Enhancing hierarchical surrogate-assisted evolutionary algorithm for high-dimensional expensive optimization via random projection

Xiaodong Ren¹ · Daofu Guo¹ · Zhigang Ren¹ · Yongsheng Liang¹ · An Chen¹

Received: 20 February 2021 / Accepted: 22 July 2021 / Published online: 31 July 2021
© The Author(s) 2021

Abstract

By remarkably reducing real fitness evaluations, surrogate-assisted evolutionary algorithms (SAEAs), especially hierarchical SAEAs, have been shown to be effective in solving computationally expensive optimization problems. The success of hierarchical SAEAs mainly profits from the potential benefit of their global surrogate models known as “blessing of uncertainty” and the high accuracy of local models. However, their performance leaves room for improvement on high-dimensional problems since now it is still challenging to build accurate enough local models due to the huge solution space. Directing against this issue, this study proposes a new hierarchical SAEA by training local surrogate models with the help of the random projection technique. Instead of executing training in the original high-dimensional solution space, the new algorithm first randomly projects training samples onto a set of low-dimensional subspaces, then trains a surrogate model in each subspace, and finally achieves evaluations of candidate solutions by averaging the resulting models. Experimental results on seven benchmark functions of 100 and 200 dimensions demonstrate that random projection can significantly improve the accuracy of local surrogate models and the new proposed hierarchical SAEA possesses an obvious edge over state-of-the-art SAEAs.

Keywords High-dimensional expensive optimization · Hierarchical surrogate-assisted evolutionary algorithm · Local surrogate model · Random projection

Introduction

Metaheuristic algorithms, such as differential evolution (DE) [1], genetic algorithm (GA) [2], and particle swarm optimization (PSO) [3], have been widely employed to solve real-world engineering optimization problems [4–6]. These metaheuristic algorithms generally require a large number of fitness evaluations (FEs) to find a satisfying solution, which make them unsuitable for computationally expensive optimization problems [7–10]. The reason consists in that a single FE of these expensive problems often consumes lots of time or material resources.

Surrogate model provides an effective tool to reduce the computational cost by partly replacing computationally expensive FEs during the evolution process. Over the past decades, several types of surrogate models, including polynomial regression [11], radial basis function (RBF)

[12–14], Gaussian process (GP) [15, 16], and support vector machine [17], have been developed and deeply analyzed, yielding various surrogate-assisted evolutionary algorithms (SAEAs). According to the role of the employed surrogate model, existing SAEAs can be roughly divided into the following three categories: (1) global SAEAs, (2) local SAEAs, and (3) hierarchical SAEAs. Global surrogate models aim to approximate the expensive fitness function in the whole solution space [18–22]. As a contrast, local surrogate models focus on a small solution region for the purpose of ensuring the approximation accuracy [23–26]. Fusing the exploration capability of the global surrogate and the exploitation capability of the local surrogate together, hierarchical SAEAs attracted much research attention in recent years and showed certain superiority over a single type of SAEAs on most expensive problems [27–30].

So far, SAEAs have achieved great success in tackling low- and medium-dimensional expensive problems, but they generally lose efficiency on high-dimensional problems due to the “curse of dimensionality” [31]. On the one hand, EAs cannot fully explore the huge solution space of a high-dimensional problem at the cost of acceptable computational

✉ Zhigang Ren
renzg@xjtu.edu.cn

¹ School of Automation Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China

resources. On the other hand, it is also impracticable to directly build an accurate enough surrogate model with a limited number of available training samples. Directing at the above two challenges, several beneficial attempts have been made. Tian et al. [32] developed a multiobjective sample infill criterion according to non-dominated sorting and enhanced the adaptability of a GP-assisted PSO algorithm. Li et al. [33] proposed a surrogate-assisted multiswarm optimization algorithm, where a swarm is specially evolved to enhance the exploration capability of the whole algorithm.

Besides, it has been verified that hierarchical SAEAs are of great potential in solving high-dimensional expensive problems [34–39]. Profiting from its “blessing of uncertainty” [40], the global surrogate model generally helps to smooth out some local optima and thus to reduce the search space, whereas the local surrogate model helps to identify better solutions in the located local promising regions. Taking a recently proposed hierarchical SAEA called evolutionary sampling assisted optimization (ESAO) [34] as an example, it builds a global RBF model to assist DE to conduct a global search by prescreening promising solutions. On the other side, it performs a local search by taking another RBF model trained in the neighborhood of the current best solution as the objective function. ESAO alternates the two types of search if one of them cannot lead to a better solution. By this means, it not only enhances the possibility of finding the global optimum but also speeds up the optimization process. Compared with the global surrogate model, the local model in a hierarchical SAEA is expected to be of much higher accuracy. Despite the reduced solution region, it is still a nontrivial task to build an accurate enough local surrogate model in a high-dimensional solution space.

To alleviate this issue, this study proposes a random projection-enhanced hierarchical SAEA (RPHSA). RPHSA inherits the fine framework of ESAO but adapts the local surrogate model therein, i.e., RBF, to high-dimensional problems with the random projection (RP) technique. As a commonly-used dimension reduction technique, RP can expediently project high-dimensional data onto subspaces of much lower dimensions while maintaining the geometric structure among the data to a great extent. Now, RP has been successfully applied in many fields such as signal processing [41], machine learning [42], and high-dimensional optimization [43], but has seldom been employed to train surrogate models for high-dimensional problems. With the introduction of RP, RPHSA builds its local RBF model as follows: first randomly project original training samples onto several low-dimensional subspaces, then train low-dimensional RBF models in respective subspaces so as to capture the characteristics of the original problem from different perspectives, and finally construct the final RP-based RBF (RP-RBF) by averaging all low-dimensional RBF models. In this way, the final RP-RBF is expected to achieve ideal approximation

capability with a small number of training samples and thus to enhance the performance of RPHSA.

The remainder of this paper is organized as follows. “[Related work](#)” briefly reviews the relevant work of SAEAs. “[Proposed RPHSA](#)” describes the proposed RP-RBF model and the resulting RPHSA algorithm in detail. “[Experimental studies](#)” reports experimental settings and results along with some analyses. The concluding remarks and future research directions are finally given in “[Conclusion](#)”.

Related work

SAEAs have been receiving more and more research attention in recent years for their effectiveness in solving computationally expensive problems. They replace most of the expensive real FEs with surrogate estimations in the optimization process, such that many computational resources can be avoided and the optimization performance can be greatly improved. Early SAEAs tend to employ global surrogate models to fit the whole optimization problem. Their excellent performance on simple and low-dimensional expensive optimization problems have been verified. Jin et al. [18] adopted an artificial neural network as a global surrogate to assist the covariance matrix adaptation evolution strategy and also proposed an empirical criterion to switch between expensive real FEs and cheap fitness estimations during the evolutionary search process. Ratle [19] suggested using GP to construct a global surrogate so as to guide the search process. Regis et al. [20] developed a SAEA based on PSO and a global RBF model, where the former generates multiple trial positions for each particle in each iteration, while the later prescreens the most promising trial positions to generate new particles. Liu et al. [21] employed a GP with lower confidence bound to select promising solutions in the evolution process of a DE algorithm. They also utilized Sammon mapping, a dimension reduction technique, to enhance the surrogate accuracy on medium-scale expensive optimization. Dong et al. [22] proposed a multi-start approach with the goal of finding all the local optima of a pre-trained global GP model. And the search was performed within the located local optimal solution regions.

Global surrogate models take effect on expensive problems of simple fitness landscape, but they suffer from “curse of dimensionality” and cannot adapt themselves well to complicated problems. To alleviate this issue, researchers developed local surrogate models to improve the approximation accuracy in local solution regions. Ong et al. [23] employed a trust-region method for interleaving the use of exact models for the objective and constraint functions with computationally cheap RBF models in the local search process. The idea of fitness inheritance suggested by Smith et al. [24] can also be seen as a local surrogate, where the fitness value of

an individual is estimated based on its neighbors and parents. Similarly, Sun et al. [25] proposed a fitness estimation strategy to approximate particles in PSO based on their positional relationship with other particles. Lu and Tang [26] integrated a local surrogate model into DE, where the model was used not only for regression but also for classification.

Compared with global surrogate models, local models are more likely to improve the solution quality, but they tend to lack the capability of jumping out of local optima. To achieve complementary advantages of these two types of models, many studies in recent years focused on developing hierarchical SAEAs by integrating global and local surrogates together. Zhou et al. [27] proposed a hierarchical SAEA, where a global GP and a local RBF network were jointly employed to assist a GA. The former was used to identify promising individuals at the global search level, while the later was adopted to accelerate the convergence of a trust-region-enabled search strategy at the local search level. Tenne and Armfield [28] proposed a memetic optimization framework consisting of variable global and local surrogates and employed RBF in a trust-region approach for expensive optimization. Sun et al. [29] introduced a two-layer surrogate-assisted PSO, where a global surrogate model was intended to smooth out some local optima of the original multimodal fitness functions and a local surrogate model was employed for fitness estimation. Inspired by committee-based active learning, Wang et al. [30] proposed an ensemble of several global surrogates to search for the best and most uncertain solutions to be evaluated by the real fitness function and employed a local surrogate to model the neighborhood of the current best solution with the goal of further improving it by optimizing the model.

As for high-dimensional expensive problems, they have become a research hotspot in the field of SAEAs. In addition to the ESAO algorithm introduced in "Introduction", some other significant research efforts have been made to push the boundary of SAEAs in solving this kind of problems. Tian et al. [32] revealed that the approximation uncertainty of GP becomes less reliable on high-dimensional problems and the commonly-used scalar sample infill criteria, which combines the approximated fitness and the approximation uncertainty in a scalar function, tends to lose efficacy. To overcome this defect, they developed a multiobjective sample infill criterion by considering the above two factors as separate objectives and selecting promising solutions according to non-dominated sorting, and thereby achieved a good balance between exploitation and exploration. Li et al. [33] proposed an SAEA involving two swarms, where the first one uses the learner phase of teaching–learning-based optimization to enhance exploration and the second one uses PSO for faster convergence. Moreover, they also designed a novel sample infill criterion by selecting particles predicted to be with self-improvement for real FEs. Sun et al. [35] proposed

a surrogate-assisted cooperative swarm optimization algorithm, where an RBF-assisted social learning PSO focuses on exploration and a fitness estimation strategy-assisted PSO concentrates on local search. Yu et al. [36] embedded the social learning PSO into an RBF-assisted PSO framework. The former aims to find the optimum of the RBF model, which is constructed with a certain number of the best solutions found so far, and thereby refines its local approximation of the fitness landscape around the optimum. The latter conducts a search in a wider solution region, enabling the RBF model to capture the global landscape of the fitness function. Yang et al. [37] developed a two-layer surrogate-assisted DE algorithm. This algorithm measures its evolutionary status according to the improvement times of the best solution. According to the feedback status, three different DE mutation operators are employed to generate new offspring, which are further prescreened by a global or local GP model. Dong et al. [38] proposed a surrogate-assisted grey wolf optimization algorithm. It conducts a global search and a multi-start local search based on a dynamically updated RBF model. The resulting promising solutions are further used to guide the whole wolf pack such that the design space can be properly explored. Cai et al. [39] employed both global and local surrogates to assist the search process of GA. They specially designed a surrogate-assisted updating mechanism for GA. For a parent solution, this mechanism first builds a surrogate with all its evaluated neighbors, and then replaces it with the predicted optimum of the surrogate.

These recent studies enhance the adaptability of SAEAs on high-dimensional expensive problems. However, it is still an open problem to build accurate enough surrogate models, especially local models in hierarchical SAEAs, for high-dimensional problems. This study attempts to tackle this issue with the RP technique.

Proposed RPHSA

This section first introduces how to scale up RBF to high-dimensional problems with the RP technique, then discusses the integration of the resulting RP-RBF model and the basic ESAO algorithm, and finally presents the implementation of the proposed RPHSA algorithm.

RP-RBF model

As a commonly-used surrogate model, RBF has been shown to fit nonlinear functions well and be capable of both local and global modeling [20, 23, 34–36].

Given a set $X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \in \mathbb{R}^{d \times n}$ of n distinct training samples in $\mathbb{R}^d$, where $\mathbf{x}_i$ is a d -dimensional vector with its fitness value being $f(\mathbf{x}_i)$, $i = 1, 2, \dots, n$, RBF approximates the fitness value of a new solution $\mathbf{x}$ as

$$\hat{f}(x) = \sum_{i=1}^n \omega_i \varphi(\|x - x_i\|), \quad (1)$$

where $\|\cdot\|$, $\varphi(\cdot)$, and ω_i denote the Euclidean norm, the basis function, and the weight coefficient to be learnt, respectively. There are several types of kernel functions such as multiquadric kernel, linear kernel, and Gaussian kernel. In this study, the multiquadric kernel $\phi(r) = \sqrt{r^2 + c^2}$ with the bias term being 1 is used to construct RBF. Compared with other types of surrogate models such as GP, RBF is relatively easy to train. Let Φ be an $n \times n$ matrix with each element $\Phi_{ij} = \varphi(\|x_i - x_j\|)$ and f denote the vector $(f(x_1), f(x_2), \dots, f(x_n))^T$, then the unknown coefficients $\omega = (\omega_1, \omega_2, \dots, \omega_n)^T$ of RBF can be obtained by solving the following linear equation:

$$\omega = \Phi^{-1}f. \quad (2)$$

Although the performance of RBF is relatively insensitive to problem dimension, its accuracy on high-dimensional problems is still difficult to be guaranteed due to the huge modeling space and the very limited number of available training samples. To cope with this dilemma, a promising way is to reduce the modeling space with a dimension reduction technique. This study selects RP to execute this task for its simplicity and capability of preserving the geometric structure among training samples [44].

To perform projection with RP, a random projection matrix independent of training data is required. According to [45], the Gaussian random projection matrix, each of whose elements obeys Gaussian distribution $N(0, \sigma^2)$, meets this requirement. In general, the variance σ^2 is set to $1/k$ with k representing the space dimension after projection. To ensure the orthogonality required by RP, we further perform row orthogonalization on the initially generated Gaussian matrix. Let $P \in \mathbb{R}^{k \times d}$ denote the final projection matrix, then a set of k -dimensional training samples can be obtained by performing projection on the original d -dimensional training samples as follows:

$$X' = PX = (x'_1, x'_2, \dots, x'_n) \in \mathbb{R}^{k \times n}. \quad (3)$$

Taking $X' = (x'_1, x'_2, \dots, x'_n)$ as new training samples, a low-dimensional RBF model $\hat{f}(Px)$ can be generated according to (1) and (2).

The excellent dimension reduction property $k \geq k_0 = O(\log(n)/\epsilon^2)$ of RP can be theoretically guaranteed by the following Johnson–Lindenstrauss Lemma [45]:

Lemma 1 (Johnson–Lindenstrauss Lemma): *Given $0 < \epsilon < 1$, a set X of n points in $\mathbb{R}^d$, and a positive integer k satisfying, there exists a linear map $P : \mathbb{R}^d \rightarrow \mathbb{R}^k$ such that:*

$$(1 - \epsilon)\|x_i - x_j\|^2 \leq \|P(x_i) - P(x_j)\|^2 \leq (1 + \epsilon)\|x_i - x_j\|^2, \quad (4)$$

for $\forall x_i, x_j \in X$.

This lemma suggests that the distance between any two points in high-dimensional Euclidean space, i.e., the geometric structure of the original training data, can be preserved after projection within a certain error range, which depends on the dimension of the new space. The lower the new dimension is, the more greatly the difficulty of training an RBF can be reduced, but the broader the error range tends to be, which is harmful to the accuracy of the trained RBF model. To balance this contradiction, the developed RP-RBF model first projects the original training samples onto a group of low-dimensional random subspaces instead of a single one and then trains an RBF in each subspace. Note that we just conduct projection in decision space and each low-dimensional training sample inherits the fitness value of the corresponding original sample. In this way, the original training samples can be shared in different subspaces and a low-dimensional RBF capturing part of the characteristics of the original problem can be easily trained in each subspace. By averaging all low-dimensional RBFs, the final RP-RBF is expected to be able to learn more characteristics of the original problem and to achieve higher accuracy. Figure 1 illustrates the construction process of RP-RBF, where m denotes the number of subspaces, $P_i \in \mathbb{R}^{k \times d}$ denotes the i th projection matrix, and $\hat{f}_i(P_i x)$ denotes the RBF model trained in the i th subspace. Note that x is an arbitrary d -dimensional solution to be estimated, and $P_i x$ denotes its k -dimensional projection in the i th subspace. According to the idea described above, the final RP-RBF model can be formulated as

$$\hat{f}(x) \triangleq \sum_{i=1}^m \hat{f}_i(P_i x) / m. \quad (5)$$

Algorithm 1 presents the pseudocode for constructing RP-RBF, where steps 2–4 show the specific process of

Fig. 1 Construction process of RP-RBF

building an RBF in each subspace. It is notable that the subspace dimension k , the subspace number m , and the sample number n have an important influence on the accuracy of RP-RBF. We will empirically analyze their influence in the experimental section.

As for the computational complexity of constructing an RP-RBF model, it is determined by the computational complexity of generating projection matrices, projecting training samples, and building RBFs with low-dimensional samples. To generate and orthogonalize a $k \times d$ random projection matrix, the computational complexity is $O(kd^2)$. To project n pieces of d -dimensional samples onto a k -dimensional subspace, the computational complexity is $O(nkd)$. The computational complexity of training a k -dimensional RBF model with n samples is $O(n^3)$ computational complexity of training a k -dimensional RBF model with RP is $O(kd^2 + nkd + n^3)$. Since RP-RBF concerns m initial low-dimensional RBF models, the total computational complexity of constructing RP-RBF is $O(mkd^2 + mnkd + mn^3)$. According to the recommendation in [43], m is generally set to $4\lceil d/k \rceil$. Besides, the parameters n and k are linearly proportional to d and satisfy $k < n < d$ in general. So the expression of total computational complexity $O(mkd^2 + mnkd + mn^3)$ can be reduced to $O(d^3)$. Note that the computational complexity of training an RBF model in the original d -dimensional space with n_0 samples is $O(n_0^3)$, where n_0 is linearly greater than d . Therefore, the process of constructing RP-RBF shares the same computational complexity, i.e., $O(d^3)$, with the process of training an RBF model in the original solution space.

and crossover operator. The offspring with the lowest prediction will be further evaluated with the real fitness function and will replace its parent if it is better. ESAO will also employ this offspring to update the current best solution and continue the global search process if it is of better fitness, otherwise, ESAO will switch to the local search process. The local RBF is trained with a certain number of best solutions that have been truly evaluated. ESAO takes it as an approximate fitness function and employs DE to find its optimum. This optimum will undergo real evaluation. If it is better than the current best solution, ESAO will start a new local process after updating the current best solution with it and adding it to the population of the global process, otherwise, ESAO will return to the global search process.

A straightforward way to integrate RP-RBF and ESAO is to build both the global and local surrogate model for ESAO with RP-RBF. However, the proposed RPHSA tends to just replace the local model with RP-RBF. The reasons are threefold. First, the local RBF model aims to capture fitness landscape details of the current promising local solution region. It is directly used as the fitness function for local search and thereby requires much on estimation accuracy. On the contrary, the global RBF model is designed to describe all explored solution regions. It is allowed to be of certain estimation error such that some local optima can be smoothed out. Second, the structural characteristics of a local high-dimensional solution region are relatively simple and are more likely to be preserved and learned in low-dimensional subspaces. As a contrast, the structural

Algorithm 1: Pseudocode of constructing RP-RBF

Input: training samples $X \in \mathbb{R}^{d \times n}$ with corresponding fitness vector $f \in \mathbb{R}^n$, subspace dimension k , and subspace number m ;

Output: RP-RBF model $\hat{f}(x)$.

1. **for** $i = 1$ to m
 2. Generate a random projection matrix $P_i \in \mathbb{R}^{k \times d}$ by performing column orthogonalization on a Gaussian matrix;
 3. Generate the i th group of training samples X'_i by performing projection with P_i according to (3);
 4. Construct an RBF model $\hat{f}_i(P_i x)$ based on X'_i and f according to (1) and (2);
 5. Generate the final RP-RBF model $\hat{f}(x)$ by averaging all low-dimensional models according to (5).
-

Integration of RP-RBF and ESAO

As a recently developed hierarchical SAEA, ESAO maintains a global and a local RBF model to assist the global and the local search conducted by DE, respectively [34]. The global RBF is trained in each generation of DE with all truly evaluated solutions. It is employed to predict the fitness values of all offspring generated by DE through mutation

characteristics of the solution regions covered by the global RBF model are much more complicated and can hardly be modeled with high enough accuracy. Finally, experimental results in [34] reveal that more real FEs are conducted in the local search process than in the global one, which means that ESAO depends more on the local search process in seeking high-quality solutions. Therefore, it is hopeful to achieve

better optimization performance by adopting RP-RBF as the local surrogate model in ESAO.

Taking the basic framework of ESAO, Algorithm 2 presents the pseudocode of RPHSA. As suggested by ESAO, step 1 initializes the population for global search by optimal Latin hypercube sampling (OLHS) [46]. Steps 3–13 execute the global search and steps 14–22 conducts the local search. The two types of search alternate if they fail to find a better

solution. RPHSA terminates its optimization process when meeting a stopping criterion, which is generally set as a maximum number of real FEs. It is notable that when locating the optimum of $\hat{f}_{\text{local}}(\mathbf{x})$ with DE in step 17, we still conduct optimization operations in the original d -dimensional solution space, but evaluate each candidate solution $\mathbf{x}$ by projecting it into multiple k -dimensional subspaces as required by (5).

Algorithm 2: Pseudocode of RPHSA

Input: subspace dimension k and subspace number m ;

Output: the current best solution $\mathbf{x}_b$ and its true fitness function value $f(\mathbf{x}_b)$.

1. Generate the initial population using OLHS and evaluate each individual therein with the real function $f(\mathbf{x})$;
 2. Initialize the current best solution $\mathbf{x}_b$ with the best solution in the current population;
 3. **While** stopping criterion is not met **do**
 4. Generate offspring by performing mutation and crossover operations of DE on the current population;
 5. Build a global RBF model $\hat{f}_{\text{global}}(\mathbf{x})$ with all truly evaluated solutions;
 6. Predict the fitness values of all offspring using $\hat{f}_{\text{global}}(\mathbf{x})$;
 7. Find out the offspring with the lowest prediction $\mathbf{x}_g$ and evaluate it with $f(\mathbf{x})$;
 8. **if** $f(\mathbf{x}_g) < f(\mathbf{x}_{gp})$ with $\mathbf{x}_{gp}$ being the parent of $\mathbf{x}_g$
 9. Replace $\mathbf{x}_{gp}$ with $\mathbf{x}_g$;
 10. **if** $f(\mathbf{x}_g) < f(\mathbf{x}_b)$
 11. Replace $\mathbf{x}_b$ with $\mathbf{x}_g$;
 12. **else**
 13. | Go to step 14;
 14. **While** stopping criterion is not met **do**
 15. Find out n best solutions that have been truly evaluated;
 16. Build a local RBF model $\hat{f}_{\text{local}}(\mathbf{x})$ by running **Algorithm 1**;
 17. Locate the optimum $\mathbf{x}_l$ of $\hat{f}_{\text{local}}(\mathbf{x})$ by taking DE as optimizer;
 18. Evaluate $\mathbf{x}_l$ with $f(\mathbf{x})$;
 19. **if** $f(\mathbf{x}_l) < f(\mathbf{x}_b)$
 20. Replace $\mathbf{x}_b$ with $\mathbf{x}_l$ and add $\mathbf{x}_l$ to the population for global search;
 21. **else**
 22. | Go to step 3.
-

Table 1 Characteristics of benchmark functions

Function	Description	Property	Global optimum
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 [47])	Very complicated multimodal	− 330
F6	Rotated hybrid composition function (F16 in [47])	Very complicated multimodal	120
F7	Rotated hybrid composition function (F19 in [47])	Very complicated multimodal	10

Experimental studies

To investigate the effectiveness and efficiency of RPHSA, we tested it on seven widely-used benchmark functions and empirically compared it with ESAO and some other state-of-the-art SAEAs for high-dimensional expensive problems. Table 1 lists the characteristics of these benchmark functions. They can all be set to 100 dimensions (100-D) or 200 dimensions (200-D). In our experiments, each algorithm was required to independently run 30 times on each function, and the maximum number of real FEs for each run was set to 1000 for both 100-D and 200-D functions. The results reported below are all average ones on 30 runs.

Influence of parameters

Compared with ESAO, RPHSA introduces two new parameters, i.e., the dimension (k) and number (m) of subspaces used for training RP-RBF. As revealed by Lemma 1 in "RP-RBF model", the value of k depends on the number (n) of training samples to be projected and the allowed projection error. A large value of k is beneficial to keeping the geometric structure among training samples and thus to reducing projection error, but simultaneously limits the difficulty reduction of training an RBF. Therefore, we tested the joint influence of k and n on the performance of RPHSA. The candidate values $k \in \{20, 30, 40, 50, 60\}$ and $n \in \{50, 100, 150, 200\}$ were considered in the experiment. As for m , a proper value is helpful to preserve the characteristics of the original problem. However, a too large value of m will increase the computational burden for building RP-RBF. According to the recommendation in [43], we directly set $m = 4 \lceil d/k \rceil$ as it is large enough to ensure the effectiveness of RP. The other parameters of RPHSA were strictly set to the same values as the corresponding ones in ESAO.

Figure 2 shows the performance variation of RPHSA with respect to different combinations of k and n , where the seven 100-D functions are taken as examples. It can be seen that k and n show a similar joint influence on the performance of RPHSA on F1–F4. A larger value of k and a smaller value of n help PRHSA to achieve better performance, and the

combination of $k=50$ and $n=100$ could achieve the overall best results on these four functions. In contrast, the RP-RBF model in RPHSA requires much fewer training samples than the traditional RBF models, which generally need at least $2d$ samples [34]. This merit mainly profits from the dimension reduction capability of RP. As for k , when it is set to a value within $\{40, 50, 60\}$, RPHSA demonstrates similar and acceptable performance. However, the performance of RPHSA significantly deteriorates when k becomes smaller. This is understandable because a too small value of k will make the original high-dimensional training samples lose too much structural information during projection, which will further reduce the approximation accuracy of RP-RBF.

As for the three more complicated functions F5–F7, the performance of RPHSA becomes less insensitive to k and requires sharply different values of n for different functions. For F5 and F6, it obtains the best results when n is 200; while for F7, it obtains the best result when n is 50. In our opinion, a proper parameter combination is determined by the structural characteristics of the function solution space. Since it is hard to get the specific characteristics of the problem to be solved in advance, the combination of $k=50$ and $n=100$ is a good trade-off among the seven test functions. We also tested RPHSA on the seven functions of 200 dimensions. RPHSA shows similar performance variation with respect to different combinations of k and n on 200-D functions as it does on 100-D ones, and the combination of $k=50$ and $n=100$ also enables RPHSA to gain comprehensive superiority. Therefore, we set this combination as the default setting of RPHSA and employ it in the following experiments.

Effectiveness of RP-RBF

To verify the effectiveness of the developed RP-RBF model, we empirically compared it with the traditional RBF model in terms of approximation accuracy and capability of enhancing optimization performance.

1. Approximation accuracy. In this experiment, we specially built a traditional local RBF model in RPHSA

Fig. 2 Mean fitness values obtained by RPHSA with different combinations of k and n on 100-D functions

Fig. 3 Fitness evaluation of individuals in a population of DE that conducts a local search on 100-D functions

besides RP-RBF. This model was strictly built according to the method described in ESAO. It did not participate in any algorithmic operation but being used for com-

parison. In the middle or late evolution stage of RPHSA, we picked out a population of DE that conducts local search and evaluated each individual therein by real fit-

Table 2 Optimization results of ESAO and RPHSA on 100-D functions

Function	ESAO	RPHSA
F1	1.28E+03±1.34E+02	3.48E+01±1.13E+01
F2	5.79E+02±4.48E+01	1.20E+02±2.19E+01
F3	1.04E+01±2.11E-01	3.06E+00±2.94E-01
F4	5.73E+01±5.84E+00	1.79E+00±2.51E-01
F5	7.13E+02±2.65E+01	7.83E+02±6.72E+01
F6	5.38E+02±3.82E+01	5.23E+02±3.55E+01
F7	1.37E+03±2.75E+01	1.35E+03±2.08E+01

ness function, the traditional RBF, and RP-RBF. Figure 3 presents the evaluation results, where the abscissa indicates individuals sorted in ascending order according to their real fitness values.

It can be observed that compared with the traditional RBF, the curves of RP-RBF on F1–F4 are more similar to the ones of real FE, and they are also closer to the latter. The superiority of RP-RBF over the traditional RBF on F5–F7 is not so obvious as it shows on F1–F4, but its estimation errors are still much smaller than the corresponding ones of the traditional RBF. This indicates that RP-RBF has higher approximation accuracy than the traditional RBF and thus is more likely to help the optimizer find high-quality solutions. Overall, the superiority of RP-RBF also verifies the

effectiveness of RP in reducing the burden of high-dimensional modeling.

- Optimization performance. We verified the superiority of RP-RBF in enhancing optimization performance by directly comparing the optimization results of RPHSA and ESAO since the difference of these two algorithms only lies in the local surrogate model. Table 2 reports the results on seven 100-D functions, where the mean, the standard deviation. The results in bold indicate that they are the better ones, according to the result of Wilcoxon rank-sum test at a significance level of 0.05. It can be observed from Table 2 that RPHSA significantly outperforms ESAO on all benchmark functions except F5. It improves the results of ESAO on F1, F3, and F4 by at least an order of magnitude in terms of the mean. It is notable that the difference between the optimization results obtained by ESAO and RPHSA on F5–F7 are not very obvious, which is consistent with the results shown in Fig. 3. Nevertheless, compared to ESAO, RPHSA could still gain a slight advantage on F6 and F7.

To gain a further insight into the difference between RP-RBF and the traditional RBF, we specially calculated the total number of local searches (NLS), the number of true improvements (NTI) of local searches, and their ratios when conducting RPHSA and ESAO. Table 3 reports the average result on 30 independent runs on each 100-D function.

Table 3 Average NLS, NTI, and NTI/NLS of ESAO and RPHSA on 100-D functions

Function	ESAO			RPHSA		
	NLS	NTI	NTI/NLS	NLS	NTI	NTI/NLS
F1	414.3	30.1	7.3%	459.4	120.0	26.1%
F2	416.3	33.9	8.1%	461.3	123.7	26.8%
F3	414.8	31.2	7.5%	438.1	77.4	17.7%
F4	418.1	38.3	9.2%	458.7	119.8	26.1%
F5	406.7	14.0	3.5%	408.0	13.7	3.4%
F6	409.0	19.1	4.7%	409.9	22.0	5.4%
F7	403.2	7.7	1.9%	404.1	9.2	2.3%

Table 4 Optimization results of RPHSA with different RBF kernels on 100-D functions

Function	Multiquadric	Linear	Cubic	Gaussian	Thin plate spline
F1	3.48E+01±1.13E+01	4.86E+01±1.24E+01	9.38E+03±1.18E+03	5.22E+01±1.23E+01	9.74E+03±9.01E+02
F2	1.20E+02±2.19E+01	1.57E+02±5.66E+01	5.38E+03±9.11E+02	2.11E+02±4.76E+01	5.87E+03±8.41E+02
F3	3.06E+00±2.94E-01	3.63E+00±4.48E-01	1.91E+01±2.39E-01	3.34E+00±4.32E-01	1.84E+01±2.92E-01
F4	1.79E+00±2.51E-01	2.11E+00±4.46E-01	6.88E+02±4.76E+01	2.07E+00±3.76E-01	7.30E+02±7.93E+01
F5	7.83E+02±6.72E+01	8.23E+02±7.49E+01	1.46E+03±1.83E+02	8.45E+02±6.98E+01	1.49E+03±9.86E+01
F6	1.35E+03±2.08E+01	1.32E+03±2.21E+01	1.47E+03±2.30E+01	1.34E+03±1.98E+01	1.46E+03±2.94E+01
F7	5.12E+02±3.13E+01	5.69E+02±3.12E+01	8.10E+02±3.63E+01	6.08E+02±4.05E+01	8.09E+02±5.67E+01

It indicates that RPHSA has an obvious edge over ESAO in terms of all three indicators on all tested functions. In particular, RPHSA outperforms ESAO by several times in terms of NTI and NTI/NLS on the first four functions, which is consistent with the results shown in Table 2. These results reveal that the local search of RPHSA is more efficient than that of ESAO, which benefits from the higher approximation accuracy of RP-RBF.

3. Generality of the RP technique. To verify the generality of the RP technique, we tested the performance of RPHSA with different RBF kernel functions. Besides the default multiquadric kernel, the other four types of kernels, including linear kernel, cubic kernel, Gaussian kernel, and thin-plate spline kernel, were employed in the experiment. They can be formulated as follows:

$$\text{Multiquadric kernel: } \phi(r) = \sqrt{r^2 + c^2}, \quad (6)$$

$$\text{Linear kernel: } \phi(r) = r, \quad (7)$$

$$\text{Cubic kernel: } \phi(r) = r^3, \quad (8)$$

$$\text{Gaussian kernel: } \phi(r) = e^{-\frac{r^2}{2\sigma^2}}, \quad (9)$$

$$\text{Thin plate spline kernel: } \phi(r) = r^2 \ln(r), \quad (10)$$

where r denotes the Euclidean distance from a solution $\mathbf{x}$ to be evaluated to a training sample. In the experiment, the bias term c in (6) and the kernel width σ in (9) are both set to 1. Table 4 presents the optimization results, where the mean and the standard deviation are reported. The results in bold indicate that they are the best ones. It can be observed that RPHSA with multiquadric kernel achieves the best results on all test functions except F6. RPHSAs with linear and Gaussian kernels perform a little worse than the version with multiquadric kernel, but also get acceptable results. This verifies the generality of the RP technique. As for cubic and thin plate spline kernels, which are seldom employed to tackle high-dimensional problems, they significantly deteriorates the performance of RPHSA. The default version of RPHSA uses the multiquadric kernel. One reason consists in its excellent performance, and the other is that the baseline algorithm of RPHSA, i.e., ESAO, also employs this kernel.

Table 5 Optimization results of the five algorithms on 100-D functions

Function	ESAO	SA-COSO	SHPSO	SAMSO	RPHSA
F1	1.28E+03±1.04E+02 ⁺	9.66E+02±2.67E+02 ⁺	1.19E+02±3.34E+01 ⁺	6.47E+01±1.41E+01 ⁺	3.48E+01±1.13E+01
F2	5.67E+02±4.16E+01 ⁺	2.57E+03±8.71E+02 ⁺	2.00E+02±3.92E+01 ⁺	2.04E+02±3.26E+01 ⁺	1.20E+02±2.19E+01
F3	1.05E+01±2.88E−01 ⁺	1.58E+01±6.57E−01 ⁺	5.47E+00±7.40E−01 ⁺	6.00E+00±3.57E−01 ⁺	3.06E+00±2.94E−01
F4	5.74E+01±4.09E+00 ⁺	6.07E+01±1.60E+01 ⁺	1.12E+00±4.96E−02 [−]	1.07E+00±1.55E−02[−]	1.80E+00±2.51E−01
F5	7.25E+02±3.45E+01[−]	1.27E+03±1.25E+02 ⁺	8.16E+02±5.09E+01 ⁺	7.67E+02±4.42E+01 [−]	7.83E+02±6.72E+01
F6	5.60E+02±3.55E+01 ⁺	8.27E+02±4.90E+01 ⁺	5.44E+02±2.29E+01 ⁺	5.07E+02±2.10E+01[−]	5.39E+02±1.29E+01
F7	1.38E+03±2.80E+01 ⁺	1.36E+03±3.39E+01 ⁺	1.42E+03±1.56E+01 ⁺	1.29E+03±3.87E+01[−]	1.35E+03±2.08E+01
+/-/≈	6/1/0	7/0/0	6/1/0	3/4/0	—

Table 6 Optimization results of the four algorithms on 200-D functions

Function	ESAO	SA-COSO	SAMSO	RPHSA
F1	1.79E+04±1.20E+03 ⁺	1.70E+04±2.74E+03 ⁺	1.50E+03±2.29E+02[−]	1.62E+03±2.40E+02
F2	4.47E+03±3.86E+02 ⁺	1.70E+04±4.59E+03 ⁺	1.28E+03±3.98E+02 ⁺	6.29E+02±8.39E+01
F3	1.48E+01±2.10E−01 ⁺	1.81E+01±3.14E−01 ⁺	1.21E+01±3.89E−01 ⁺	6.39E+00±2.39E−01
F4	5.61E+02±2.56E+01 ⁺	5.49E+02±9.30E+01 ⁺	8.84E+00±1.25E+00[−]	7.48E+01±1.37E+01
F5	5.34E+03±1.76E+02 ⁺	4.18E+03±2.18E+02[−]	4.95E+03±1.26E+02 ⁺	4.92E+03±1.14E+02
F6	1.66E+03±4.59E+01 ⁺	1.68E+03±2.57E+01 ⁺	1.66E+03±5.10E+01 ⁺	1.64E+03±3.23E+01
F7	1.45E+03±1.75E+01 ⁺	1.38E+03±1.89E+01 ⁺	1.34E+03±3.07E+01 ⁺	1.29E+03±4.91E+01
+/-/≈	7/0/0	6/1/0	5/2/0	—

Fig. 4 Convergence curves of the five algorithms on 100-D functions

Comparison with state-of-the-art algorithms

To comprehensively verify the efficiency of RPHSA in solving high-dimensional expensive problems, we compared it with four state-of-the-art algorithms, including ESAO [34], SA-COSO [35], SHPSO [36], and SAMSO [33] by testing them on all seven functions of 100 and 200 dimensions. The four algorithms are recently published promising algorithms for high-dimensional expensive problems. To ensure the fairness of the comparison, the results of the four competitors were all obtained by running their source codes with recommended parameter settings. Tables 5 and 6 report the final optimization results generated by the total five algorithms on 100-D and 200-D functions, respectively, where the best results are highlighted in bold. And the symbols “+”, “−” and “≈” indicate that the result of RPHSA is better than, worse than, and similar to the corresponding one of the comparison algorithm according to Wilcoxon rank test. Note that the performance of SHPSO on 200-D problems

was not investigated in the original paper [36], so we did not compare it with RPHSA on 200-D functions. According to the results in Tables 5 and 6, the following observations can be made:

1. 100-D functions: RPHSA demonstrates competitive performance among all involved algorithms on 100-D functions. It outperforms ESAO, SA-COSO, and SHPSO on all functions except being surpassed by ESAO and SHPSO on F5 and F4, respectively. Compared with SAMSO, RPHSA wins obvious superiority on F1–F3, but shows some weakness on F4–F7. On the whole, RPHSA can be ranked second on 100-D functions.
2. 200-D functions: When the function dimension increases from 100 to 200, the performance of all the four algorithms listed in Table 6 degenerates to a certain extent. Despite this fact, RPHSA has the smallest performance degeneration and shows more obvious superiority over its competitors on most functions. It definitely defeats

Fig. 5 Convergence curves of the four algorithms on 200-D functions

ESAO and SA-COSO on all seven functions except being surpassed by SA-COSO on F5. It is worth noting that compared with SAMSO, RPHSA obtains better solutions for the three complicated multimodal functions, i.e., F5–F7, and outperforms the former on 5 out of total 7 functions. This reveals the good scalability of RPHSA, which should be mainly attributed to the dimension reduction capability of RP.

It is notable that almost all solutions obtained by involved algorithms for each of the three complicated multimodal functions share the same order of magnitude, although Wilcoxon rank-sum test reveals that these algorithms really show significantly different performance on this type of functions. This confirms the challenge in solving them. For this type of functions, even in a local solution region, their structural characteristics tend to be very complex and are

very hard for the traditional RBF and also RP-RBF to fit. This limits the superiority of RPHSA to a certain extent.

To further demonstrate the performance differences among the five algorithms, we present their convergence curves on 100-D and 200-D functions in Figs. 4 and 5, respectively. It can be seen that no matter the function dimension is 100 or 200, RPHSA always keeps a more stable improvement tendency and continuously finds new better solutions during the whole evolution process. Its convergence rate is also faster than those of other algorithms on most functions, leading to better final optimization results.

Conclusion

In this paper, an RP-enhanced hierarchical SAEA, i.e., RPHSA, is proposed to solve high-dimensional computationally expensive optimization problems. RPHSA inherits

the fine framework of ESAO, but builds the local surrogate model therein, i.e., RBF, with the help of RP and develops a new local model named RP-RBF. Different from ESAO which directly trains its local model in the original high-dimensional space, RPHSA trains a group of RBFs in their respective subspaces generated by RP and constructs the final RP-RBF by averaging all low-dimensional RBFs. With the introduction of RP, not only the main characteristics of the original high-dimensional problem can be preserved to a large extent, but also the number of samples required for modeling can be significantly reduced. Experimental results on seven 100-D and 200-D functions indicate that RP-RBF has higher approximation accuracy and thus greatly enhances the optimization capability of RPHSA. Compared with four state-of-the-art SAEAs, RPHSA presents obvious superiority in solution quality, scalability, and convergence performance.

In future work, we will attempt to apply RPHSA to solve some real-world high-dimensional expensive problems. It is also interesting to scale up to other kinds of surrogate models such as GP with RP under the framework of RPHSA.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 61873199 and in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2020JM-059. Besides, the authors would like to thank Xinjing Wang, Chaoli Sun, Haibo Yu, and Fan Li for providing source codes of ESAO, SA-COSO, SHPSO, and SAMSO, respectively.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Price K, Storn RM, Lampinen JA (2005) Differential evolution: a practical approach to global optimization. Springer, Berlin
- Mitchell M (1998) An introduction to genetic algorithms. MIT Press, Cambridge
- Eberhart R and Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, pp 39–43
- Sun J, Garibaldi JM, Hodgman C (2012) Parameter estimation using metaheuristics in systems biology: a comprehensive review. *IEEE/ACM Trans Comput Biol Bioinform* 9(1):185–202
- Yoon Y, Kim YH (2013) An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks. *IEEE Trans Cybern* 43(5):1473–1483
- Wu T, Lin C (2015) Low-SAR path discovery by particle swarm optimization algorithm in wireless body area networks. *IEEE Sens J* 15(2):928–936
- Bernardino HS, Barbosa HJC, Fonseca LG (2011) Surrogate-assisted clonal selection algorithms for expensive optimization problems. *Evol Intel* 4(2):81–97
- Koziel S, Ogurtsov S (2013) Multi-objective design of antennas using variable-fidelity simulations and surrogate models. *IEEE Trans Antennas Propag* 61(12):5931–5939
- Shan X, Wang B, Xu K, Li H (2020) Surrogate model-based structure optimization of jetting system. *IEEE Trans Compon Packag Manuf Technol* 1(3):494–501
- Zhang B, Pan Q, Gao L, Meng L, Li X, Peng K (2020) A three-stage multiobjective approach based on decomposition for an energy-efficient hybrid flow shop scheduling problem. *IEEE Trans Syst Man Cybern Syst* 50(12):4984–4999
- Wang GG (2003) Adaptive response surface method using inherited latin hypercube design points. *J Mech Des* 125(2):210–220
- Hardy RL (1971) Multiquadric equations of topography and other irregular surfaces. *J Geophys Res Atmos* 76(8):1905–1915
- Dyn N, Levin D, Rippa S (1986) Numerical procedures for surface fitting of scattered data by radial functions. *SIAM J Sci Stat Comput* 7(2):639–659
- Gutmann HM (2001) A radial basis function method for global optimization. *J Glob Optim* 19(3):201–227
- Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J Glob Optim* 13(4):455–492
- Martin JD, Simpson TW (2005) Use of kriging models to approximate deterministic computer models. *AIAA J* 43(4):853–863
- Clarke SM, Griebisch JH, Simpson TW (2005) Analysis of support vector regression for approximation of complex engineering analyses. *J Mech Des* 127(6):1077–1877
- Jin Y, Olhofer M, Sendhoff B (2002) A framework for evolutionary optimization with approximate fitness functions. *IEEE Trans Evol Comput* 6(5):481–494
- Ratle A (2001) Kriging as a surrogate fitness landscape in evolutionary optimization. *Artif Intell Eng Des Anal Manuf* 15(1):37–49
- Regis RG (2014) Particle swarm with radial basis function surrogates for expensive black-box optimization. *J Comput Sci* 5(1):12–23
- Liu B, Zhang Q, Gielen GGE (2014) A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Trans Evol Comput* 18(2):180–192
- Dong H, Song B, Wang P, Huang S (2015) A kind of balance between exploitation and exploration on kriging for global optimization of expensive functions. *J Mech Sci Technol* 29:2121–2133
- Ong YS, Nair PB, Keane AJ (2003) Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA J* 41(4):687–696
- Smith RE, Dike BA and Stegmann SA (1995) Fitness inheritance in genetic algorithms. In: Proceedings of the 1995 ACM symposium on applied computing, pp 345–350
- Sun C, Zeng J, Pan J, Xue S, Jin Y (2013) A new fitness estimation strategy for particle swarm optimization. *Inf Sci* 221:355–370
- Lu X, Tang K (2012) Classification- and regression-assisted differential evolution for computationally expensive problems. *J Comput Sci Technol* 27:1024–1034
- Zhou Z, Ong YS, Nair PB, Keane AJ, Lum KY (2007) Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Trans Syst Man Cybern Part C (Appl Rev)* 37(1):66–76

28. Tenne Y, Armfield SW (2009) A framework for memetic optimization using variable global and local surrogate models. *Soft Comput* 13(8–9):781–793
29. Sun C, Jin Y, Zeng J, Yu Y (2015) A two-layer surrogate-assisted particle swarm optimization algorithm. *Soft Comput* 19(6):1461–1475
30. Wang H, Jin Y, Doherty J (2017) Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Trans Cybern* 47(9):2664–2677
31. Jin Y (2011) Surrogate-assisted evolutionary computation: recent advances and future challenges. *Swarm Evol Comput* 1(2):61–70
32. Tian J, Tan Y, Zeng J, Sun C, Jin Y (2019) Multiobjective infill criterion driven Gaussian process-assisted particle swarm optimization of high-dimensional expensive problems. *IEEE Trans Evol Comput* 23(3):459–472
33. Li F, Cai X, Gao L (2020) A surrogate-assisted multiswarm optimization algorithm for high-dimensional computationally expensive problems. *IEEE Trans Cybern* 51(3):1390–140234
34. Wang X, Wang G, Song B, Wang P, Wang Y (2019) A novel evolutionary sampling assisted optimization method for high-dimensional expensive problems. *IEEE Trans Evol Comput* 23(5):815–827
35. Sun C, Jin Y, Cheng R, Ding J, Zeng J (2017) Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Trans Evol Comput* 21(4):644–660
36. Yu H, Tan Y, Zeng J, Sun C, Jin Y (2018) Surrogate-assisted hierarchical particle swarm optimization. *Inf Sci* 454–455:59–72
37. Yang Z, Qiu H, Gao L, Jiang C, Zhang J (2019) Two-layer adaptive surrogate-assisted evolutionary algorithm for high-dimensional computationally expensive problems. *J. Glob Optim* 74(2):327–359
38. Dong H, Dong Z (2020) Surrogate-assisted grey wolf optimization for high-dimensional, computationally expensive black-box problems. *Swarm Evol Comput*. <https://doi.org/10.1016/j.swevo.2020.100713>
39. Cai X, Gao L, Li X (2020) Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems. *IEEE Trans Evol Comput* 24(2):365–379
40. Lim D, Jin Y, Ong YS, Sendhoff B (2010) Generalizing surrogate-assisted evolutionary computation. *IEEE Trans Evol Comput* 14(3):329–355
41. Vwlasco-Forero S, Angulo J (2012) Random projection depth for multivariate mathematical morphology. *IEEE J Sel Top Signal Process* 6(7):753–763
42. Majumdar A, Ward RK (2010) Robust classifiers for data reduced via random projections. *IEEE Trans Syst Man Cybern Part B (Cybern)* 40(5):1359–1371
43. Kabán A, Bootkrajang J and Durrant RJ (2013) Towards large scale continuous EDA: a random matrix theory perspective. In: *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pp 383–389
44. Frankl P, Maehara H (1988) The Johnson–Lindenstrauss lemma and the sphericity of some graphs. *J Comb Theory* 44(3):355–362
45. Anupam SD, Gupta A (1999) An elementary proof of the Johnson–Lindenstrauss lemma. *Random Struct Algorithms* 22(1):1–5
46. Jin R, Chen W, Subjianto A (2005) An efficient algorithm for constructing optimal design of computer experiments. *J Stat Plan Inference* 134(1):268–287
47. Suganthan PN, Hansen N, Liang JJ et al (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Nanyang Technological University, Singapore (**Report**)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.