phi = c_1 + c_2 > 4$. Generally, c_1 and c_2 are all set to 2.05 [5]. Hereinafter, for simplicity, we also use PSO to denote the particle swarm optimization algorithm with the constriction factor.

The velocity of each particle in the SL-PSO algorithm is updated as follows:

$$v_{ij}^{(t+1)} = r_1 v_{ij}^{(t)} + r_2\left(x_{kj}^{(t)} - x_{ij}^{(t)}\right) + r_3\varepsilon\left(\bar{x}_j^{(t)} - x_{ij}^{(t)}\right), \tag{2}$$

where r_1, r_2 and r_3 are three random numbers uniformly distributed within [0, 1]. x_{kj} is the j^{th} element in the particle k (known as the demonstrator of particle i) whose fitness is better than that of particle i, $\bar{x}_j^{(t)} = (\sum_{i=1}^{N} x_{ij}^{(t)})/N$ represents the mean position on j^{th} dimension of the current swarm, ε is denoted as the social influence factor that controls the influence of mean position $\bar{x}_j^{(t)}$.

After each individual updates its velocity, the new position of the particles of both PSO and SL-PSO will be updated using Eq. (3).

$$\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} + \boldsymbol{v}_i^{(t+1)}. \tag{3}$$

2.2. Radial basis function network

Radial-basis-function networks are a type of artificial neural networks [33], which have been shown to work well for approximating high-dimensional nonlinear functions. In addition, RBFNs have shown to be capable of both local and global modeling [11]. The RBFN utilized in this paper is of an interpolation form defined as follows.

$$\hat{y}(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \varphi(\|\boldsymbol{x} - \boldsymbol{x}_i\|), \tag{4}$$

where $\|\cdot\|$ and $\varphi(\cdot)$ are the Euclidian norm and kernel function, respectively. Representative radial basis kernels include Gaussian function, thin-plate splines, linear splines, cubic splines, and multi-quadrics splines. In this paper, Gaussian kernel function defined as $\varphi(x) = \exp(-x^2/\beta)$ is utilized to construct the RBF model. In addition, $\{\alpha_i\}_{i=1}^{N}$ denotes the weight coefficients, which can be obtained by solving the following linear system.

$$\boldsymbol{\alpha} = \boldsymbol{\Phi}^{-1}\boldsymbol{F}, \tag{5}$$

where $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}^T$ is the parameter vector, $\boldsymbol{\Phi} = [\varphi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)]_{N\times N}$ is the kernel matrix. Note that the interpolation matrix $\boldsymbol{\Phi}$ is positive definite as long as the points $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ are all different [27]. In terms of the shape parameter in Gaussian kernel function, we empirically set $\beta = D_{\max}(dN)^{-1/d}$, where $D_{\max}$ is the maximal distance between the training data. Once the training dataset and parameter β are available, a radial basis function network can be easily trained.

3. Surrogate-assisted hierarchical particle swarm optimization

One central issue in surrogate-assisted evolutionary optimization is to enable the surrogate to learn both the global profile of the fitness landscape so that the optimizer can find the region in which the global optimum is located as soon as possible, while it should also be able to properly describe the details of the fitness functions near the optimum so that the optimizer can approach the optimum accurately. An idea that has been exploited in the literature is to employ two surrogates, one global and one local. One challenge of using two surrogates lies in the achievement of the right training samples to train the global surrogate and the local surrogate. In this paper, we use an RBFN to learn the local details of the fitness landscape while approximating the global profile of the fitness function in the region the current swarm is searching. To achieve this target, we embed an SL-PSO based search process in each search iteration of the PSO, hoping to find the optimum learned by the current surrogate.

For minimization problems, the main steps of the proposed method are listed below:

Step1: Generate initial solutions (more than the population size) using Latin Hypercube Sampling (LHS) and evaluate them using the exact fitness function. Store them in the archive.
Step2: Rank the solutions in the archive in an ascending order according to their fitness values and select the first N best ones to form the initial population of PSO. Calculate the personal best of each particle as well as the best position of the swarm.
Step3: Build an RBF model using the first P non-duplicated best samples in the archive.
Step4: Find the optimum of the RBF model by SL-PSO, evaluate its fitness value using the exact fitness function, and save it into the archive.
Step5: Update the RBF model if the first P non-duplicated best samples in the archive have been changed.
Step6: Update the global best by Eq. (7), and generate a new swarm for PSO by Eqs. (3) and (6).
Step7: Estimate the fitness of each particle in the new swarm by using the RBF model, and screen out the particles whose estimated values are better than that of their personal bests.
Step8: Evaluate the fitness of the screened particles in Step 7 using the real fitness function and store them into the archive.
Step9: Update the personal best position of each particle and the global best position of the swarm.
Step10: Exit if the predefined stopping criterion is satisfied, otherwise go to Step 3.

In the proposed method, Latin Hypercube Sampling (LHS) [18] is used to generate a set of initial solutions that are evaluated using the computationally expensive real fitness function. Then, all data in the archive will be sorted in ascending order according to the fitness value and the first N data will be chosen to form the initial population of PSO, where N denotes the size of the swarm.

The following remarks shall be made. First, the SL-PSO used in this work is exactly the same as in [3], except that the fitness evaluations are all based on the RBF surrogate. The optimum found by SL-PSO, denoted by ***sbest***, will always be evaluated using the real computationally expensive fitness function so that only one expensive fitness evaluation is required for the SL-PSO at each generation of PSO. Second, if ***sbest*** is better than the current global best, denoted by ***gbest***, then the velocity of the particles of the PSO will be updated based on ***sbest*** instead of the ***gbest***, as described in Eqs. (6) and (7). Third, all particles whose fitness value is better than the current personal best according to the surrogate will be evaluated using the real computationally expensive fitness function and will be added into the archive.

$$\begin{cases} \boldsymbol{v}_i^{(t+1)} = \chi(\boldsymbol{v}_i^{(t)} + c_1 r_1(\boldsymbol{pbest}_i^{(t)} - \boldsymbol{x}_i^{(t)}) + c_2 r_2(\boldsymbol{sbest}^{(t)} - \boldsymbol{x}_i^{(t)})), & if\ f(\boldsymbol{sbest}^{(t)}) < f(\boldsymbol{gbest}^{(t)}), \\ \boldsymbol{v}_i^{(t+1)} = \chi(\boldsymbol{v}_i^{(t)} + c_1 r_1(\boldsymbol{pbest}_i^{(t)} - \boldsymbol{x}_i^{(t)}) + c_2 r_2(\boldsymbol{gbest}^{(t)} - \boldsymbol{x}_i^{(t)})), & otherwise. \end{cases} \tag{6}$$

$$\begin{cases} \boldsymbol{gbest} = \boldsymbol{sbest}, \quad \boldsymbol{pbest}_g = \boldsymbol{sbest}, & if\ f(\boldsymbol{sbest}) < f(\boldsymbol{gbest}), \\ \boldsymbol{gbest} = \boldsymbol{gbest}, \quad \boldsymbol{pbest}_g = \boldsymbol{pbest}_g, & otherwise. \end{cases} \tag{7}$$

Note that it is hard to build a high-fidelity global model when only a limited number of sample data are available, especially when the search dimension is high. Meanwhile, the computational cost will increase if all the samples in the archive are used to train the RBF model. So in our method, only a predefined number of best solutions are utilized to train the RBF model in each generation except in the first generation, in which all data in the archive are used to train the surrogate. It is conceivable that the candidate solutions generated during optimization process in the PSO algorithm will gradually aggregate in the region in which the local or global optimum is located. As a result, a relatively high-quality surrogate model for the region in which the optimum is located will be constructed.

4. Empirical study

Seven benchmark functions with different characteristics in the fitness landscape [21,22,34,45] are taken to evaluate the effectiveness of SHPSO. Table 1 lists the characteristics of these test problems. We also consider two variants of SHPSO, an RBF-assisted PSO (RBF-PSO) and an RBF-assisted global searcher (RBF-GL), for comparison. Note that the RBF-PSO algorithm is the same as SHPSO except that no SL-PSO is utilized to search for the optimum of the surrogate. By contrast, in the RBF-GL algorithm, only the SL-PSO algorithm is employed to search for the optimum of an RBF model, which is updated every time the optimum found by SL-PSO is evaluated using the expensive fitness function. Algorithm 1 gives the pseudo code of the RBF-GL algorithm. Comparative studies of SHPSO, RBF-PSO, RBF-GL, a standard PSO algorithm [5] and three state-of-the-art surrogate-assisted algorithms, i.e., GPEME [22], GS-SOMA [21], and SA-COSO [35] are conducted on the seven problems with

Table 1
Benchmark functions used in the experiment.

Benchmark problem	Description	Characteristics	Global optimum ($f(x^*)$)
F1	Ellipsoid	Unimodal	0
F2	Rosenbrock	Multimodal with narrow valley	0
F3	Ackley	Multimodal	0
F4	Griewank	Multimodal	0
F5	Shifted Rotated Rastrigin (F10 in [34])	Very complicated multimodal	−330
F6	Rotated Hybrid composition function (F16 in [34])	Very complicated multimodal	120
F7	Rotated Hybrid composition function (F19 in [34])	Very complicated multimodal	10

Algorithm 1 RBF-assisted global searcher (RBF-GL).

Initialization: Generate candidate solutions using LHS in the decision space, evaluate their fitness value using the expensive fitness function and store them in the archive. Determine the best position ***cbest*** found so far.
Construct an RBF global surrogate model using the P top-ranking best data in the archive.
While (computational budget is not exhausted) **do**
  Find the optimum ***sbest*** of the RBF model using the SL-PSO algorithm, evaluate its fitness value using the expensive fitness function.
  Save the optimum into the archive.
  If $f(\boldsymbol{sbest}) < f(\boldsymbol{cbest})$ **then**
    $\boldsymbol{cbest} = \boldsymbol{sbest}$
  End If
  If (The P top-ranking best data of the archive are changed) **then**
    Update the RBF model using the new P top-ranking best data.
  End If.
End While
Output

30, 50 and 100 dimensions. All compared algorithms are implemented in MATLAB®R2016b and run on an Intel(R) Core(TM) i7-6500U CPU @ 2.50 GHz laptop. Note that the convergence profiles of GPEME, GS-SOMA, and SA-COSO on the selected benchmark functions are plotted using data extracted from the corresponding references. Since the results of GPEME and SA-COSO on function F6 and the results of GS-SOMA on function F1 were not reported.

4.1. Parameter settings

In the experiments, the population sizes of SHPSO, RBF-PSO, RBF-GL and PSO are all set to 50 for 30-, 50-, and 100-dimensional problems. The constriction factor χ is fixed to 0.729 [8,24], and the acceleration coefficients c_1 and c_2 are all set to 2.05. The number of samples collected by the Latin hypercube sampling (LHS) before the optimization starts is set to 100 for 30- and 50-dimensional problems, and 200 for 100-dimensional problems, respectively. The sizes of data to train the RBF surrogate model are set to 100 for 30- and 50-dimensional problems, and 150 for 100-dimensional problems. The SL-PSO algorithm used in SHPSO adopts the same parameter settings suggested in [3], and the stopping criterion of SL-PSO is that the global optimum of RBF surrogate remains unchanged for 20 generations. The stopping criteria of each algorithm under comparison is that the maximum number of expensive fitness evaluations reaches 1000 and 8000, respectively.

4.2. Comparison results on 30-dimensional benchmark problems

Tables 2 and 3 present the comparison results including the *t*-test results at a confidence level of 95% comparing SHPSO to other algorithms, where '+', '−', and '≈' indicates that SHPSO is significantly better than, significantly worse than, or comparable to the compared algorithm, respectively. The best mean result of each test instance is highlighted and symbol '*' indicates that the *t*-test results on this algorithm cannot be able to given, as we are not able to replicate the results of GPEME [22] and GS-SOMA [21]. Figs. 1 and 2 plot the convergence profiles of the compared algorithms on 30-dimensional problems.

The statistical results of Table 2 show that SHPSO performs significantly better or comparably well. For instance, compared with GPEME, SHPSO obtains better average results on all problems except for the Ellipsoid function (F1). From Fig. 1, we can also see that SHPSO achieves significantly better convergence performance on most of the test problems. For the Ellipsoid function (F1), although the optimal solution obtained after 1000 expensive fitness evaluations is not better than GPEME, we can find that SHPSO can converge faster than GPEME at the early search stage (before 800 expensive fitness evaluations). This performance degradation at the later search stage may be attributed to the nature of the asymmetrical fitness landscape of the Ellipsoid function, which results in low accuracy of the RBFN using the symmetrical Gaussian basis kernel function. We can also find from Fig. 1 that both SHPSO and RBF-PSO can get better solutions than GPEME after 1000 expensive fitness evaluations on F5, and always maintain a good convergence after 8000 expensive fitness evaluations as showed in Fig. 2. The function F5 with a single funnel structure is the only one among all benchmark problems on which RBF-PSO obtains a better convergence property than GPEME. We speculate that this is due to the better performance of RBF

Table 2
Statistical results of proposed algorithm SHPSO and comparison algorithms on 30-D benchmark problems.

Problem	Algorithm	Best	Worst	Mean	Std.	t-test
F1	PSO	2.7834E + 02	6.9944E + 02	4.4236E + 02	1.0568E + 02	+
	RBF-PSO	2.9653E + 01	6.2506E + 01	4.6265E + 01	9.6713E + 00	+
	RBF-GL	1.1213E-02	1.4626E + 01	2.0553E + 00	3.4294E + 00	+
	GPEME	1.5500E-02	1.6470E-01	**7.6200E-02**	4.0100E-02	*
	SHPSO	4.4782E-02	7.2024E-01	2.1199E-01	1.5229E-01	
F2	PSO	5.3741E + 02	1.2405E + 03	8.5907E + 02	1.8764E + 02	+
	RBF-PSO	6.9388E + 01	2.5172E + 02	1.4491E + 02	4.6689E + 01	+
	RBF-GL	7.8685E + 01	2.3511E + 02	1.6694E + 02	3.8776E + 01	+
	GPEME	2.6262E + 01	8.8233E + 01	4.6177E + 01	2.5520E + 01	*
	SHPSO	2.7726E + 01	2.9290E + 01	**2.8566E + 01**	4.0441E-01	
F3	PSO	1.4404E + 01	1.8252E + 01	1.6491E + 01	1.0076E + 00	+
	RBF-PSO	7.0875E + 00	1.1070E + 01	8.4328E + 00	9.6422E-01	+
	RBF-GL	9.7007E + 00	1.2942E + 01	1.0963E + 01	1.0511E + 00	+
	GPEME	1.9491E + 00	4.9640E + 00	3.0105E + 00	9.2500E-01	*
	SHPSO	5.6091E-01	2.9574E + 00	**1.4418E + 00**	7.7404E-01	
F4	PSO	6.3000E + 01	1.7116E + 02	1.0780E + 02	2.8209E + 01	+
	RBF-PSO	5.8667E + 00	2.3930E + 01	1.2205E + 01	4.5623E + 00	+
	RBF-GL	6.5297E-01	1.0457E + 00	9.5813E-01	9.9526E-02	≈
	GPEME	7.3680E-01	1.0761E + 00	9.9690E-01	1.0800E-01	*
	SHPSO	7.0609E-01	1.0275E + 00	**9.2053E-01**	8.8062E-02	
F5	PSO	−5.1321E + 01	1.1370E + 02	1.2941E + 01	3.8647E + 01	+
	RBF-PSO	−9.6661E + 01	4.0560E + 01	−3.5083E + 01	3.4839E + 01	+
	RBF-GL	−3.0526E + 01	5.7939E + 01	1.3743E + 01	2.4372E + 01	+
	GPEME	−5.7068E + 01	1.8033E + 01	−2.1861E + 01	3.6449E + 01	*
	SHPSO	−1.3297E + 02	−5.9993E + 01	**−9.2830E + 01**	2.2544E + 01	
F6	PSO	4.5176E + 02	1.0945E + 03	6.2545E + 02	1.4411E + 02	+
	RBF-PSO	4.1305E + 02	6.9852E + 02	5.6210E + 02	1.1136E + 02	+
	RBF-GL	5.2422E + 02	8.6997E + 02	6.6125E + 02	1.1069E + 02	+
	GPEME	–	–	–	–	*
	SHPSO	3.2715E + 02	6.4948E + 02	**4.6433E + 02**	8.5125E + 01	
F7	PSO	9.7824E + 02	1.1696E + 03	1.0373E + 03	4.6517E + 01	+
	RBF-PSO	9.6013E + 02	1.0079E + 03	9.8436E + 02	1.4174E + 01	+
	RBF-GL	1.0005E + 03	1.2843E + 03	1.1433E + 03	8.7666E + 01	+
	GPEME	9.3316E + 02	9.9286E + 02	9.5859E + 02	2.5695E + 01	*
	SHPSO	9.2248E + 02	9.6363E + 02	**9.3961E + 02**	9.0177E + 00	

Table 3
Statistical results of algorithms PSO, RBF-PSO, GS-SOMA, and SHPSO on 30-D benchmark problems after 8000 expensive fitness evaluations.

Problem	Algorithm	Best	Worst	Mean	Median	Std.	t-test
F1	PSO	1.4140E-01	2.6436E + 00	2.9960E-01	5.6264E-01	6.6341E-01	+
	RBF-PSO	6.3371E-12	2.7357E-09	2.4943E-10	7.6499E-11	6.0738E-10	+
	GS-SOMA	–	–	–	–	–	*
	SHPSO	3.1233E-22	2.1892E-18	**2.7135E-19**	1.4894E-20	6.2613E-19	
F2	PSO	2.2811E + 01	1.4057E + 02	6.1615E + 01	6.0380E + 01	3.5420E + 01	+
	RBF-PSO	2.2513E + 01	8.4003E + 01	3.3180E + 01	2.5500E + 01	1.8949E + 01	+
	GS-SOMA	2.8300E + 01	1.2600E + 02	4.6300E + 01	3.0200E + 01	2.9200E + 01	*
	SHPSO	1.9563E + 01	2.5648E + 01	**2.0721E + 01**	2.0575E + 01	1.2201E + 00	
F3	PSO	1.3164E + 00	3.3136E + 00	2.5104E + 00	2.5878E + 00	4.4461E-01	+
	RBF-PSO	4.3589E-05	5.0919E + 00	1.6034E + 00	1.6462E + 00	1.1977E + 00	+
	GS-SOMA	2.8700E + 00	4.2800E + 00	3.5800E + 00	3.6700E + 00	5.0900E-01	*
	SHPSO	1.8141E-11	4.3856E-09	**5.9361E-10**	1.3298E-10	1.0316E-09	
F4	PSO	1.0207E + 00	1.2786E + 00	1.0894E + 00	1.0702E + 00	6.2170E-02	+
	RBF-PSO	1.3917E-08	7.5394E-02	1.7118E-02	7.3961E-03	2.4899E-02	+
	GS-SOMA	1.4000E-10	1.5400E-02	**2.2000E-03**	8.9500E-09	4.6000E-03	*
	SHPSO	0.0000E + 00	3.6799E-02	4.9188E-03	7.6938E-14	9.3947E-03	
F5	PSO	−2.5454E + 02	−5.8044E + 01	−1.3025E + 02	−1.1167E + 02	5.4160E + 01	+
	RBF-PSO	−2.9115E + 02	−1.2794E + 02	−2.1868E + 02	−2.2146E + 02	4.1148E + 01	+
	GS-SOMA	−1.6400E + 02	−9.9700E + 01	−1.2600E + 02	−1.2300E + 02	1.6000E + 01	*
	SHPSO	−3.0513E + 02	−2.2354E + 02	**−2.7015E + 02**	−2.7030E + 02	2.0419E + 01	
F6	PSO	2.6898E + 02	6.5018E + 02	4.2789E + 02	3.7055E + 02	1.2889E + 02	+
	RBF-PSO	2.1269E + 02	6.2168E + 02	3.5908E + 02	2.8791E + 02	1.4963E + 02	≈
	GS-SOMA	2.2300E + 02	5.5400E + 02	**3.2500E + 02**	2.8600E + 02	1.1700E + 02	*
	SHPSO	1.9080E + 02	6.2000E + 02	3.3562E + 02	2.7807E + 02	1.6387E + 02	
F7	PSO	9.4923E + 02	1.0041E + 03	9.6745E + 02	9.6180E + 02	1.6404E + 01	+
	RBF-PSO	9.4275E + 02	1.0199E + 03	9.6571E + 02	9.6000E + 02	1.9824E + 01	+
	GS-SOMA	9.3000E + 02	9.8600E + 02	9.4200E + 02	9.3700E + 02	1.7500E + 01	*
	SHPSO	9.2685E + 02	9.5002E + 02	**9.3689E + 02**	9.3660E + 02	6.3489E + 00	

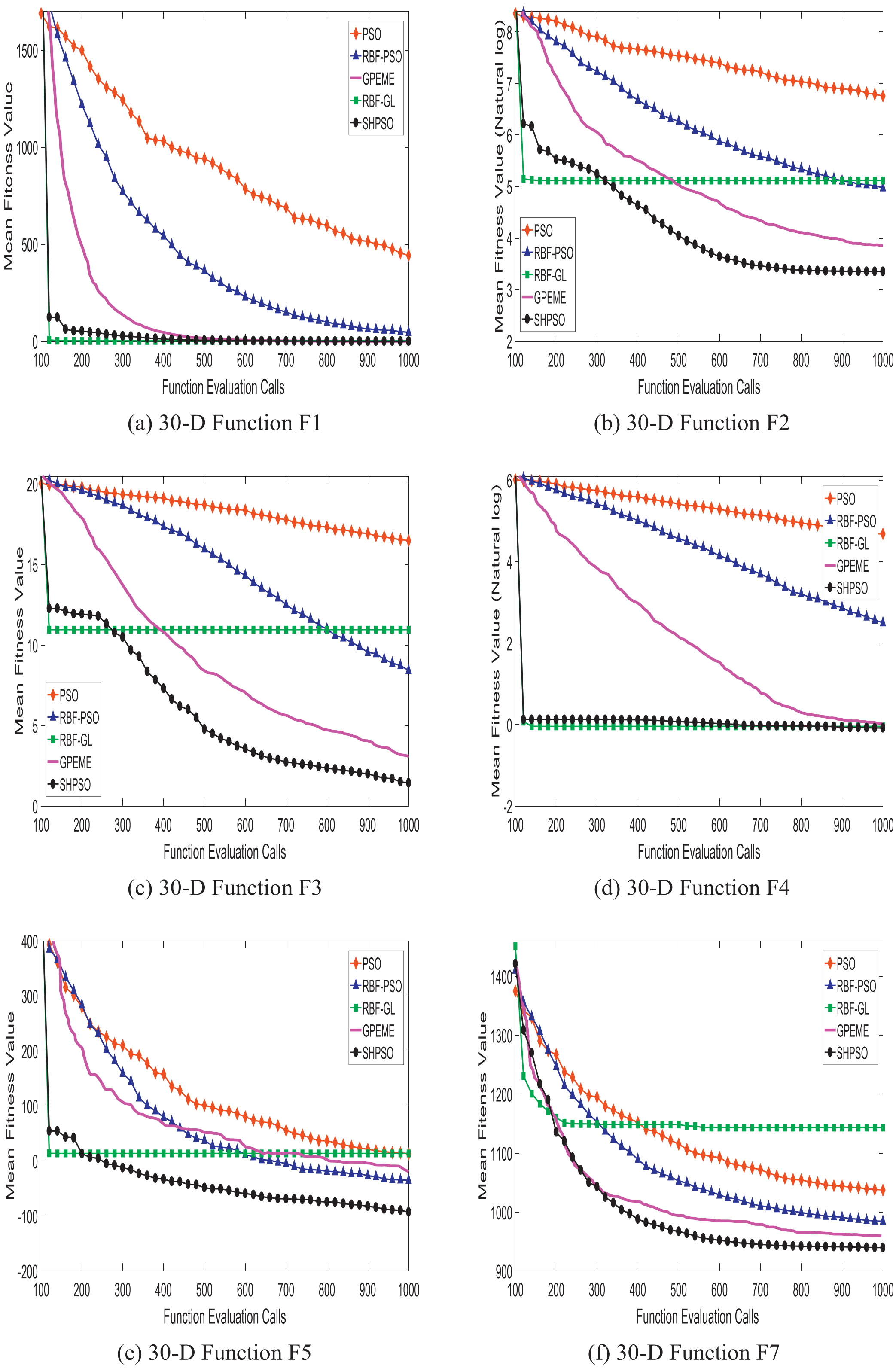


Fig. 1. Convergence profiles of algorithms PSO, RBF-PSO, RBF-GL, GPEME, and SHPSO on 30-D benchmark problems with 1000 expensive fitness evaluations.

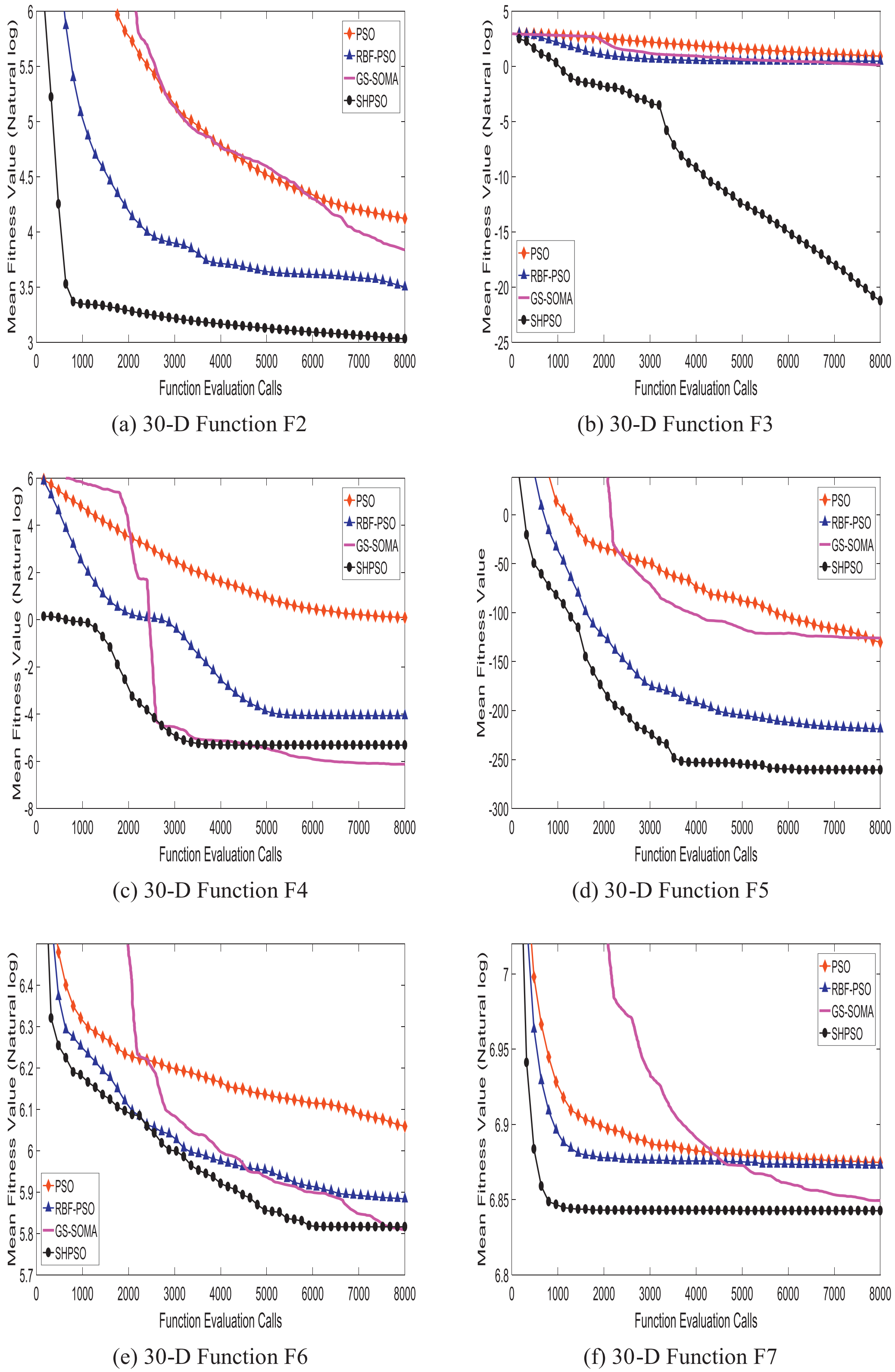
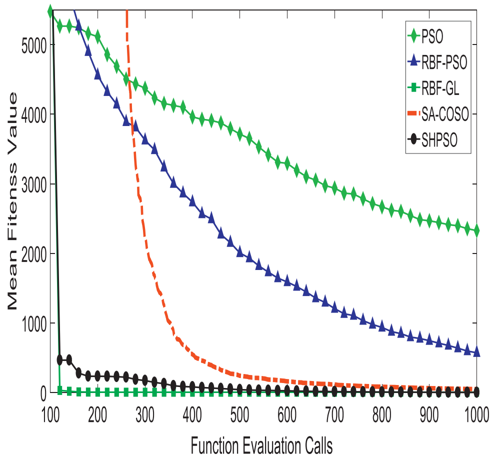
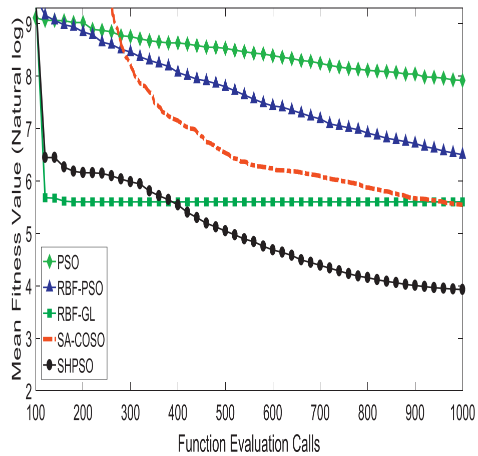


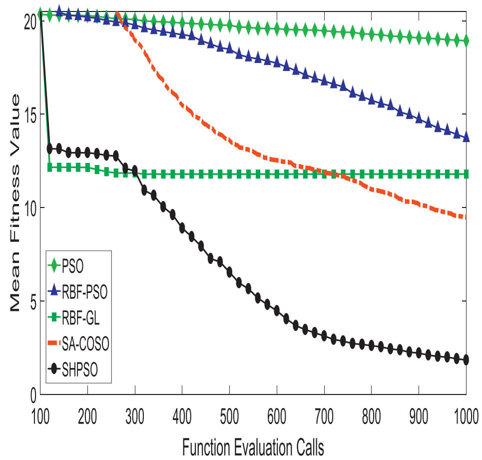
Fig. 2. Convergence profiles of algorithms PSO, RBF-PSO, GS-SOMA and SHPSO on 30-D benchmark problems after 8000 expensive fitness evaluations.



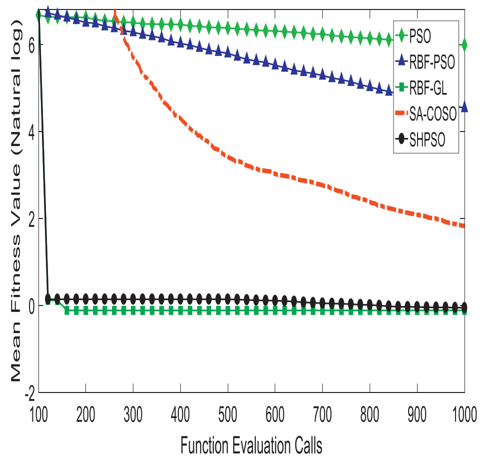
(a) 50-D Function F1



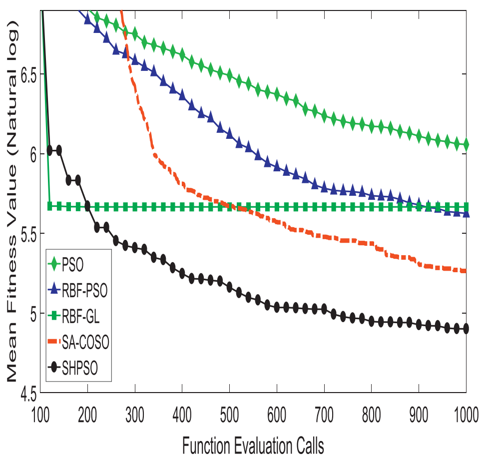
(b) 50-D Function F2



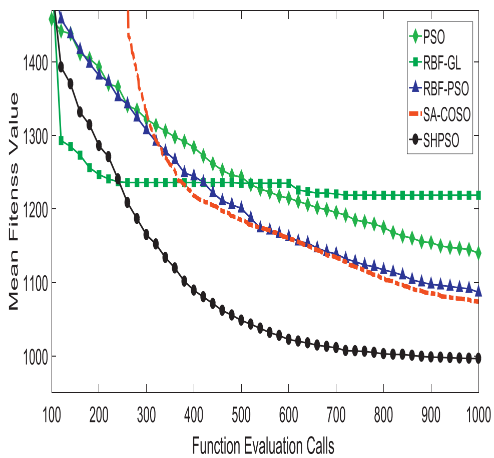
(c) 50-D Function F3



(d) 50-D Function F4

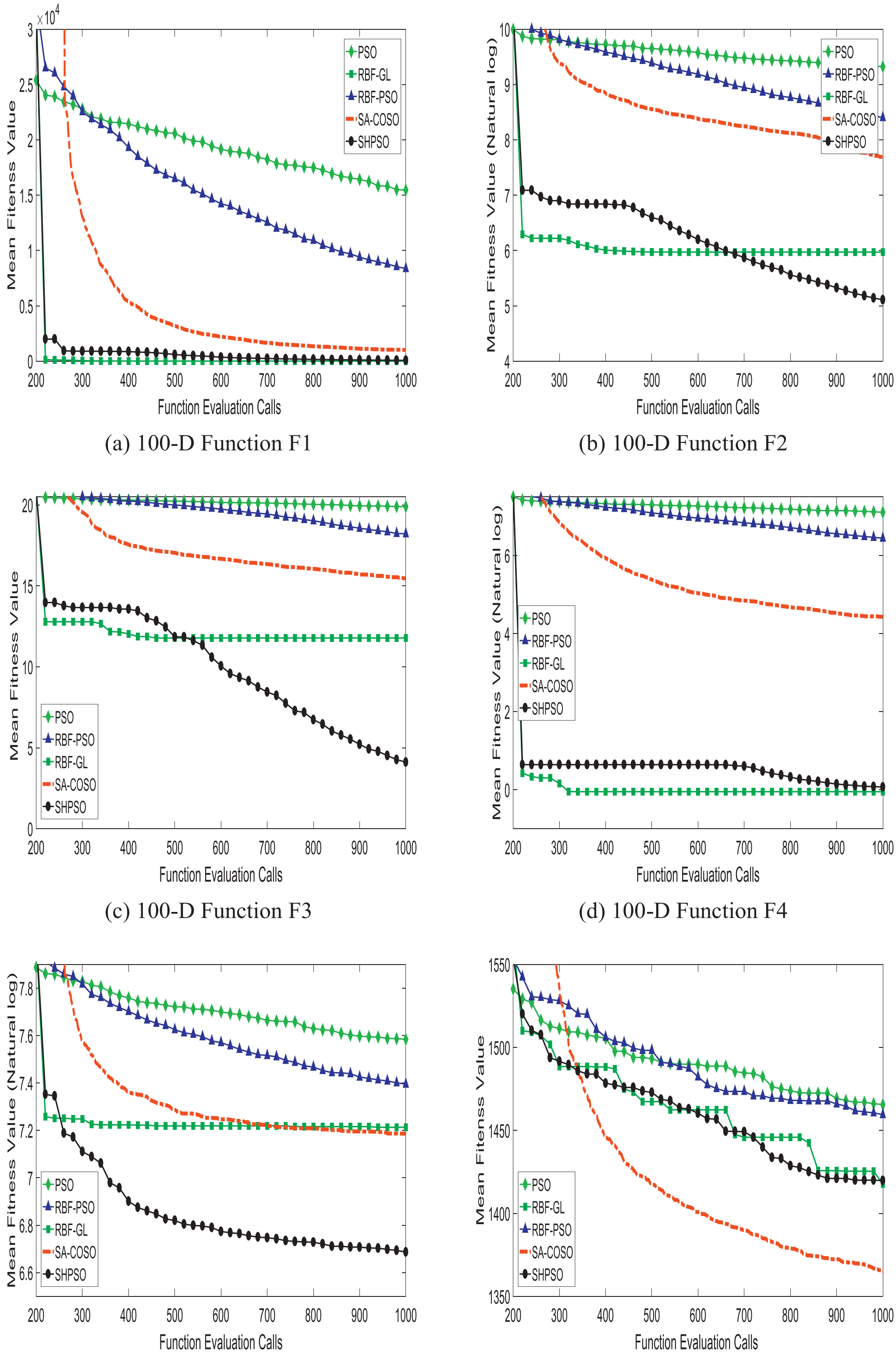


(e) 50-D Function F5



(f) 50-D Function F7

Fig. 3. Convergence profiles of algorithms PSO, RBF-PSO, RBF-GL, SA-COSO, and SHPSO on 50-D benchmark problems after 1000 expensive fitness evaluations.



(a) 100-D Function F1 (b) 100-D Function F2

(c) 100-D Function F3 (d) 100-D Function F4

(e) 100-D Function F5 (f) 100-D Function F7

Fig. 4. Convergence profiles of algorithms PSO, RBF-PSO, RBF-GL, SA-COSO, and SHPSO on 100-D benchmark problems after 1000 expensive fitness evaluations.

Table 4
Statistical results of comparison algorithms PSO, RBF-PSO, RBF-GL, GPEME, SA-COSO, and SHPSO on 50-D benchmark problems.

Problem	Algorithm	Mean	Std.	t-test
F1	PSO	2.3286E + 03	4.3063E + 02	+
	RBF-HSPSO	5.7380E + 02	1.2103E + 02	+
	RBF-GL	7.0405E + 00	1.3247E + 01	+
	GPEME	2.2108E + 02	8.1612E + 01	*
	SA-COSO	5.1475E + 01	1.6246E + 01	+
	SHPSO	**4.0281E + 00**	2.0599E + 00	
F2	PSO	2.7384E + 03	7.1075E + 02	+
	RBF-HSPSO	6.6448E + 02	1.5042E + 02	+
	RBF-GL	2.7044E + 02	4.6009E + 01	+
	GPEME	2.5828E + 02	8.0188E + 01	*
	SA-COSO	2.5258E + 02	4.0744E + 01	+
	SHPSO	**5.0800E + 01**	3.0305E + 00	
F3	PSO	1.8935E + 01	3.5350E-01	+
	RBF-HSPSO	1.3732E + 01	9.6023E-01	+
	RBF-GL	1.1797E + 01	8.4180E-01	+
	GPEME	1.3233E + 01	1.5846E + 00	*
	SA-COSO	8.9318E + 00	1.0668E + 00	+
	SHPSO	**1.8389E + 00**	5.6370E-01	
F4	PSO	3.9779E + 02	4.5050E + 01	+
	RBF-HSPSO	9.3277E + 01	2.2309E + 01	+
	RBF-GL	8.9125E-01	1.1696E-01	≈
	GPEME	3.6646E + 01	1.3176E + 01	*
	SA-COSO	6.0062E + 00	1.1043E + 00	+
	SHPSO	**9.4521E-01**	6.1404E-02	
F5	PSO	4.2748E + 02	6.9498E + 01	+
	RBF-HSPSO	2.7748E + 02	5.5275E + 01	+
	RBF-GL	2.8903E + 02	4.2247E + 01	+
	SA-COSO	1.9916E + 02	3.0599E + 01	+
	SHPSO	**1.3442E + 02**	3.2256E + 01	
F6	PSO	6.6298E + 02	5.5912E + 01	+
	RBF-HSPSO	5.4243E + 02	6.7409E + 01	+
	RBF-GL	9.2381E + 02	1.6245E + 02	+
	SA-COSO	–	–	*
	SHPSO	**4.7438E + 02**	4.2029E + 01	
F7	PSO	1.1399E + 03	4.3747E + 01	+
	RBF-HSPSO	1.0865E + 03	3.3714E + 01	+
	RBF-GL	1.2186E + 03	6.3168E + 01	+
	SA-COSO	1.0809E + 03	3.2859E + 01	+
	SHPSO	**9.9660E + 02**	2.2145E + 01	

in capturing the global structure of highly nonlinear high-dimensional problems, enabling the SLPSO algorithm to locate more quickly the optimum of the problem. On the Griewank function (F4) with a single funnel structure and regularly distributed local optima, SHPSO can quickly find a local optimal solution, but then appears to stagnate in the later generation. On the complex composite test function F7, SHPSO also achieves significantly better convergence property than GPEME.

From Fig. 1, we also can find that RBF-GL exhibits a fast convergence on all benchmark problems considered in this work within the first 200 expensive fitness evaluations. It is because RBF-GL performs only one fitness evaluation using the expensive fitness function at each iteration, while SHPSO consumes more computational budget in each generation. However, RBF-GL converges prematurely on all benchmark functions. An interesting phenomenon can be observed in Fig. 1(d) is that the convergence profiles of RBF-GL and SHPSO nearly overlap on function F4, indicating that SL-PSO plays a major role in solving function F4.

In order to examine the performance of SHPSO when more expensive fitness evaluations are allowed, Table 3 presents the comparative results of all the compared algorithms listed in Table 2 except for GPEME when 8000 expensive fitness evaluations are allowed. Fig. 2 plots the convergence profiles of the compared algorithms. RBF-GL experiences premature converge on all problems with 1000 expensive fitness evaluations, as shown in Fig. 1. Therefore, the convergence profile of RBF-GL is not presented in Fig. 2. From Fig. 2, we can find that compared to GS-SOMA, SHPSO achieves better convergence performance on F2, F3 and F5, and comparable results on F6 and F7. In addition, we can observe in Fig. 2(c) that the convergence profile of SHPSO on F4 improves quickly between 1000 and 4000 expensive fitness evaluations. From both Figs. 1 and 2, we can conclude that SHPSO converges much faster than GPEME and GS-SOMA.

4.3. Comparison results on 50 and 100-dimensional benchmark problems

Most recently, surrogate-assisted evolutionary optimization of high-dimensional problems has also been reported [2,32,35]. In order to further evaluate the efficiency of SHPSO on high-dimensional problems, here we perform experiments

Table 5
Statistical results of comparison algorithms PSO, RBF-PSO, RBF-GL, SA-COSO, and SHPSO on 100-D benchmark problems.

Problem	Algorithm	Mean	Std.	t-test
F1	PSO	1.5466E + 04	1.2748E + 03	+
	RBF-PSO	8.3685E + 03	9.1474E + 02	+
	RBF-GL	**7.6202E + 00**	2.0039E + 01	−
	SA-COSO	1.0332E + 03	3.1718E + 02	+
	SHPSO	7.6106E + 01	2.1447E + 01	
F2	PSO	1.1206E + 04	1.4045E + 03	+
	RBF-PSO	4.4615E + 03	6.3903E + 02	+
	RBF-GL	3.9210E + 02	1.7049E + 02	+
	SA-COSO	2.7142E + 03	1.1702E + 02	+
	SHPSO	**1.6559E + 02**	2.6366E + 01	
F3	PSO	1.9894E + 01	2.6250E-01	+
	RBF-PSO	1.8216E + 01	3.0406E-01	+
	RBF-GL	1.1781E + 01	9.5226E-01	+
	SA-COSO	1.5756E + 01	5.0245E-01	+
	SHPSO	**4.1134E + 00**	5.9247E-01	
F4	PSO	1.2162E + 03	1.1332E + 02	+
	RBF-PSO	6.2696E + 02	5.7795E + 01	+
	RBF-GL	**9.5063E-01**	4.7833E-02	−
	SA-COSO	6.3353E + 01	1.9021E + 01	+
	SHPSO	1.0704E + 00	2.0485E-02	
F5	PSO	1.9665E + 03	2.4489E + 02	+
	RBF-PSO	1.6276E + 03	2.1109E + 02	+
	RBF-GL	1.3569E + 03	1.3875E + 02	+
	SA-COSO	1.2731E + 03	1.1719E + 02	+
	SHPSO	**8.0173E + 02**	7.2252E + 01	
F6	PSO	8.7594E + 02	8.2484E + 01	+
	RBF-PSO	7.1796E + 02	5.6516E + 01	+
	RBF-GL	7.8464E + 02	8.8113E + 01	+
	SA-COSO	–	–	*
	SHPSO	**5.1619E + 02**	3.2060E + 01	
F7	PSO	1.4655E + 03	3.3238E + 01	+
	RBF-PSO	1.4595E + 03	4.0844E + 01	+
	RBF-GL	1.4177E + 03	9.2042E + 01	≈
	SA-COSO	**1.3657E + 03**	3.0867E + 01	−
	SHPSO	1.4198E + 03	3.8238E + 01	

on 50- and 100-dimentional benchmark problems. Tables 4 and 5 give the statistical results on problems with 50 and 100 dimensions, respectively. Since no experimental results of GPEME on F5 and F7 are reported in [22], we do not compare the performance on these two problems with dimension 50. From Table 4, we can see that SHPSO obtains better results with 1000 expensive fitness evaluations on 50-dimensional problems. From Table 5, we can also find that SHPSO achieves better results on all problems except F7. The reason might be that there is a large number of irregularly distributed local optima in the fitness landscape of F7 and the global optimum is located in a narrow basin [34], the RBF surrogate built on limited amount of data may smooth out the global optimum, especially when the dimension is high.

Figs. 3 and 4 provide the convergence profiles of SHPSO and the compared algorithms on 50- and 100-dimensional problems, respectively. From Figs. 3 and 4, we can see that similar to the performance on 30-dimensional problems, RBF-GL might also have converged to a local optimum on 50- and 100-dimnsional test problems. Although the convergence performance of SHPSO is worse than that of RBF-GL on 100-dimensional F1, SHPSO continuously improves while RBF-GL gets stuck in a local optimum. Nevertheless, both perform better than SA-COSO. These results evidence the importance of PSO in SHPSO in exploiting the promising regions where the current optimum is located. From Figs. 3(d) and 4(d), we can see that both RBF-GL and SHPSO converge much faster than others to the region where the global optimum is located on the Griewank function but has difficulties in accurately identifying the global optimum. This might be attributed to the fact that the global contour of the Griewank function looks like a single funnel and a global surrogate model can easily capture the fitness landscape, thereby effectively speeding up the search process. However, the Griewank function has many local optima near the global optimum. Consequently, the global surrogate model used in RBF-GL may not be able to accurately learn the position of the global optimum. Therefore, the RBF-GL algorithm can quickly converge to an optimum that is close to the global one, but has difficulties in finding global one. For the SHPSO algorithm, we can notice that there is a slight stagnation during the search of SHPSO on 50-dimensional Griewank function, which is shown in Fig. 3(d). However, a clear improvement can be seen on 100-dimensional F4 after 700 expensive fitness evaluations, as shown in Fig. 4(d). We surmise that when more data are collected around the optimum by SLPSO and PSO, a more accurate RBF model can be obtained near the optimum, which can help SHPSO escape from a local optimum and find a better optimum. To verify this hypothesis, we increase the maximum expensive fitness evaluations to 2000 for 50- and 100-dimensional F4. Fig. 5 shows the convergence

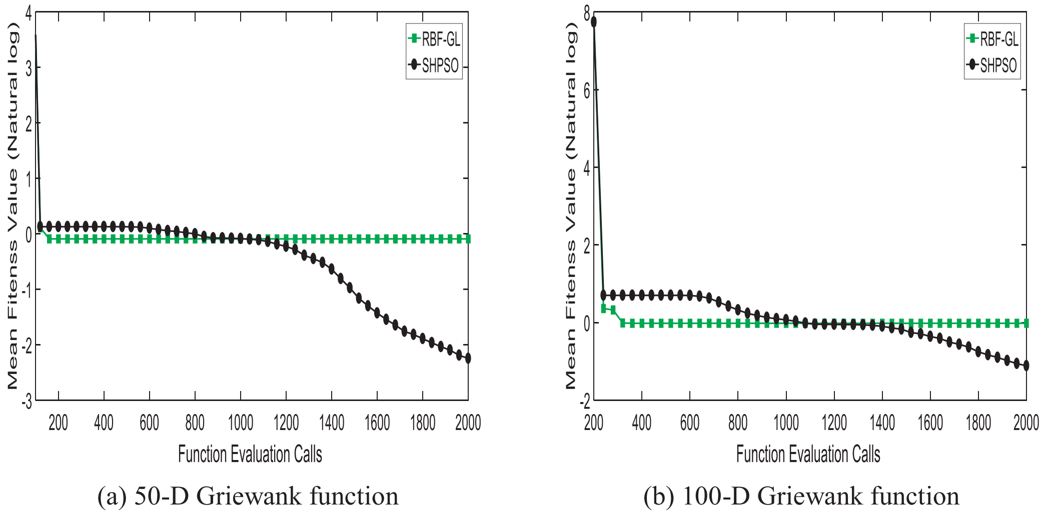


Fig. 5. Convergence profiles of algorithms SHPSO and RBF-GL on 50-dimensional (left) and 100-dimensional (right) Griewank function after 2000 expensive fitness evaluations.

profiles of SHPSO and RBF-GL, from which we can clearly see that SHPSO continues to improve while RBF-GL gets stuck in a local optimum after 1000 expensive fitness evaluations, which further indicates the importance of PSO in SHPSO.

5. CONCLUSION

In this paper, we propose a surrogate-assisted hierarchical particle swarm optimization algorithm, SHPSO, for computationally expensive optimization problems, in which the SL-PSO algorithm is embedded in the PSO framework to find the global optimum of the RBF surrogate model, thereby refining the local approximation of the fitness landscape around the optimum. Meanwhile, the PSO algorithm searches gradually in a wider space, enabling the surrogate to capture the global landscape of the fitness function. The proposed SHPSO algorithm is evaluated on seven widely used 30-, 50-, and 100-dimensional benchmark problems with different challenges. The experimental results show the effectiveness of our proposed algorithm in comparison to the state-of-the-art algorithms compared in this work.

SHPSO has much room for improvement, in particular in enhancing the surrogate's capability of capturing the global profile and local features of the fitness function by selecting training samples and developing dedicated learning algorithms. Future work also includes the extension of the proposed algorithm to large-scale and multi-objective optimization problems and verification of its performance in real-world problems.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [grant Nos. 61472269 and 61403272]; and the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, China; as well as the Fund Program for the Scientific Activities of Selected Returned Overseas Professionals in Shanxi Province; and Shanxi Province Science Foundation for Youths (201601D021083).

References

[1] D. Buche, N.N. Schraudolph, P. Koumoutsakos, Accelerating evolutionary algorithms with Gaussian process fitness function models, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 35 (2005) 183–194.
[2] X. Cai, H. Qiu, L. Gao, X. Shao, Metamodeling for high dimensional design problems by multi-fidelity simulations, Struct. Multidiscip. Optim. 56 (2017) 151–166.
[3] R. Cheng, Y. Jin, A social learning particle swarm optimization algorithm for scalable optimization, Inf. Sci. 291 (2015) 43–60.
[4] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, K. Sindhya, A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization, IEEE Trans. Evol. Comput. 22 (2018) 129–142.
[5] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), 1953, 1999, pp. 1–1957.
[6] A. Díaz-Manríquez, G. Toscano, C.A. Coello Coello, Comparison of metamodeling techniques in evolutionary algorithms, Soft Comput. (2016) 1–17.
[7] J. Eason, S. Cremaschi, Adaptive sequential sampling for surrogate model generation with artificial neural networks, Comput. Chem. Eng. 68 (2014) 220–232.
[8] R.C. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, in: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), 81, 2000, pp. 84–88.
[9] M.T.M. Emmerich, K.C. Giannakoglou, B. Naujoks, Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels, IEEE Trans. Evol. Comput. 10 (2006) 421–439.
[10] A.I.J. Forrester, A.J. Keane, Recent advances in surrogate-based optimization, Prog. Aerosp. Sci. 45 (2009) 50–79.

[11] L.J. Herrera, H. Pomares, I. Rojas, A. Guillén, G. Rubio, J. Urquiza, Global and local modelling in RBF networks, Neurocomputing 74 (2011) 2594–2602.
[12] R. Jin, W. Chen, T.W. Simpson, Comparative studies of metamodelling techniques under multiple modelling criteria, Struct. Multidiscip. Optim. 23 (2001) 1–13.
[13] Y. Jin, A comprehensive survey of fitness approximation in evolutionary computation, Soft Comput. 9 (2005) 3–12.
[14] Y. Jin, Surrogate-assisted evolutionary computation: recent advances and future challenges, Swarm Evol. Comput. 1 (2011) 61–70.
[15] Y. Jin, M. Olhofer, B. Sendhoff, A framework for evolutionary optimization with approximate fitness functions, IEEE Trans. Evol. Comput. 6 (2002) 481–494.
[16] A. Kattan, Y.-S. Ong, Surrogate genetic programming: a semantic aware evolutionary search, Inf. Sci. 296 (2015) 345–359.
[17] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Neural Networks, 1995. Proceedings., IEEE International Conference on, vol. 1944, 1995, pp. 1942–1948.
[18] J.P.C. Kleijnen, An overview of the design and analysis of simulation experiments for sensitivity analysis, Eur. J. Oper. Res. 164 (2005) 287–300.
[19] M.N. Le, Y.S. Ong, S. Menzel, Y. Jin, B. Sendhoff, Evolution by adapting surrogates, Evol. Comput. 21 (2013) 313–340.
[20] F.H. Lesh, Multi-dimensional least-squares polynomial curve fitting, Commun. ACM 2 (1959) 29–30.
[21] D. Lim, Y. Jin, Y.S. Ong, B. Sendhoff, Generalizing surrogate-assisted evolutionary computation, IEEE Trans. Evol. Comput. 14 (2010) 329–355.
[22] B. Liu, Q. Zhang, G.G.E. Gielen, A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems, IEEE Trans. Evol. Comput. 18 (2014) 180–192.
[23] H. Liu, Y.-S. Ong, J. Cai, A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design, Struct. Multidiscip. Optim. 57 (2018) 393–416.
[24] N. Lynn, P.N. Suganthan, Ensemble particle swarm optimizer, Appl. Soft Comput. 55 (2017) 533–548.
[25] B. Niu, F. Zhang, L. Li, L. Wu, Particle swarm optimization for yard truck scheduling in container terminal with a cooperative strategy, in: G. Leu, H.K. Singh, S. Elsayed (Eds.), Intelligent and Evolutionary Systems: The 20th Asia Pacific Symposium, IES 2016, Canberra, Australia, November 2016, Proceedings, Springer International Publishing, Cham, 2017, pp. 333–346.
[26] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, Y. Jin, A classification based surrogate-assisted evolutionary algorithm for expensive many-objective optimization, IEEE Trans. Evol. Comput. (2018) 1–1.
[27] M. Powell, Radial basis function methods for interpolation to functions of many variables, in: HERCMA, Citeseer, 2001, pp. 2–24.
[28] C. Praveen, R. Duvigneau, Low cost PSO using metamodels and inexact pre-evaluation: application to aerodynamic shape design, Comput. Methods Appl. Mech. Eng. 198 (2009) 1087–1096.
[29] H. Qiu, D. Wang, Y. Wang, Y. Yin, MRI appointment scheduling with uncertain examination time, J. Comb. Optim. (2017). https://doi.org/10.1007/s10878-017-0210-5.
[30] R.G. Regis, Particle swarm with radial basis function surrogates for expensive black-box optimization, J. Comput. Sci. 5 (2014) 12–23.
[31] J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, Design and analysis of computer experiments, Stat. Sci. 4 (1989) 409–423.
[32] S. Shan, G.G. Wang, Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black--box functions, Struct. Multidiscip. Optim. 41 (2010) 219–241.
[33] L. Shi, K. Rasheed, A survey of fitness approximation methods applied in evolutionary algorithms, in: Y. Tenne, C.-K. Goh (Eds.), Computational Intelligence in Expensive Optimization Problems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 3–28.
[34] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.-P. Chen, A. Auger, S. Tiwari, in: Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization, KanGAL, 2005, p. 2005. Report, 2005005.
[35] C. Sun, Y. Jin, R. Cheng, J. Ding, J. Zeng, Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems, IEEE Trans. Evol. Comput. 21 (2017) 644–660.
[36] C. Sun, Y. Jin, J. Zeng, Y. Yu, A two-layer surrogate-assisted particle swarm optimization algorithm, Soft Comput. 19 (2015) 1461–1475.
[37] C. Sun, J. Zeng, J. Pan, S. Xue, Y. Jin, A new fitness estimation strategy for particle swarm optimization, Inf. Sci. 221 (2013) 355–370.
[38] Y. Tang, J. Chen, J. Wei, A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions, Eng. Optim. 45 (2013) 557–576.
[39] M.H. Tayarani-N, X. Yao, H. Xu, Meta-heuristic algorithms in car engine design: a literature survey, IEEE Trans. Evol. Comput. 19 (2015) 609–629.
[40] Y. Tenne, S.W. Armfield, A framework for memetic optimization using variable global and local surrogate models, Soft Comput. 13 (2008) 781.
[41] D. Wang, F. Liu, Y. Wang, Y. Jin, A knowledge-based evolutionary proactive scheduling approach in the presence of machine breakdown and deterioration effect, Knowledge-Based Syst. 90 (2015) 70–80.
[42] D. Wang, Z. Wu, Y. Fei, W. Zhang, Structural design employing a sequential approximation optimization approach, Comput. Struct. 134 (2014) 75–87.
[43] G.G. Wang, S. Shan, Review of metamodeling techniques in support of engineering design optimization, J. Mech. Des. 129 (2006) 370–380.
[44] H. Wang, Y. Jin, J. Doherty, Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems, IEEE Trans. Cybern. 47 (2017) 2664–2677.
[45] Z. Wu, D. Wang, P. Okolo N, F. Hu, W. Zhang, Global sensitivity analysis using a Gaussian radial basis function metamodel, Reliab. Eng. Syst. Saf. 154 (2016) 171–179.
[46] Q. Yang, W.N. Chen, J.D. Deng, Y. Li, T. Gu, J. Zhang, A level-based learning swarm optimizer for large scale optimization, IEEE Trans. Evol. Comput. (2017) 1–1.
[47] A. Younis, Z. Dong, Trends, features, and tests of common and recently introduced global optimization methods, Eng. Optim. 42 (2010) 691–718.
[48] H. Yu, Y. Tan, C. Sun, J. Zeng, Y. Jin, An adaptive model selection strategy for surrogate-assisted particle swarm optimization algorithm, in: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), 2016, pp. 1–8.
[49] D. Zhao, D. Xue, Performance comparison of metamodeling methods from the perspective of sample quality merits, in: Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, 2009, pp. 699–709.
[50] Z. Zhou, Y.S. Ong, P.B. Nair, A.J. Keane, Y.L. Kai, Combining global and local surrogate models to accelerate evolutionary optimization, IEEE Trans. Syst. Man Cybern. Part C 37 (2007) 66–76.