Journal of Global Optimization (2019) 74:327–359
https://doi.org/10.1007/s10898-019-00759-0

Two-layer adaptive surrogate-assisted evolutionary algorithm for high-dimensional computationally expensive problems

Zan Yang[1] · Haobo Qiu[1] · Liang Gao[1] · Chen Jiang[1] · Jinhao Zhang[1]

Received: 12 July 2017 / Accepted: 1 March 2019 / Published online: 6 March 2019

Abstract
Surrogate-assisted evolutionary algorithms (SAEAs) have recently shown excellent ability in solving computationally expensive optimization problems. However, with the increase of dimensions of research problems, the effectiveness of SAEAs for high-dimensional problems still needs to be improved further. In this paper, a two-layer adaptive surrogate-assisted evolutionary algorithm is proposed, in which three different search strategies are adaptively executed during the iteration according to the feedback information which is proposed to measure the status of the algorithm approaching the optimal value. In the proposed method, the global GP model is used to pre-screen the offspring produced by the DE/current-to-best/1 strategy for fast convergence speed, and the DE/current-to-randbest/1 strategy is proposed to guide the global GP model to locate promising regions when the feedback information reaches a presetting threshold. Moreover, a local search strategy (DE/best/1) is used to guide the local GP model which is built by using individuals closest to the current best individual to intensively exploit the promising regions. Furthermore, a dimension reduction technique is used to construct a reasonably accurate GP model for high-dimensional expensive problems. Empirical studies on benchmark problems with 50 and 100 variables demonstrate that the proposed algorithm is able to find high-quality solutions for high-dimensional problems under a limited computational budget.

Keywords Surrogate-assisted evolutionary algorithms · Computationally expensive problems · Differential evolution · Dimension reduction technique

1 Introduction

Metaheuristic optimization algorithms, such as differential evolution (DE), genetic algorithm (GA), ant colony optimization (ACO) and particle swarm optimization (PSO), have

✉ Haobo Qiu
hobbyqiu@163.com

[1] The State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan 430074, People's Republic of China

been empirically demonstrated that they can achieve great success on many real-world engineering problems, such as electric power systems [1, 2], job shop scheduling [3], wireless networks [4, 5] and mechanical design optimization [6]. However, many engineering design optimization problems involve the use of high fidelity simulation methods such as Finite Element Analysis (FEA), Computational Fluid Dynamics (CFD) and Computational Electro Magnetics (CEM) for performance evaluations, which are often computationally expensive, ranging from several minutes to days of supercomputer time [7]. Since most metaheuristic algorithms typically require a large number of fitness evaluations (FES) to obtain an acceptable optimum solution, the application of metaheuristic algorithms to these expensive problems becomes relatively intractable. A promising method to reduce computation time for the optimization of expensive optimization problems is to employ surrogates (also known as meta-models and approximation models), whose computational efforts required for surrogate-assisted evolutionary algorithms (SAEAs) are usually much lower, to replace in part the highly time-consuming exact function evaluations for saving computational cost [8]. Over recent years, the most commonly used surrogate models include artificial neural networks (ANNs) [9, 10], polynomial regression (PR, also known as response surface method) [11], support vector machines (SVMs) [12, 13], radial basis function (RBF) [14, 15], and Gaussian Processes (GP, also referred as to Kriging) [16–19].

In the earlier stage of the research on SAEAs, global-surrogate models, which aim to model in the whole search space, are often used in evolutionary algorithms (EAs) to approximate the highly time-consuming objective functions. Ratle [20] used the Kriging model to replace the exact function evaluation during the evolutionary search. Jin et al. [21] examined strategies for integrating evolutionary algorithm with an artificial neural network and proposed an empirical criterion to switch between the computationally expensive real fitness function and cheap fitness function during the evolutionary search. Different strategies were proposed in Ulmer et al. [22] based on Gaussian Process (GP) models. Karakasis [23] utilized the radial basis function (RBF) network for expensive multi-objective problems, and the RBF assisted evolutionary search could pre-screen the most promising individuals. Parno et al. [24] used the Kriging as a substitute for the time-consuming objective function within a particle swarm optimization (PSO) framework. A comprehensive survey about the surrogate-assisted evolutionary computation has been given by Jin et al. [25]. Nuovo et al. [26] analyzed an empirical study on evaluating candidate individuals by fuzzy function approximation to speed up evolutionary multi-objective optimization. Liu et al. [27] proposed a Gaussian Process (GP) surrogate model assisted evolutionary algorithm, in which dimension reduction and new surrogate model-aware search mechanisms were utilized to solve medium-scale computationally expensive optimization problems. A cheap surrogate model based on density estimation was proposed by Gong et al. [28] for pre-screening the candidate individuals during the evolutionary search.

However, with the increase of dimensions of research problems, it is often difficult to build reliable and accurate global surrogate models as substitutes for the real objective functions due to the “curse of dimensionality” [25]. Therefore, local surrogate models, which are built by utilizing specific training data selection strategy, are intensively explored in order to enhance the accuracy of the surrogate model. For instance, Ong et al. [29] employed a trust-region method to interleave real objective models for the objective and constrained functions with computationally cheap RBF surrogate models used in the local search process. Smith et al. [30] proposed the concept of fitness inheritance in genetic algorithm (GA), in which the fitness values of individuals are inherited from other individuals or their parents. Furthermore, a fitness inheritance strategy was adopted by Hendtlass [31] in PSO and a reliability measure was employed to enhance the accuracy of fitness estimation. A fitness estimation strategy was proposed by Sun et al. [32] based on the analysis on the positions of the particles to reduce the number of real fitness function evaluations.

Considering the issues that EAs based on local surrogate models can't solve multi-modal problems effectively and EAs based on global surrogate models can't build a relatively accurate surrogate model to cope with high-dimensional problems, some researchers used both global and local surrogate models in EA. A hierarchical surrogate-assisted evolutionary algorithm was introduced by Zhou et al. [33], in which the ensemble models of Gaussian Process and Polynomial regression models are used as global surrogate models. Then the global surrogate models are employed to pre-screen the current population during the evolutionary search for promising individuals, and a local search is undergone in the form of Lamarckian learning by using local surrogate models. Tenne and Armfield [34] introduced a memetic algorithm combining global and local surrogate models and employed the trust-region approach for the optimization of time-consuming objective functions. Lim et al. [7] used the ensemble model of diverse surrogate models to enhance the accuracy of estimation values, where the global surrogate model was used to speed up evolutionary search by traversing through the multi-modal landscape. Müller et al. [35] examined the influence of two major aspects on the individual quality of surrogate model algorithms for computationally expensive black-box global optimization problems, namely the surrogate model selection and the method of iteratively adding sample points. Sun et al. [36] proposed a two-layer surrogate-assisted PSO algorithm, in which the global surrogate model is built to smooth out the local optima and guide the swarm to fly quickly to an optimum and a number of local surrogate models are employed for fitness estimation to obtain the optimum.

There are also some surrogated-based optimization methods that can handle high-dimensional expensive problems. A very recent method named as KPLS [37] applied Kriging to high-dimensional problems through a dimension reduction technique, and a covariance kernel is constructed based on the information obtained using the partial least squares method. Some other methods are proposed to deal with high-dimensional expensive problems with constraints. ConstrLMSRBF [38] built RBF surrogate models for the objective and constraint functions, which were used to guide the selection of next point. Regis [39] utilized the surrogate functions to identify the trial offspring that are predicted to be feasible with the best predicted fitness or those offspring with the minimum number of predicted constrained violations.

In addition, some surrogate-assisted DEs have been proposed to solve computationally expensive optimization problems. Liu et al. [40] proposed a new surrogate-assisted DE optimization framework for handling discrepancies between simulation models with multiple fidelities. A local ensemble surrogate-assisted crowding DE (LES-CDE) was put forward by Jin et al. [41]. Awad et al. [42] introduced an efficient adapted surrogate model assisted L-SHADE algorithm. Sa-DE-DPS algorithm [43] employed a mechanism to dynamically select the best combinations of parameters. ESMDE [44] used an ensemble of different mutation strategies to guide the evolutionary search. However, none of these methods are proposed to deal with high-dimensional problems. Furthermore, they also do not consider the adaptive adjustment of multiple mutation strategies and different surrogate construction strategies using the update status information of the best individual during the iterative process.

From the above literatures, we can see that various approaches are proposed for SAEAs both on the aspects of efficient iteration architecture and high performance surrogate modeling, with the aim of improving the overall optimization ability. Moreover, most of these SAEAs utilize the information produced in the iterative process such as the previous sample points only to construct surrogates without further using the iterative status information of the optimization algorithm. While this information is very important in effectively guiding the algorithm to adapt to different search strategies and surrogate building strategies, thus achieving an adaptive searching. Furthermore, these SAEAs can't accurately measure the degree of how much the algorithm gets stuck into the local optimum during the iterative process, and the corresponding strategies for adjusting the algorithm to jump out local optimum are also lacked. This may be one of the reasons that lead some SAEAs fall into premature convergence more easily when confronted with complicated multi-modal problems. In this paper, we propose a concept of state value that can measure the status of the algorithm approaching the optimal value during the iteration. And according to the information brought from this value, the algorithm is guided to be more partial to the global or local search. Therefore, a global strategy is proposed with global GP model in order to achieve an efficient global search, and a local strategy is also applied with adaptive local GP model for a good local search.

In our methods, GP modeling with specific pre-screening strategy is used. It is mainly based on the consideration that GP modeling can provide an uncertainty measure in the form of standard deviation to each predicted point. Therefore, several pre-screening methods, such as the lower confidence bound (LCB) [45], the probability of improvement (PI) [46], the expected improvement (EI) [2, 47], have been put forward for GP modeling in optimization. We can use the appropriate pre-screening strategy to further improve the performance of our method.

Furthermore, it is always unaffordable to build a sufficiently accurate surrogate model in the original high-dimensional space for high-dimensional problems. A dimension reduction technique is also used in this paper. It employs a specific dimension reduction technique to map the training data, which in the original space, to a relatively lower dimensional space in which the Gaussian Process model will be built. Then, the quality of the global and adaptive local GP models, which are built in the reduced space, can be improved and the computational cost of modeling can be reduced significantly.

The remainder of this paper is organized as follows. Section 2 briefly introduces the related background techniques including the GP modeling and dimension reduction method. In Sect. 3, the TASEA method is proposed in detail. Section 4 presents the experimental results of TASEA on some widely used benchmark problems of a dimension 50 and 100. Comparisons with some state-of-the-art methods are also presented in this section. Section 5 concludes the paper with a summary and future work.

2 Background

2.1 Gaussian process modeling

Without loss of generality, we consider the following optimization problem:

$$\begin{aligned} \text{minimize}: &\ f(\mathbf{x}) \\ \text{subject to}: &\ \mathbf{x}_l \le \mathbf{x} \le \mathbf{x}_u \end{aligned} \tag{1}$$

where $f(\mathbf{x})$ is a scalar-valued objective function, $\mathbf{x} = (x_1, x_2, \ldots, x_D) \in \Re^D$ is a vector of continuous decision variables, $\mathbf{x}_l$ and $\mathbf{x}_u$ are vectors of the lower and upper bounds of the search space respectively.

To model an unknown function $y = f(\mathbf{x})$, $\mathbf{x} \in \Re^D$, GP modeling assumes that $f(\mathbf{x})$ at any point $\mathbf{x}$ is a Gaussian random variable $N(\mu, \sigma^2)$, where μ and σ^2 represent respectively the constant mean and constant variance. For any $\mathbf{x}$, $f(\mathbf{x})$ is a sample of $\mu + \varepsilon(\mathbf{x})$, where

$\varepsilon(\mathbf{x}) \sim N(\mu, \sigma^2)$. For any $\mathbf{x}$, $\mathbf{x}' \in \Re^D$, $\varphi(\mathbf{x}, \mathbf{x}')$, the correlation between $\varepsilon(\mathbf{x})$ and $\varepsilon(\mathbf{x}')$, depends on $\mathbf{x} - \mathbf{x}'$. More precisely

$$\varphi(\mathbf{x}, \mathbf{x}') = exp\left(-\sum_{i=1}^{D} \theta_i \left|x_i - x_i'\right|^{p_i}\right) \tag{2}$$

where parameter θ_i indicates the importance of x_i on function $f(\mathbf{x})$, and parameter $p_i (1 \leq p_i \leq 2)$ indicates the smoothness of $f(\mathbf{x})$ with respect to x_i. More details about Gaussian Process modeling can be found in literature [48].

(1) *Hyper parameter Estimation* Given n points $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n \in \Re^D$ and their response values, then the hyper parameters $\mu, \sigma, \theta_1, \theta_2, \ldots, \theta_d$, and $p_1, p_2, \ldots, p_d$ can be estimated by maximizing the likelihood function $f(\mathbf{x}) = y^i$ at $\mathbf{x} = \mathbf{x}^i$ $(i = 1, \ldots, n)$ [46]

$$\frac{1}{(2\pi\sigma^2)^{n/2}\sqrt{|\mathbf{\Phi}|}} \exp\left[-\frac{(\mathbf{y} - 1\mu)^T \mathbf{\Phi}^{-1} (\mathrm{y} - 1\mu)}{2\sigma^2}\right] \tag{3}$$

where $\mathbf{\Phi}$ is a $n \times n$ matrix whose $(i, j) - element$ is $\varphi(\mathbf{x}^i, \mathbf{x}^j)$, $\mathbf{y} = (y^1, \ldots, y^n)^T$, and 1 is a n-dimensional column vector of ones.

To maximize Eq. (3), the values of μ and σ can be obtained as follow.

$$\hat{\mu} = \frac{1^T \mathbf{\Phi}^{-1} \mathrm{y}}{1^T \mathbf{\Phi}^{-1} 1} \tag{4}$$

and

$$\hat{\sigma}^2 = \frac{(\mathrm{y} - 1\mu)^T \mathbf{\Phi}^{-1} (\mathrm{y} - 1\mu)}{n} \tag{5}$$

Substituting Eqs. (4) and (5) into Eq. (3) can eliminate the unknown parameters μ and σ from Eq. (3). Therefore, the value of the likelihood function depends only on θ_i and p_i. Equation (3) can be maximized to obtain estimates values of $\hat{\theta}_i$ and $\hat{p}_i$, and the estimate values of $\hat{\theta}_i$ and $\hat{p}_i$ can be readily acquired from Eqs. (4) and (5). In our experiments, the DACE toolbox [49] with MATLAB environment is used to optimize the likelihood function.

(2) *Best Linear Unbiased Prediction and Predictive Distribution* Given the estimated hyper parameter $\hat{\theta}_i$, $\hat{p}_i$, $\hat{\mu}$ and $\hat{\sigma}^2$, one can predict $y = f(\mathbf{x})$ at any untested point $\mathbf{x}$ based on response values $f(\mathbf{x}) = y^i$ at $\mathbf{x} = \mathbf{x}^i$ $(i = 1, \ldots, n)$. The best linear unbiased predictor of $f(\mathbf{x})$ is [50]

$$\hat{f}(\mathbf{x}) = \hat{\mu} + \boldsymbol{\phi}^T \mathbf{\Phi}^{-1} (\mathrm{y} - 1\hat{\mu}) \tag{6}$$

And its mean square error is

$$s^2(\mathbf{x}) = \hat{\sigma}^2 \left[1 - \boldsymbol{\phi}^T \mathbf{\Phi}^{-1} \boldsymbol{\phi} + \frac{(1 - 1^T \mathbf{\Phi}^{-1} \boldsymbol{\phi})^2}{1^T \mathbf{\Phi}^{-1} \boldsymbol{\phi}}\right] \tag{7}$$

where $\boldsymbol{\phi} = (\varphi(\mathbf{x}, \mathbf{x}^1), \ldots, \varphi(\mathbf{x}, \mathbf{x}^n))^T$. $N(\hat{f}(\mathbf{x}), s^2(\mathbf{x}))$ can be regarded as a predictive distribution for $f(\mathbf{x})$ given the function values y^i at $\mathbf{x}^i$ for $i = 1, 2, \ldots, n$.

(3) *Pre-screening strategy* Three strategies are always used to deal with the uncertainties of surrogate models. Given the predictive distribution $N(\hat{f}(\mathbf{x}), s^2(\mathbf{x}))$ for $f(\mathbf{x})$.

(1) The Lower confidence bound (LCB) of $f(\mathbf{x})$ can be defined as [45]

$$f_{lcb}(\mathbf{x}) = \hat{f}(\mathbf{x}) - ws(\mathbf{x}) \tag{8}$$

where w is a constant that controls the balance of exploitation and exploration. As $w \to 0$, $f_{lcb}(\mathbf{x}) \to \hat{f}(\mathbf{x})$ (purely search the promising regions with lower $\hat{f}(\mathbf{x})$ values) and as $w \to \infty$, the effect of $\hat{f}(\mathbf{x})$ becomes negligible and minimizing $f_{lcb}(\mathbf{x})$ is equivalent to maximizing $\hat{s}(\mathbf{x})$ (purely search less-explored areas with high $\hat{s}(\mathbf{x})$ values).

(2) The PI of $f(\mathbf{x})$ can be defined as [46]

$$P[I(\mathbf{x})] = \frac{1}{\hat{s}\sqrt{2\pi}} \int_{-\infty}^{0} e^{-\frac{[I-\hat{y}(\mathbf{x})]^2}{2\hat{s}^2}} dI \tag{9}$$

where $I(\mathbf{x}) = y_{\text{min}} - Y(\mathbf{x})$, y_{min} is the minimum value of the current observed values so far. For each individual, the PI value represents the probability of improvement on the best observed value so far. An individual can be chosen to make the database updated when the biggest PI value is achieved in the current population.

(3) the EI of $f(\mathbf{x})$ can be defined as [2]

$$E[I(\mathbf{x})] = \begin{cases} (y_{\text{min}} - \hat{y}(\mathbf{x}))\boldsymbol{\Phi}\left(\frac{y_{\text{min}} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) + s\boldsymbol{\phi}\left(\frac{y_{\text{min}} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) & if \quad s > 0 \\ 0 & if \quad s = 0 \end{cases} \tag{10}$$

where $\boldsymbol{\Phi}(\bullet)$ and $\boldsymbol{\phi}(\bullet)$ are the cumulative distribution function and probability density function respectively. An individual can be chosen to make the database updated when the biggest EI value is achieved in the current population.

2.2 Dimension reduction technique

We use the dimension reduction technique in SAEA to reduce the computational cost and improve model accuracy. The key idea of the dimension reduction technique is to map the high-dimensional training data to a lower dimensional space and then conduct GP modeling in the lower dimensional space. Therefore, the defect caused by the insufficient number of training data points for high-dimensional problems can be overcome effectively. The method is shown in Algorithm 1 [27].

Algorithm 1 GP modeling with dimension reduction technique

```
1. Get the input data, which includes: (1) Training data: $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^K$ and $y^1, y^2, \ldots, y^K$, (2) the
     offspring of the current population: $\mathbf{x}^{K+1}, \mathbf{x}^{K+2}, \ldots, \mathbf{x}^{K+M}$.
2. Map $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{K+M} \in R^D$ into $R^L$, where $L < D$, and obtain their corresponding images in $R^L$:
     $\bar{\mathbf{x}}^1, \bar{\mathbf{x}}^2, \ldots, \bar{\mathbf{x}}^{K+M}$.
3. Build a GP model by using the $\bar{\mathbf{x}}^i$ and $y^i$ data ($i = 1, 2, \ldots, K$).
4. Use the GP model to rank each $\bar{\mathbf{x}}^{k+j}$ ($j = 1, 2, \ldots, M$) by the chosen pre-screening method.
5. Find the best individual by ranking. Then output its corresponding point in the original decision space
     as the estimated best individual.
```

There are many machine learning methods that can transform the original space to the latent space for dimension reduction, and these techniques mainly include Principle Component Analysis (PCA), Local Linear Embedding (LLE), Neighborhood Components Analysis (NCA) [51] and Sammon mapping [52]. To select an appropriate machine learning method, we have the following three considerations.

1. The correlation function is a very important part of the GP modeling, and the value of the correlation function is proportional to the distance between any two individuals. Therefore, the neighborhood relationships and pairwise distances among the individuals should be preserved as much as possible.

2. The dimension reduction technique will be used many times in our proposed methods. Therefore, the extra computational cost produced by the mapping technique should be very cheap.
3. The Sammon mapping is an algorithm that tries to preserve the structure of inter-point distances in high-dimensional space during the lower-dimensional projection.

Based on these considerations, the Sammon mapping is used in our paper. Sammon mapping aims to minimize the following error function.

$$E = \frac{1}{\sum\limits_{i<j} d_{ij}^*} \sum_{i<j} \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*}$$

where d_{ij}^* is the distance between i-th and j-th objects in the original space, and d_{ij} is the distance between their projections. In our experiments, the SOM toolbox [53] with MATLAB environment is used to minimize the error function.

3 The proposed TASEA algorithm

3.1 Differential evolution (DE) in TASEA

Differential evolution (DE), proposed by Storn and Price [54], has exhibited remarkable performances for many optimization problems in diverse fields. The performance of DE mainly depends on the search strategies and control parameter settings (i.e., population size NP, scaling factor F and cross control parameter C_r). In the proposed TASEA, we use three effective search strategies with corresponding control parameter settings to search adaptively during the evolutionary optimization process. Compared to DE/rand/k, greedy strategies such as DE/current-to-best/k and DE/best/k benefit from their fast convergence by incorporating best individual information in the evolutionary search [55]. However, incorporating too much information about the best individual, these greedy algorithms may be easier to fall into premature convergence because of the reduced population diversity. Based on the above considerations, the DE/current-to-best/1 strategy is used as a main search strategy for achieving fast convergence speed. Then, a DE/current-to-randbest/1 strategy is proposed for increasing the diversity of population to avoid falling into premature convergence, and this strategy is conducted adaptively according to the feedback information from the evolutionary search process. As an intensive local search strategy, the DE/best/1 strategy is implemented in combination with the adaptive local GP model for improving the convergence speed in promising regions. The search strategies mentioned above are shown as follows.

1. DE/current-to-best/1

$$\vec{\mathbf{v}}_{i,G} = \vec{\mathbf{x}}_{i,G} + F \cdot (\vec{\mathbf{x}}_{best,G} - \vec{\mathbf{x}}_{i,G} + \vec{\mathbf{x}}_{r1,G} - \vec{\mathbf{x}}_{r2,G}) \tag{11}$$

2. DE/current-to-randbest/1

$$\vec{\mathbf{v}}_{i,G} = \vec{\mathbf{x}}_{i,G} + rand \cdot (\vec{\mathbf{x}}_{best,G} - \vec{\mathbf{x}}_{i,G}) + F \cdot (\vec{\mathbf{x}}_{r1,G} - \vec{\mathbf{x}}_{r2,G}) \tag{12}$$

3. DE/best/1

$$\vec{\mathbf{v}}_{i,G} = \vec{\mathbf{x}}_{best,G} + F \cdot (\vec{\mathbf{x}}_{r1,G} - \vec{\mathbf{x}}_{r2,G}) \tag{13}$$

In the above equations, $r1$ and $r2$ are distinct integers randomly selected from the range $[1, NP]$ and different from i, $rand$ is a uniformly distributed random number between 0 and 1 which is generated for each i, and $F \in (0, 2]$ is a scaling factor. After mutation, a binomial crossover operator on $\vec{\mathbf{x}}_{i,G}$ and $\vec{\mathbf{v}}_{i,G}$ is performed to generate a trial vector $\vec{\mathbf{u}}_{i,G} = (u_{i,1,G}, u_{i,2,G}, \ldots, u_{i,D,G})$

$$u_{i,j,G} = \begin{cases} v_{i,j,G}, & if \;\; rand_j(0,1) \leq C_r \;\; or \;\; j = j_{rand} \\ x_{i,j,G}, & otherwise \end{cases} \tag{14}$$

where C_r is a constant named the crossover rate. The performance of DE is very sensitive to the search strategies and their associated parameter settings. Generally, a larger F results in a higher diversity of mutation vectors in order to achieve more global search. However, a smaller F is inclined to local exploitation because the mutation vector changes a little relative to the target vector. A larger C_r can lead the trial vector to inherit more information from the mutation vector, which results in a more global exploration. While a smaller C_r can ensure the trial vector inherit more information from the target vector, which results in a more local exploitation. Hence, as a main strategy, the DE/current-to-best/1 aims to ensure the convergence speed and in part the diversity of the child individuals, and we set F to 0.8 and C_r to 0.8 following [56]. In the DE/current-to-randbest/1 strategy, the value of $rand$ and F represent the weight of the information obtained from the current best individual and the information offered by the random individuals respectively. The parameter $rand$ is a uniformly distributed variable of mean 0.5. So we set F to 0.8 for attaching great importance to the diversity of population and less importance to the information from the current best individuals. We set C_r to 0.6 for enhancing the diversity of the trial vector. In the DE/best/1 strategy, we set F to 0.9 and C_r to 0.2 for achieving fast convergence speed in promising regions. Detailed sensitivity analyses about these parameters can be seen in the later section.

3.2 Feedback mechanism

In recent years, adaptive operator selection strategies and adaptive parameter control strategies have been extensively applied in evolutionary algorithms. In the evolutionary algorithms assisted by adaptive operator selection strategies, the key issue is how to assign the mutation or reproduction operator adaptively. Many strategies have been proposed to achieve this objective, such as adaptive operator probability matching [57], adaptive pursuit strategy [58], and collaborative conduction of different strategies [59]. In the evolutionary algorithms assisted by adaptive parameter control strategies, the key issue is how to dynamically change the control parameters according to the feedback from the evolutionary search process, such as fuzzy adaptive DE [60], SaDE [61], jDE [62], JADE [55]. However, these adaptive strategies are only used in evolutionary algorithms for solving computationally cheap problems, in which expensive function evaluations are not involved. As for high-dimensional computationally expensive problems, new methods should be put forward to use surrogate models in these adaptive evolutionary algorithms. So based on the feedback information during the evolutionary search process, TASEA is proposed that can adaptively guide the optimization directions with different surrogate modeling and search strategies for improving optimization efficiency under limited computational budget. The feedback mechanism of the proposed method is shown as follows in Algorithm 2. The concept of the state value N_f is introduced to represent this feedback information during the evolutionary search process. It is defined as the number of keeping invariant of the best individual in the database during the successive iterations, and the state value is reset when the best individual is updated. In other words, the parameter N_f can keep track of the number of iterations where the best individual is not updated. There are also some surrogate-based algorithms that use similar parameters to keep track of how long an algorithm has gotten stuck. Regis and shoemaker [63] proposed a complete restart strategy for CG-RBF and CORS-RBF [64] when the algorithm fails to make any substantial progress after some threshold number of consecutive iterations. Holmström introduced an adaptive radial basis algorithm ARBF [65], in which two Indicator variables are used to switch between the global and local grid mode. In [38], two parameters were introduced to keep track of the number of consecutive iterations that yielded feasible or infeasible points. In ConstrLMSRBF [66] and DYCORs-LMSRBF [67], the optimization processes are monitored by recording the number of consecutive failed iterations. In summary, although the concept of these parameters introduced in this paper and the above literatures is similar, the usage of this parameter in the proposed TASEA is different. The parameter used in TASEA can adaptively select the mutation strategies and surrogate models according to the optimization status of the algorithm.

Algorithm 2 Feedback Mechanism

```
Inputs: $N_c, N_f, \lambda, F, CR, \kappa, \mu, \gamma, good$.
Outputs: $\lambda$ child individuals and the modeling information of GP (Local GP or Global GP).
Apply DE/current-to-best/1/bin to generate the child individuals in the first generation
if $N_f < N_c$ and $good = 0$ do
    Apply DE/current-to-best/1 to generate the child individuals.
    Take the $\mu$ newest individuals in the database and their function values as the training data to construct the global GP model in the reduced space.
else $N_f \geq N_c$ and $good = 0$ do
    Apply DE/current-to-randbest/1/bin to generate the child individuals.
    Take the $\mu$ newest solutions in the database and their function values as the training data to construct the global GP model in the reduced space.
else $good = 1$ do
    Apply DE/best/1/bin to generate the child individuals
    Take the $\mu + \gamma$ nearest individuals which are closest to the best individual in the current population to construct adaptive local GP model in the reduced space.
end if
```

We can make the following remarks on the feedback mechanism.

1. The parameter *good* is an indicator variable, which indicates whether the algorithm should be adjusted to achieve specific search goals. The initial value of the parameter *good* is 0. This parameter will be set to 1 when the best individual is updated in the database, which indicates that the algorithm should be adjusted to achieve stronger local search in a promising region around the updated best individual. And it will be reset to 0 while the best individual is not updated in the database. The value of parameter γ is increased gradually with the iteration in order to improve the accuracy of the local GP model. The parameter N_c is a constant that determines when to adjust the search strategy during the iteration. If the parameter N_c is set relatively smaller, the DE/current-to-randbest/1 will be used more frequently, which may result in lower convergence speed. However, if this parameter is set relatively bigger, the DE/current-to-best/1 will be used more frequently, which may cause the algorithm being trapped into local optimum. In TASEA, N_c is set to 20, and the sensitivity of the parameter N_c will be discussed in the later section.

2. The DE/current-to-best/1 strategy can achieve faster convergent speed in solving unimodal problems, but it is sometimes not suitable for multimodal problems. The DE/current-to-randbest/1 strategy pays more attention to the information obtained by the random individuals than those offered by the best individuals, thus can achieve stronger global search. Therefore, these two strategies are combined effectively by the state value to achieve faster convergence speed in solving both unimodal and multimodal problems.
3. The μ newest individuals in the database are the most recently generated best candidate individuals, and these individuals may in some degree represent the direction of searching the promising regions, particularly after several iterations. Thus it should be accurate to locate the promising region by the global GP model, which is built by using these individuals.
4. During the iteration process, the position of the best individual can in some degree indicates the search direction of approaching the global or local optimal. Thus, the algorithm will enter into a more promising region when the best individual is updated, especially when optimization goes on for some iterations. And the majority of the individuals which are closest to the best individual are likely to be located in this promising region. Therefore, the local GP model is constructed by using these individuals which are closest to the best individual as the training data, which can ensure that the local GP model has a higher accuracy in the promising region. Furthermore, the majority of the offspring produced by the DE/best/1 strategy with corresponding parameter setting ($F = 0.2$, $C_r = 0.9$) are also more likely to be located in the promising region. Hence, the intensive local search can be achieved by using high accurate local GP model to pre-screen the offspring produced by the DE/best/1 strategy in the promising region.

3.3 The description of the proposed TASEA

The outline of the proposed TASEA algorithm is described in Algorithm 3. In this algorithm, the initial population is generated using the Latin hypercube sampling (LHS) [49]. The Latin hypercube sampling (LHS) can sample uniformly in the design space for achieving more effective sampling. All of the initial individuals will be evaluated using the real function and be archived to the database. Then, the adaptive search is conducted by the feedback mechanism which is guided by both the indicator variable *good* and state value N_f. After that, both the indicator variable and state value are updated correspondingly as the best individual is updated, and the individual evaluated by the true function and its true function value will be archived in the database. The overall flow diagram of the TASEA is given in Fig. 1.

```
Algorithm 3 The Proposed TASEA
Inputs: $f(\mathbf{x}), \mathbf{x}_l, \mathbf{x}_u, D, \beta, \alpha, \lambda, L, F, CR, N_c, \kappa, \mu, MaxFES$ . // $MaxFES$ is the maximum number of fitness evaluations.
Outputs: the solution obtained by the proposed TASEA.
Initialize the population with $\beta$ individuals $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\beta$ by using Latin hypercube sampling.
Evaluate the exact function value for each individual and archive them to the database.
Initialize the feedback values: $N_f = 1, good = 0$.
while termination criterion is not satisfied do
  Select the $\alpha$ best individuals from the database to form a population $Pop$.
  Call Algorithm 2 to generate $\lambda$ child individuals and pre-screen these offspring to obtain the estimated best child individual.
  Evaluate the exact function value of the estimated best child individual which is obtained by the previous step.
  if the new exact evaluated function value is better than the current best function value do
      $N_f = 1$
      $good = 1$
      Update the current best individual and its function value with the new exact evaluated function values.
  Else
      $N_f = N_f + 1$
      $good = 0$
      Do not update the current best individual and its function value.
  end if
  Add this new evaluated individual and its exact function value to the database.
end while
```

Remarks on the proposed TASEA can be seen as follows.

1. The target vector is selected from the population Pop, which consists of the α best individuals in the database. At the beginning of the iteration process, these best individuals are in some promising regions. Hence, the trial vectors produced by mutation and crossover strategies are in corresponding promising regions, which are convenient for the global GP models to locate the more promising regions. With the iteration process preceding, most of these individuals may not be far away from each other, the more intensively global search by using GP models is achieved in a relatively small promising region.
2. In Algorithm 3, the estimated best child individual may not be necessarily the highest ranked individual when considering the uncertainty of the GP model. Therefore, we randomly choose an individual from the top κ individuals, and the setting of this parameter will be analyzed in the following section.

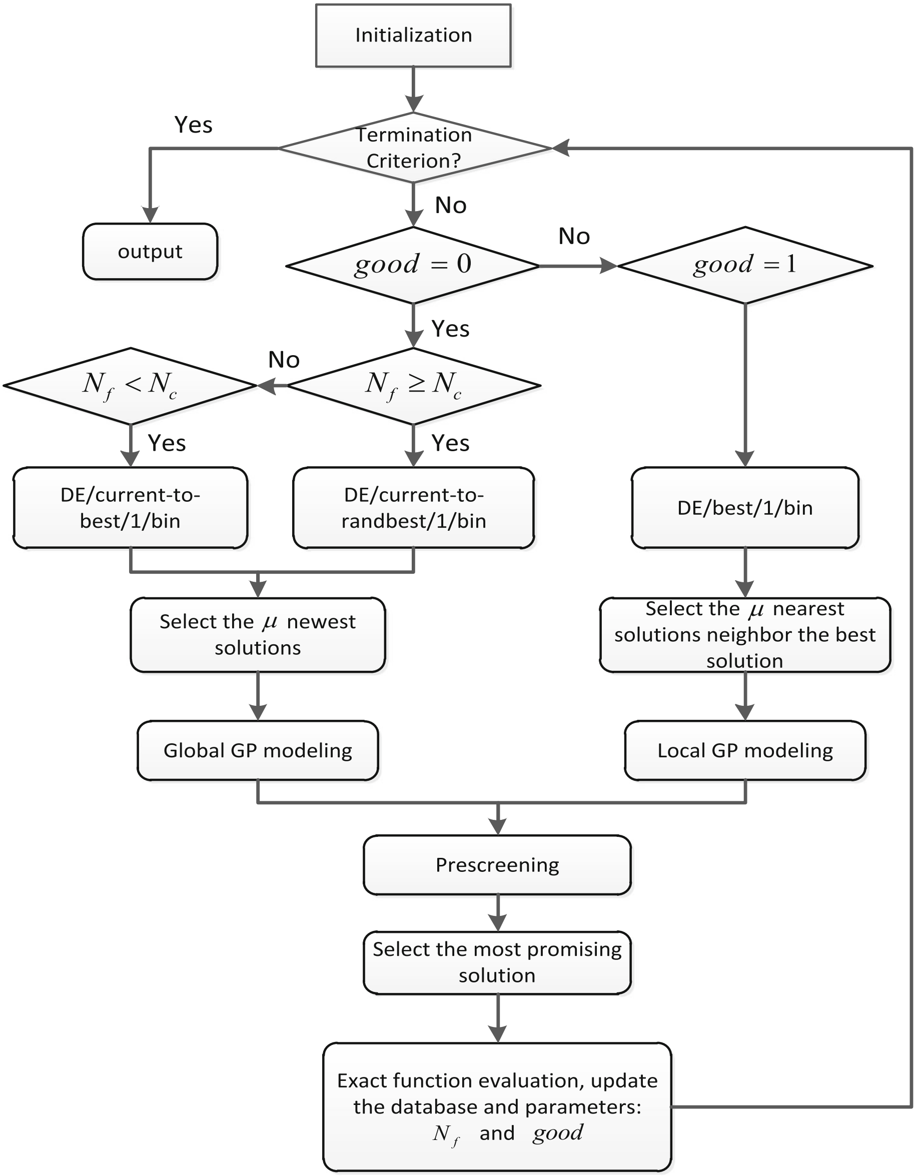

Initialization
Termination Criterion?
Yes
output
No
$good = 0$
No
$good = 1$
Yes
No
$N_f < N_c$
$N_f \geq N_c$
Yes
Yes
DE/current-to-best/1/bin
DE/current-to-randbest/1/bin
DE/best/1/bin
Select the μ newest solutions
Select the μ nearest solutions neighbor the best solution
Global GP modeling
Local GP modeling
Prescreening
Select the most promising solution
Exact function evaluation, update the database and parameters: N_f and $good$

Fig. 1 Flow diagram of TASEA

4 Experimental studies

4.1 Test problems

To investigate the effectiveness of the proposed TASEA algorithm for solving high-dimensional problems, we conducted an empirical study on twenty widely used uni-modal and multi-modal benchmark problems. Twelve of them are 50-dimensional benchmark prob-

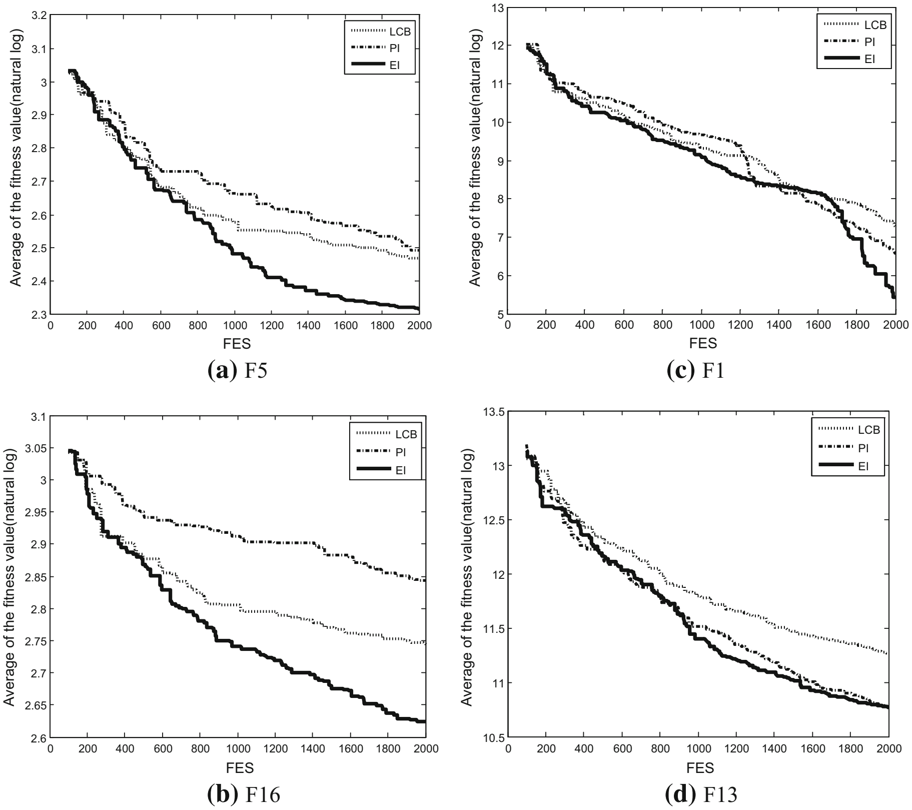
Table 1 Problems used in the experimental studies

Problem	Description	Dimensions	Global optimum	Characteristics
F1 [28]	Sphere	50	– 1400	Unimodal
F2 [32]	Different Powers	50	– 1000	Unimodal
F3 [27]	Ellipsoid	50	0	Unimodal
F4 [27, 28, 32]	Rosenbrock	50	0	Multimodal with narrow valley
F5 [27, 28]	Ackley	50	0	Multimodal
F6 [27, 28]	Griewank	50	0	Multimodal
F7 [68]	Rotated Griewank	50	– 500	Multimodal
F8 [68]	Expanded Griewank plus Rosenbrock	50	500	Multimodal
F9 [27, 28]	Shifted Rotated Rastrigin	50	– 330	Very complicated multimodal
F10 [27, 28]	Rotated Hybrid Composition Function	50	120	Very complicated multimodal
F11 [28]	Rotated Hybrid Composition Function with narrow basin global optimum	50	10	Very complicated multimodal
F12 [68]	Composition Function 8 (n = 5,Rotated)	50	1400	Very complicated multimodal
F13 [28]	Shifted Sphere	100	– 450	Unimodal
F14 [27]	Ellipsoid	100	0	Unimodal
F15 [27, 28]	Rosenbrock	100	0	Multimodal with narrow valley
F16 [27, 28]	Ackley	100	0	Multimodal
F17 [27, 28]	Griewank	100	0	Multimodal
F18 [28]	Shifted Rosenbrock	100	390	Multimodal
F19 [69]	Hybrid Function 3 (N = 3)	100	1300	Very complicated multimodal
F20 [69]	Composition Function 5 (N = 5)	100	2500	Very complicated multimodal

lems and eight of them are 100-dimensional benchmark problems. The characteristics of these test benchmark problems are listed in Table 1. Three state-of-art methods reported in GPEME [27], CSM-JADE [28], FESPSO [32] are compared with our proposed TASEA. In order to make fair comparisons, the parameter settings about these three compared methods follow their original literatures. The maximum number of fitness evaluations is set to 2000 for F1–F20 reported in Table 1. All experimental results are obtained over 20 independent runs in Matlab R2014a.

4.2 Pre-screening strategy

There are three pre-screening strategies that have been intensively investigated in SAEAs, i.e. LCB, PI and EI. Taking the four typical test functions (e.g., F1, F5, F13, F16, which are listed in Table 1) which involve uni-modal and complicated multi-modal problems as examples, we



(a) F5

(c) F1

(b) F16

(d) F13

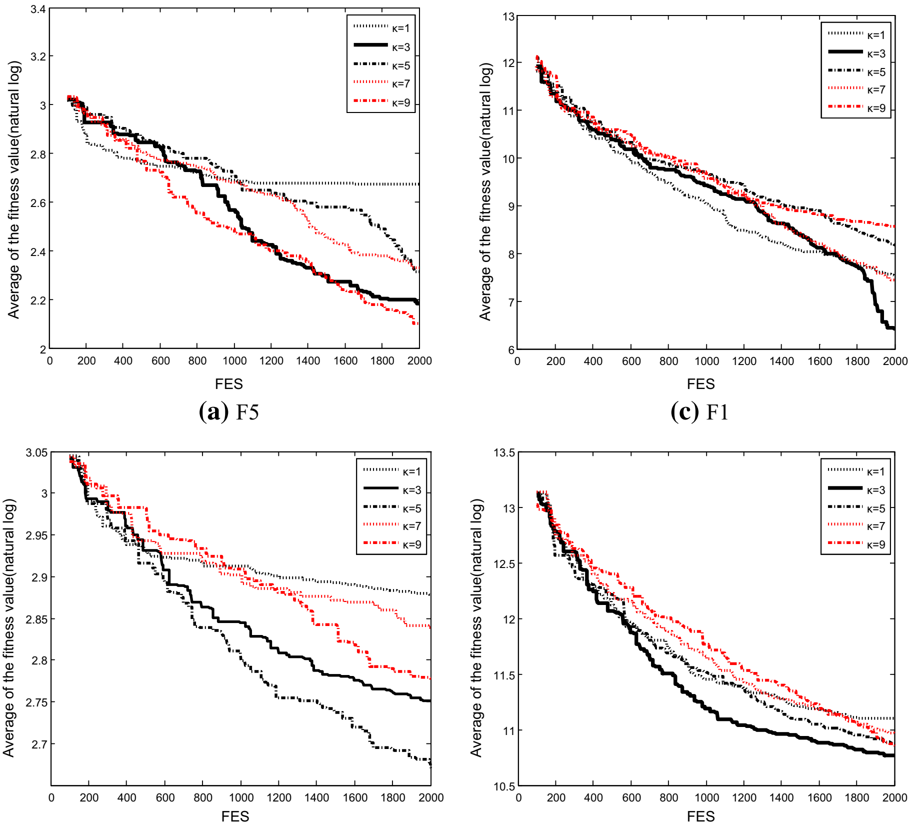
Fig. 2 The performances achieved by TASEA with different pre-screening strategies

plot the convergence curves of TASEA (the corresponding experimental settings are given in the next subsection) in Fig. 2. It is clear that the EI pre-screening strategy can achieve the fastest convergence than the other two pre-screening strategies. Therefore, we use the EI pre-screening strategy in our proposed method.

4.3 Parameter settings

The parameters of the proposed algorithm used in our experiments are set as follows.

1. The initial samples are randomly generated by LHS. We need enough initial samples uniformly distributed in the entire design space. Hence, we set the size of the initial population β to 100 in the experiments.
2. The size of population Pop has been suggested that $30 \leq \alpha \leq 60$ works well in literature [27], a large α value causes slow convergence speed and a small value can result in premature convergence. Hence, following [27], we set α to 50.
3. The number of training data points is very important for GP models. Generally speaking, the more training data points we use, the higher the quality of GP model will be, while the computational cost will be higher simultaneously. Moreover, it is not necessary to build a very accurate global GP model for pre-screening. We only require that the global GP



(a) F5 **(c)** F1

(b) F16 **(d)** F13

Fig. 3 The average function values achieved by TASEA with different κ values

model can guide the search enter into the promising region. Hence, we set the number of training points of global GP model to 100 (e.g. $\mu = 100$) in order to strike a balance between the model quality and the computational cost. However, the accuracy of local GP model should be paid more attention than the global one. We require that the accuracy of local GP model is relatively higher in the promising regions but not in the entire design space. The computational cost of GP modeling in reduced space is much lower than that required in original space. Therefore, we gradually increase the number of training data points of the local GP model to improve accuracy during the iteration process, and the maximum value of parameter γ is 200.

4. Following [45], ω used in LCB is set to 2.
5. The parameter l is the dimension of the reduced low-dimensional space in Sammon mapping. Following [27], $l > 6$ can make the search slow and $l = 2$ can lead to poor results. Hence, we set $l = 4$.
6. Based on the above parameter settings, we have tested five different κ values: 1, 3, 5, 7 and 9 on four functions. These four test functions involve uni-modal and complicated multi-modal problems. The function values of the best individual found so far versus the number of iterations is plotted in Fig. 3. It is clear from Fig. 3 that the convergence

speed is not monotonic as the parameter κ changes. This indicates that the best estimated individual chosen by the GP model with EI pre-screening strategy may not be the true best individual evaluated by the real function. However, the true best individual may be among the top κ best estimated individuals at every iteration. Therefore, we set the parameter κ to 3 for comprehensively considering the performances achieved by our proposed methods with different κ values.

4.4 Experimental results on 50-dimensional problems

The statistical results of all the algorithms are listed in Table 2, including the results of the Wilcoxon rank sum tests calculated at a significant level of $\alpha = 0.05$. Figure 4 plots the convergence profiles of the compared algorithms and the proposed algorithm on 50-dimensional benchmark problems.

In Table 2, '+' indicates that the proposed algorithm is significantly better than the compared algorithms, '−' means that TASEA is significantly outperformed by the compared algorithms according to a Wilcoxon rank sum test, while '≈' indicates that there is no statistically significant difference between the results obtained by the proposed algorithm and the compared algorithms.

It can be seen from Table 2 that TASEA has achieved significantly better or comparative results than FESPSO, CSM-JADE and GPEME on all of the 50-dimensional benchmark problems. Furthermore, the TASEA obtains the smallest optimal value on most of the uni-modal and multi-modal test problems, except on F5 and F10. The reason why this does not happens to functions F5 and F10 is due to the fact that their fitness landscape is nearly a plateau in most of the region close to the global optimum and the optimum is located in a very narrow region near the origin. Such fitness landscape is very hard for SAEAs to attain the global or local optimum under extremely limited computational cost.

In order to gain deeper insight into the performance of the proposed algorithm, we plot the convergence profiles of the compared algorithms and the proposed algorithm in Fig. 4. We can make the following observations regarding the performances of the compared algorithms. Firstly, it can be easily seen from Fig. 4 that the length of the flat part of the convergence curve of the proposed algorithm on most of the 50-dimensonal test benchmark problems is much shorter than those obtained by compared algorithms, especially on functions F3, F4, F5, F6, F7, F8, F9, F10 and F12. The longer the flat part of the convergence curve is, the worse the ability of the algorithm to jump out of the local optimum is. Hence, the feedback mechanism of TASEA can adjust the search strategy of the algorithm promptly while the optimization is trapped into local optimum. Secondly, the number of the flat part of the convergence curve may to some extent be a representation of the global search ability. The more the number of the flat part of the convergence curve, the global search ability may be relatively smaller. It can be seen from Fig. 4 that the number of the flat part of the convergence curve achieved by TASEA is much less than those achieved by the compared algorithms, especially on functions F1, F3, F4, F6, F7, F8, F9 and F12. Hence, the combination of the feedback mechanism and the two DE search strategies, such as DE/current-to-best/1 and DE/current-to-randbest/1, can achieve more effective global search ability. Thirdly, if there is almost no relatively long flat part in the convergence curve, it means the combination of the DE/current-to-randbest/1 strategy and global GP model is not used in our proposed algorithm. Therefore, in the situation where there are few relatively long flat part in the convergence curve, the local search ability of the algorithm might be stronger when the average slope of the convergence curve becomes larger. It can be seen from Fig. 4 that the average slope of the convergence curve achieved by

Table 2 Comparisons of the statistical results on 50-D test benchmark problems

Problem	Approach	Mean (Wilcoxon test)	Std.	Test
F1	FESPSO	8.9184E+04	9.0677E+03	+
	CSM-JADE	3.2047E+04	2.8966E+03	+
	GPEME	4.7894E+03	1.9849E+03	+
	TASEA	**2.8076E+02**	4.3142E+01	
F2	FESPSO	2.6758E+04	1.0428E+04	+
	CSM-JADE	4.6016E+03	8.9147E+02	+
	GPEME	1.0111E+03	5.3948E+02	+
	TASEA	− **4.7468E+02**	9.1919E+01	
F3	FESPSO	1.7082E+03	3.4917E+02	+
	CSM-JADE	3.9231E+02	1.4200E+02	+
	GPEME	1.6108E+02	8.1497E+01	+
	TASEA	**6.5123E+01**	2.6302E+01	
F4	FESPSO	1.7336E+03	4.0294E+02	+
	CSM-JADE	3.3408E+02	5.0325E+01	+
	GPEME	3.1127E+02	9.2792E+01	+
	TASEA	**2.0688E+02**	6.6004E+01	
F5	FESPSO	1.7604E+01	6.5760E−01	+
	CSM-JADE	1.1453E+01	4.0610E−01	≈
	GPEME	1.3646E+01	1.6755E+00	+
	TASEA	**1.0475E+01**	1.7687E+00	
F6	FESPSO	2.6493E+02	6.7444E+01	+
	CSM-JADE	5.0636E+01	8.0988E+00	+
	GPEME	2.5260E+01	9.3856E+00	+
	TASEA	**1.0402E+01**	4.7006E+00	
F7	FESPSO	1.1634E+04	1.5974E+03	+
	CSM-JADE	4.0819E+03	3.9087E+02	+
	GPEME	1.5903E+03	2.4612E+02	+
	TASEA	**3.0705E+02**	1.4507E+02	
F8	FESPSO	2.1839E+06	1.3370E+06	+
	CSM-JADE	8.6523E+04	5.0161E+04	+
	GPEME	4.6268E+04	3.6096E+04	+
	TASEA	**2.6808E+03**	1.0985E+03	
F9	FESPSO	9.4892E+02	8.4919E+01	+
	CSM-JADE	5.2498E+02	4.7770E+01	+
	GPEME	2.4304E+02	1.2560E+02	+
	TASEA	**2.1671E+02**	8.5909E+01	
F10	FESPSO	1.3614E+03	1.1463E+02	+
	CSM-JADE	9.8860E+02	3.0527E+01	+
	GPEME	5.0417E+02	8.5233E+01	≈
	TASEA	**5.5629E+02**	2.3214E+01	

Table 2 continued

Problem	Approach	Mean (Wilcoxon test)	Std.	Test
F11	FESPSO	1.2639E+03	1.0943E+02	+
	CSM-JADE	8.7492E+02	4.9758E+01	+
	GPEME	1.0461E+03	2.8800E+01	+
	TASEA	**4.9047E+02**	4.0898E+01	
F12	FESPSO	1.2335E+04	1.1382E+03	+
	CSM-JADE	7.7056E+03	4.5643E+02	+
	GPEME	7.2312E+03	2.0615E+03	+
	TASEA	**4.5214E+03**	2.0306E+03	

Bold values indicate the results of the proposed method

TASEA is larger than those achieved by the other three algorithms, especially on functions F1, F3, F6 and F7. Therefore, the adaptive local GP model and DE/best/1 strategy are combined effectively to achieve faster local convergence speed.

In Fig. 4, the convergence curves of different algorithms are drawn with different line type and line width. Note that the population sizes of the algorithms under comparison are not exactly the same, so the initial best fitness values of different algorithms are different.

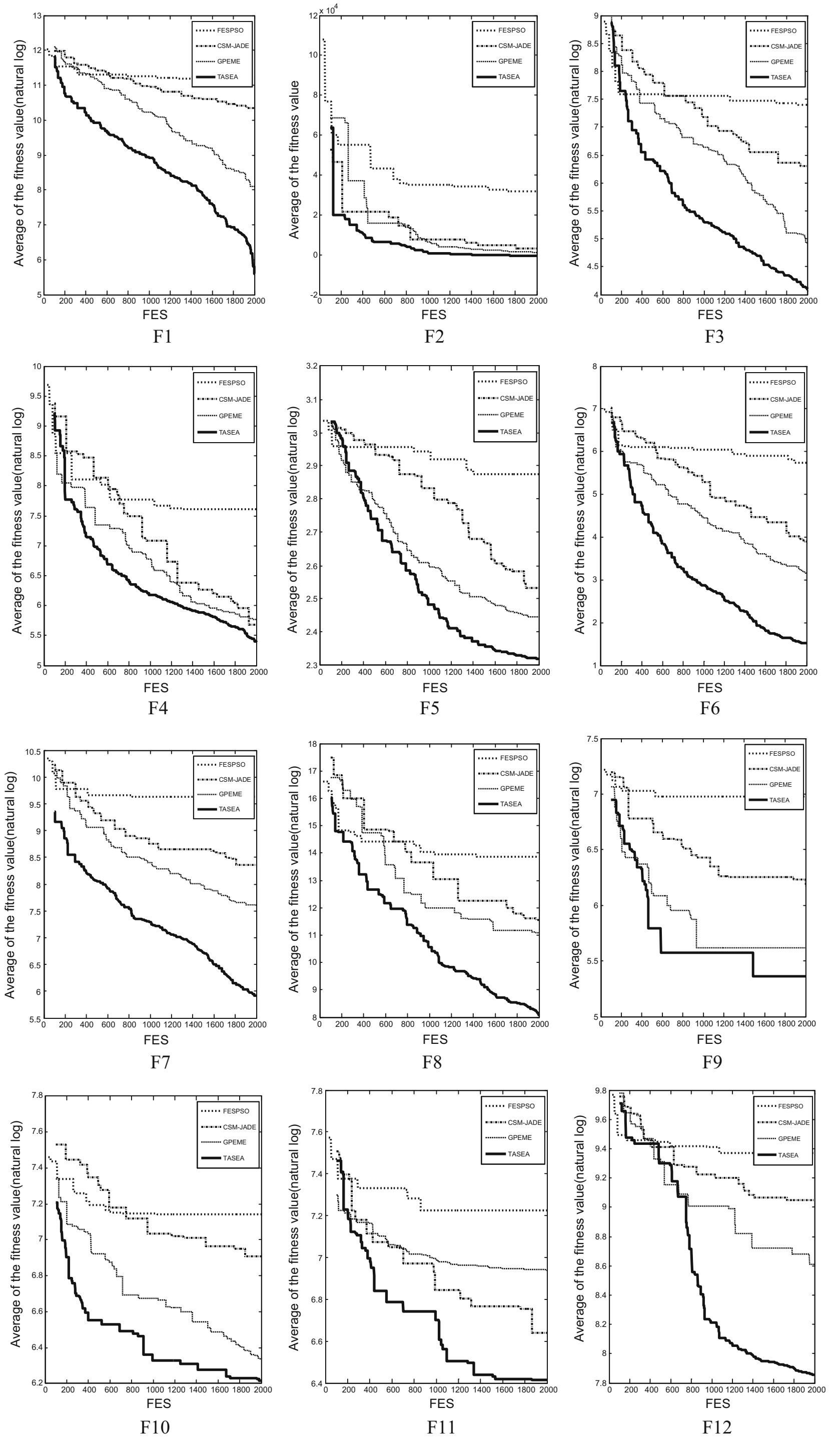
4.5 Experimental results on 100-dimensional problems

In the following, we test the performances of the proposed TASEA on 100-dimensional benchmark problems with extremely limited computational budget. Table 3 lists the statistical results of all the algorithms under comparison with averaged 20 independent runs. The test results of Wilcoxon rank sum tests are calculated at a significant level of $\alpha = 0.05$. Figure 5 plots the convergence profiles of the compared algorithms and the proposed algorithm on 100-dimensional benchmark problems.

In Table 3 and Fig. 5, as we can see, similar test results as the 50-dimensioal test problems are shown. TASEA outperforms other algorithms on the aspects of convergence speed, global optimization ability and local search ability. While for 100-dimensional problems, the dimensional reduction technique plays an important role in the construct of the relatively high accurate local GP model for fast convergence under limited computational budget.

4.6 Comparison with other SAEAs for high-dimensional computationally expensive problems

There are two algorithms proposed recently to solve high-dimensional computationally expensive problems. The first one is a surrogate-based optimization approach named as USGD [70]. It mainly deals with problems with dimensions reaching hundreds or thousands (e.g., 200- and 1000-dimensional problems). Moreover, the USGD does not search for the optimal solution through evolutionary algorithms. Hence, from the differences in both dimensions of tested problems and methodology, it may be not necessary to compare it with TASEA. The second one is surrogate-assisted cooperative particle swarm optimization (SA-COSO) [71] for solving expensive problems with dimensions ranging from 50 and 200. Then a direct comparison has been made between TASEA and SA-COSO. Considering that there are many



F1

F2

F3

F4

F5

F6

F7

F8

F9

F10

F11

F12

Fig. 4 The convergence profiles for 50-dimensonal F1–F12 of four algorithms

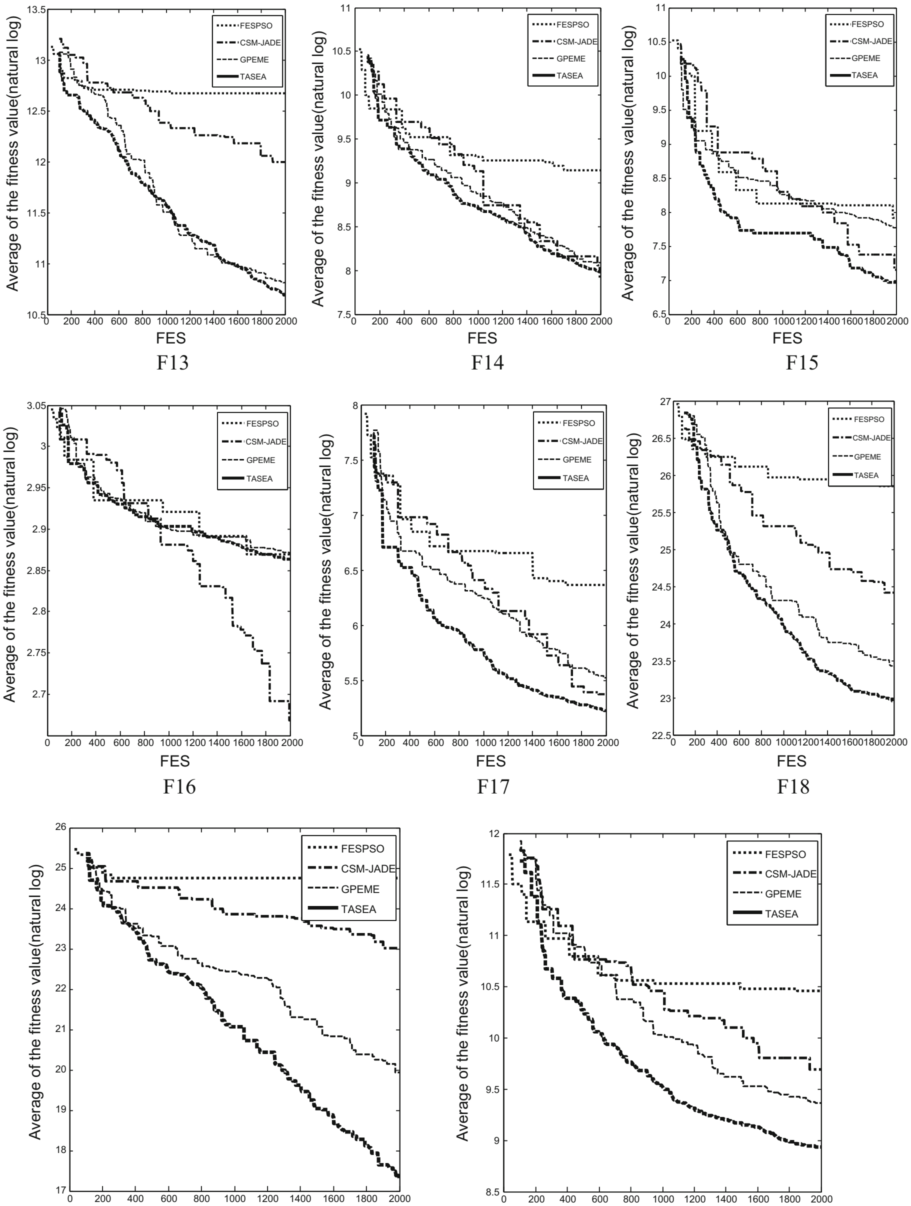
Table 3 Comparisons of the statistical results on 100-D test benchmark problems

Problem	Approach	Mean (Wilcoxon test)	Std.	Test
F13	FESPSO	3.4825E+05	3.8694E+04	+
	CSM-JADE	1.8313E+05	1.0588E+04	+
	GPEME	6.3710E+04	1.7427E+04	+
	TASEA	**4.5169E+04**	9.9660E+03	
F14	FESPSO	8.9741E+03	2.1583E+03	+
	CSM-JADE	3.1631E+03	3.8717E+02	+
	GPEME	3.3730E+03	1.0026E+03	+
	TASEA	**2.2676E+03**	6.5120E+02	
F15	FESPSO	4.1203E+03	9.6310E+02	+
	CSM-JADE	1.4381E+03	2.1322E+02	≈
	GPEME	3.2170E+03	5.3759E+02	+
	TASEA	**1.3277E+03**	3.5527E+02	
F16	FESPSO	1.7974E+01	5.6220E−01	≈
	CSM-JADE	1.4636E+01	3.7180E−01	−
	GPEME	1.8372E+01	5.2430E−01	+
	TASEA	**1.6303E+01**	1.2515E+00	
F17	FESPSO	6.0628E+02	1.3860E+02	+
	CSM-JADE	2.1728E+02	2.9452E+01	+
	GPEME	2.6564E+02	4.4382E+01	+
	TASEA	**1.5168E+02**	3.7383E+01	
F18	FESPSO	1.4268E+11	2.0117E+10	+
	CSM-JADE	4.7065E+10	6.1963E+09	+
	GPEME	2.3219E+10	6.9143E+09	+
	TASEA	**9.1363E+09**	8.5547E+08	
F19	FESPSO	4.5530E+10	1.0636E+10	+
	CSM-JADE	1.0991E+10	1.5456E+09	+
	GPEME	3.1501E+08	2.0424E+08	+
	TASEA	**7.5404E+07**	7.8384E+07	
F20	FESPSO	4.0522E+04	4.88E+03	+
	CSM-JADE	1.7389E+04	1.51E+03	+
	GPEME	1.1811E+04	1.5443E+03	+
	TASEA	**8.5826E+03**	1.5623E+03	

Bold values indicate the results of the proposed method

factors that can significantly affect the convergence performance of SA-COSO when programming this method. Therefore, only the 6 test problems and the SA-COSO results from the original paper are used (Table 4). The maximum number of fitness evaluations is set to 1000. The statistical results of these two algorithms are listed in Tables 5 and 6, including the t test values calculated at a significant level of $\alpha = 0.05$.

Generally speaking, for 50-dimensional problems, it can be seen that the convergence results of TASEA are worse than those of SA-COSO on four test problems such as F1, F2, F3, and F4, comparable to those of SA-COSO on problem F5, and better than those of SA-COSO on problem F6. For 100-dimensional problems, it can be seen that the convergence results of TASEA are worse than those of SA-COSO on three problems such as F1, F3, and



F13 F14 F15

F16 F17 F18

F19 F20

Fig. 5 The convergence profiles for 100-dimensonal F13–F20 from four algorithms

F4, and better than those of SA-COSO on the other three problems such as F2, F5, and F6. To sum up, although our proposed TASEA performs a little worse on some uni-modal or relatively simple test function such as F1, F3, and F4, it may be more suitable for solving some complicated and higher dimensional multi-modal problems such as F5 and F6. This

Table 4 Characteristics of six benchmark problems

Problem	Description	Dimensions	Global optimum	Characteristics
F1 [4]	Ellipsoid	50/100	0	Unimodal
F2 [32]	Rosenbrock	50/100	0	Multimodal with narrow valley
F3 [27]	Ackley	50/100	0	Multimodal
F4 [27, 28, 32]	Griewank	50/100	0	Multimodal
F5 [27, 28]	Shifted Rotated Rastrigin	50/100	− 330	Very complicated multimodal
F6 [27, 28]	Rotated hybrid Composition Function	50/100	120	Very complicated multimodal

Table 5 Comparison results on 50-dimensional benchmark problems

Problem	Approach	Mean	Std.	t-test
F1	SA-COSO	5.148E+01	1.625E+01	−
	TASEA	1.882E+02	1.496E+01	
F2	SA-COSO	2.526E+02	4.074E+01	−
	TASEA	4.349E+02	9.124E+01	
F3	SA-COSO	8.932E+00	1.067E+00	−
	TASEA	1.186E+01	2.947E−01	
F4	SA-COSO	6.006E+00	1.104E+00	−
	TASEA	1.564E+01	7.826E−01	
F5	SA-COSO	1.972E+02	3.060E+01	≈
	TASEA	2.595E+02	1.158E+02	
F6	SA-COSO	1.081E+03	3.286E+01	+
	TASEA	5.640E+02	1.120E+01	

may be attributed to that our proposed TASEA can fully utilize the feedback information of the iterative process and can also use dimension reduction strategy to ease pre-screen accuracy insufficiency brought about by high-dimensional searching.

4.7 Sensitivity in relation to the parameter F and C_r

To further investigate the differences brought by different DE parameter settings, we conduct some experiments on four representative examples with different F and C_r. Considering that there are three search strategies presented in TASEA, we analyze the sensitivity of the parameters of one strategy by fixing the parameter values of the other two strategies. The DE/current-to-best/1 strategy should balance the global and local search abilities. Both the information of the current best individual and random individuals should be emphasized. So we test this strategy with different F: 0.6, 0.7, 0.8, 0.9 and 1.0, and different C_r: 0.5, 0.6, 0.7, 0.8 and 0.9. The DE/current-to-randbest/1 strategy should achieve stronger global search ability, so we test this strategy with different F: 0.6, 0.7, 0.8, 0.9 and 1.0, and different C_r: 0.5, 0.6, 0.7, 0.8 and 0.9. The DE/best/1 strategy should achieve stronger local search ability, so we test this strategy with different F: 0.1, 0.2, 0.3, 0.4 and 0.5, and different C_r:0.5, 0.6, 0.7, 0.8 and 0.9. Four test functions (i.e., F1, F5, F13, F16) were selected to test the performances

Table 6 Comparison results on 100-dimensional benchmark problems

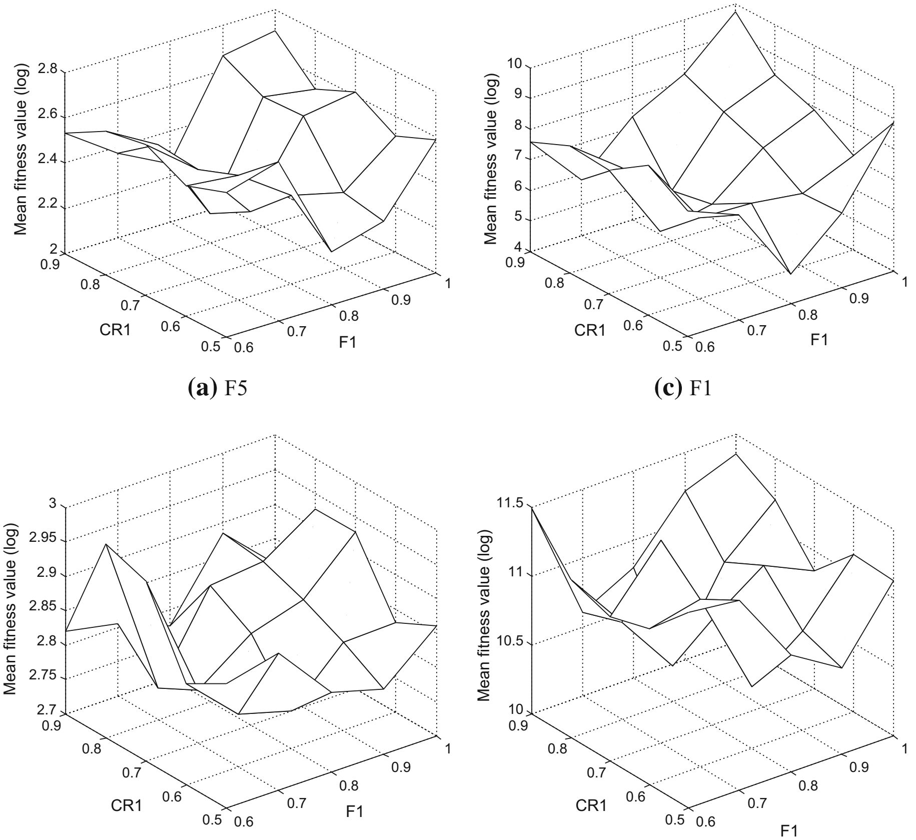
Problem	Approach	Mean	Std.	t-test
F1	SA-COSO	1.033E+03	3.172E+02	–
	TASEA	5.141E+03	6.858E+02	
F2	SA-COSO	2.714E+03	1.170E+02	+
	TASEA	2.104E+03	7.859E+01	
F3	SA-COSO	1.576E+01	5.025E−01	–
	TASEA	1.751E+01	8.456E−01	
F4	SA-COSO	6.335E+01	1.902E+01	–
	TASEA	2.660E+02	5.230E+01	
F5	SA-COSO	1.273E+03	1.172E+02	+
	TASEA	8.935E+02	3.743E+01	
F6	SA-COSO	1.366E+03	3.087E+01	+
	TASEA	1.171E+03	1.571E+01	

of TASEA with different F and C_r. These test functions involve uni-modal and complicated multi-modal problems, in which two of them are 50-dimensional problems and the other two are 100-dimensional problems. Figure 6 shows the performances of TASEA with different F and C_r values in DE/current-to-best/1 strategy. Figure 7 shows the performances of TASEA with different F and C_r values in DE/current-to-randbest/1 strategy. Figure 8 shows the performances of TASEA with different F and C_r values in DE/best/1 strategy.

In Fig. 6, it is obvious that the change of the parameter F and C_r in the DE/current-to-best/1 strategy is relatively sensitive to the performances of TASEA. Furthermore, the parameter F and C_r all should not to be set too larger or smaller because the DE/current-to-best/1 strategy should balance local and global search abilities. In general, the value of F is recommended in the interval [0.7, 0.9] and the value of C_r is recommended in the interval [0.6, 0.8]. In DE/current-to-randbest/1 strategy, the parameter F should be set bigger than 0.5 in order to pay more attention to the diversity of population than the information brought by the current best individual. Similarly, the parameter C_r also should be set to relatively bigger to achieve global search. Therefore, combining the results presented in Fig. 7, we suggest that the value of F is recommended in the interval [0.7, 0.9] and the value of C_r is recommended in the interval [0.6, 0.9]. In Fig. 8, we can see that TASEA is relatively sensitive to the two parameters in DE/best/1 strategy. In order to achieve stronger local search, the parameter F and C_r should be chosen in a reasonable range. Hence, the value of F is recommended in the interval [0.2, 0.4], and the value of C_r is recommended in the interval [0.5, 0.9].

4.8 Sensitivity in relation to the parameter N_c

As one of the most important parameters for the proposed algorithm, the parameter N_c determines when to adjust the search strategy of the algorithm during the iteration. If N_c is set to a larger value, the DE/current-to-randbest/1 strategy cannot be used in time to explore more regions, which may make the optimization trapped into local optimum with considerable possibility. However, if the value of N_c is set too small, the DE/current-to-randbest/1 strategy may be used frequently, which leads to lower convergence speed. Therefore, the appropriate parameter setting can make the algorithm jump out of local optimal region in time during iteration.



(a) F5

(c) F1

(b) F16

(d) F13

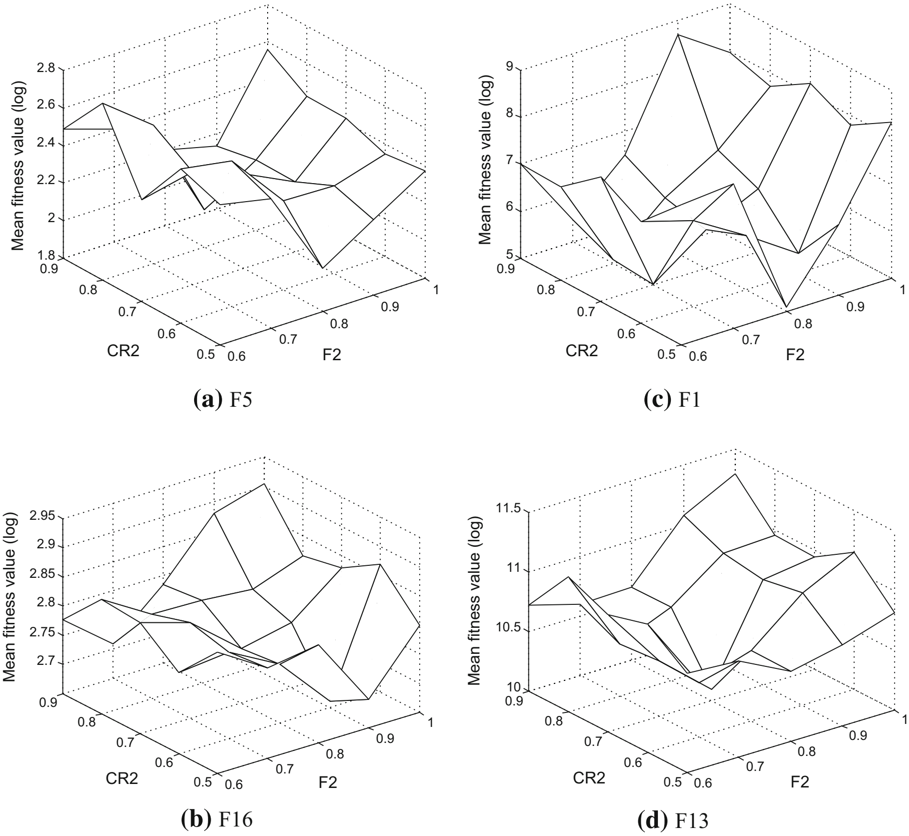
Fig. 6 The performances achieved by TASEA with different combination of F and C_r in DE/current-to-best/1 strategy

In order to investigate the sensitivity of this parameter, we tested TASEA with different N_c: 10, 15, 20, 25, 30, 35, 40, 45 and 50. Similarly, four test functions (i.e., F1, F5, F13, F16) were selected to test the performance of TASEA with different N_c. Figure 9 shows the performances of TASEA with different N_c values.

From Fig. 9, we can observe that TASEA actually is sensitive to the parameter N_c to some extent, and that N_c can be chosen from a relatively small range to achieve competitive performance for TASEA. In general, the value of N_c is recommended in the interval [20, 30].

4.9 Sensitivity in relation to the parameter *L*

As one of the most important parameters for the proposed algorithm, the parameter L determines the dimensions of low-dimensional space, which is obtained by Sammon mapping. More specifically, if L is set as a larger value, the Sammon mapping can not find the exact location of all individuals in the low-dimensional space within a limited number of iterations. This means that the neighborhood relationships and pairwise distances among the individuals will not be preserved effectively. Hence, the GP models constructed in low-dimensional



(a) F5

(c) F1

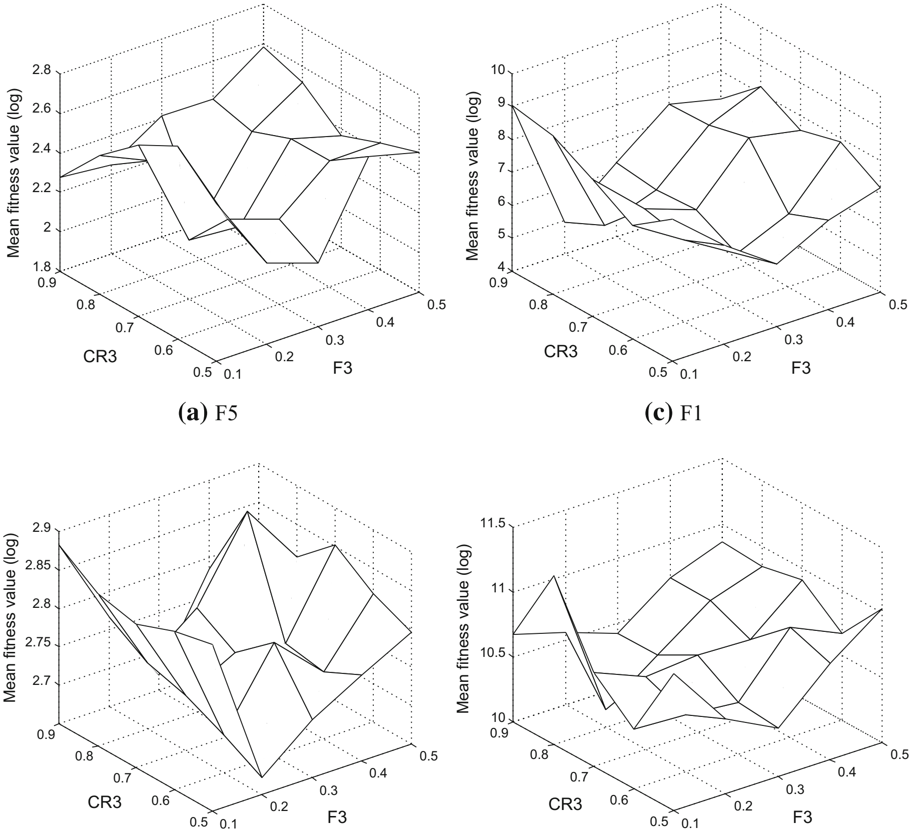
(b) F16

(d) F13

Fig. 7 The performances achieved by TASEA with different combination of F and C_r in DE/current-to-randbest/1 strategy

space may not be able to effectively pre-screen offspring individuals. In addition, the extra computation cost will increase when the Sammon mapping is used to locate the positions of individuals in low-dimensional space. And the modeling time in low-dimensional space will also be greatly increased, especially for the local GP model constructed with a large number of training sample points. However, if the value of L is set too small, large amounts of information between individuals will lose during dimension reduction technique of Sammon mapping. Hence, the GP models constructed in low-dimensional space may also not be able to effectively pre-screen offspring solutions.

Based on the above considerations, we tested TASEA with different L: 2, 4, 6, 8, 10, 15, 20, 25. Four typical test problems (i.e., F1, F5, F13, F16) were selected to test the performances of TASEA with different L. Moreover, in order to further illustrate the influences of dimensionality reduction (DR) on search performance of the proposed algorithm, we record the “Pre-screen accuracy” of TASEA with a fixed $L = 4$ value on F1. More specifically, The “Pre-screen accuracy” indicates the ratio of the number of real top κ individuals among the predicted top κ individuals selected from the offspring population by model pre-screening to κ. In our paper, the value of κ is 3. In the optimization process, we not only recorded the “Pre-screen accuracy” of the GP model after dimensionality reduction, but also recorded the



(a) F5

(c) F1

(b) F16

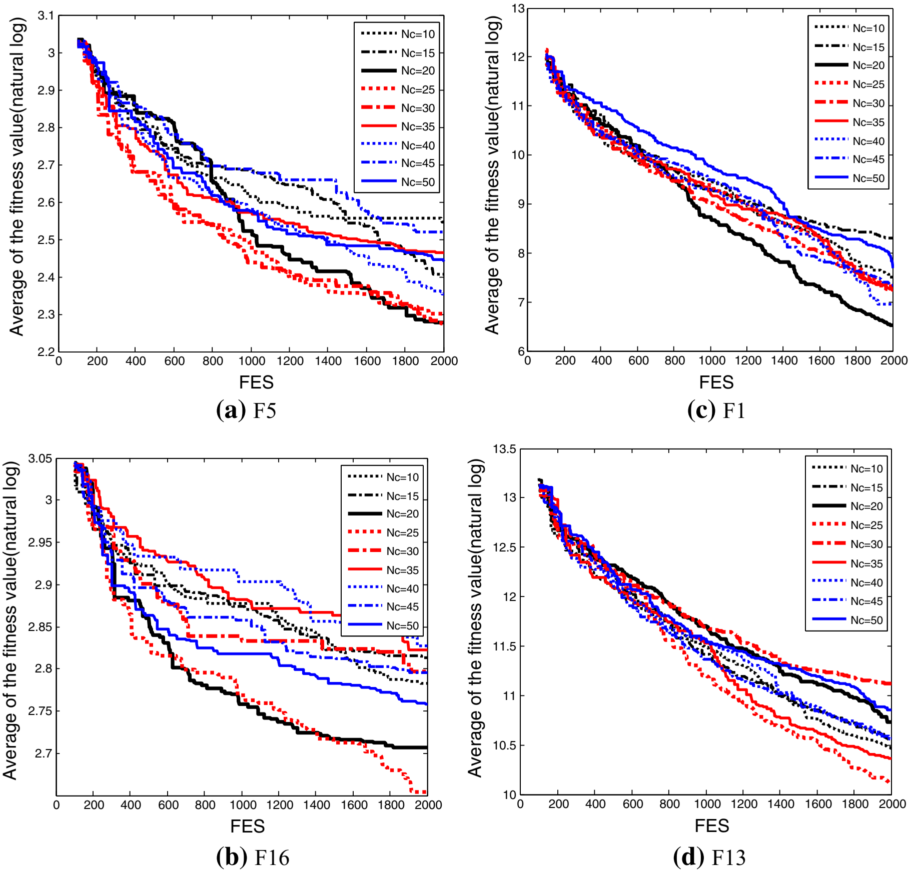
(d) F13

Fig. 8 The performances achieved by TASEA with different combination of F and C_r in DE/best/1 strategy

"Pre-screen accuracy" of the GP model constructed directly by using the training samples in the original high-dimensional space. Figure 10 shows the convergence profiles of TASEA with different L values. Figure 11 shows the curves of "Pre-screen accuracy" of TASEA with a fixed $L = 4$ value on F1.

From Fig. 10, we can observe that the performance degradation tends to occur for these four test functions (i.e., F1, F5, F13, F16) when using a relatively larger value for this parameter. In addition, a relatively small value for this parameter also has a negative effect on the performance since large amounts of information among individuals are missing. Moreover, from the test results obtained by the TASEA with different L values on the tested problems with different dimensions, we can observe that the better results may be obtained by setting a larger value of L when the dimension of the test function is larger. Therefore, the results in Fig. 10 reveal that a value between 4 and 8 is a suitable choice for this parameter when the dimension of problems is 50, and a value between 4 and 10 is a suitable choice for this parameter when the dimension of problems is 100.

From Fig. 11, we can observe that there is no significant difference in the "Pre-screen accuracy" of the GP model with and without dimensionality reduction, which means that the dimension reduction technique does not reduce the "Pre-screen accuracy" of the GP model



(a) F5

(c) F1

(b) F16

(d) F13

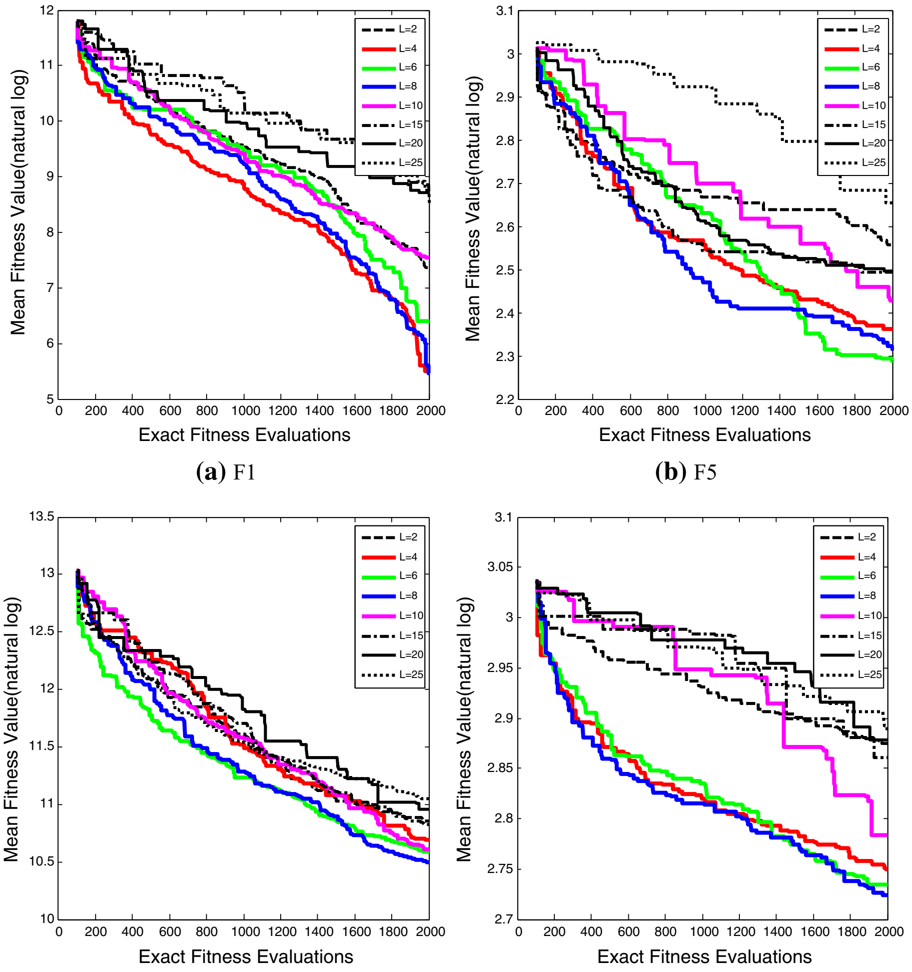
Fig. 9 The average function values achieved by TASEA with different N_c values

to a certain extent. Hence, the introduction of the dimension reduction technique not only reduces the computational cost of constructing a GP model, but also ensures that the GP model constructed in low-dimensional space is also able to pre-screen offspring individuals effectively.

4.10 The structural design optimization of the driving axle of an all-direction propeller

All-direction propellers are widely used in marine equipment for propulsion and positioning. The driving axle, as one of the core components, is used to transfer power and output movement. It endures severe alternating stress during the operation, which may cause the failure of fatigue, so the design of the axle should ensure that the fatigue life of the axle satisfies the service requirement.

The simplified 3D model diagram of the driving axle in an all-direction propeller WSP330-CP is given as Fig. 12 shows, and the design variables are shown in Fig. 13. The design objective is to minimize the mass of the driving axle while keeping a required fatigue life.



(a) F1

(b) F5

(c) F13

(d) F16

Fig. 10 The performances achieved by TASEA with different L values

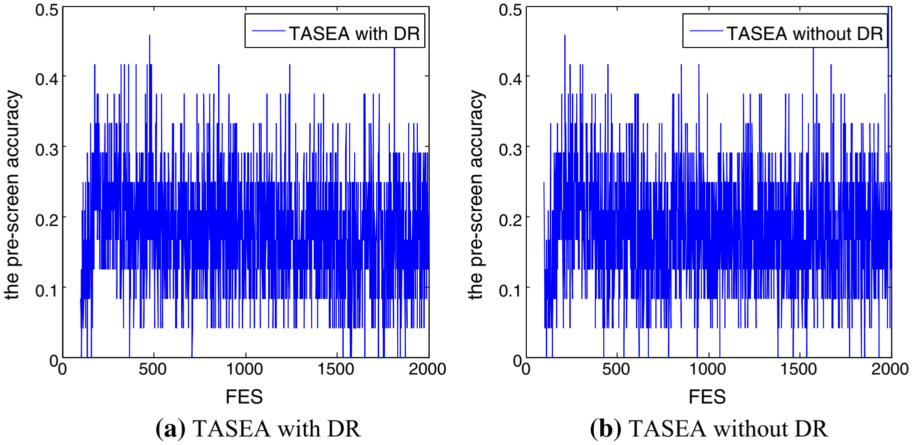
The failure event is defined as the fatigue life is smaller than 15 years. The optimizing problem can be summarized as follows.

$$\text{minmize} \quad the\,mass\,of\,the\,driving\,axle$$

$$s.t. \quad \begin{cases} 165 \leq x_1 \leq 180, \; 850 \leq x_2 \leq 880 \\ 190 \leq x_3 \leq 230, \; 270 \leq x_4 \leq 290 \\ 170 \leq x_5 \leq 185, \; 250 \leq x_6 \leq 280 \\ 65 \leq x_7 \leq 90, \quad 220 \leq x_8 \leq 245 \\ 210 \leq x_9 \leq 230, \; the\,fatigue\,life \geq 15\,(year) \end{cases}$$

where $x_i, i = 1, 2, \ldots, 9$ is the design variables of the driving axle.

The fatigue life of the driving axle needs to be obtained by simulations. The build and computation cost of these simulations is relatively high, so it can be considered to be computa-



(a) TASEA with DR **(b)** TASEA without DR

Fig. 11 The performances achieved by TASEA with a fixed $L = 4$ value on F1

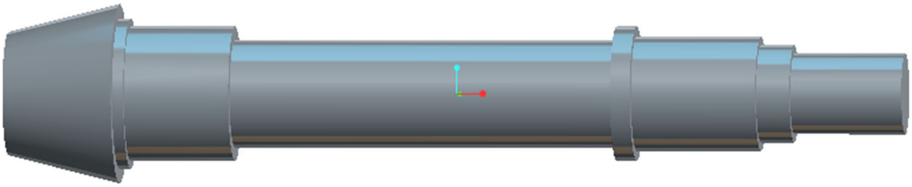


Fig. 12 The simplified 3D model diagram of the driving axle

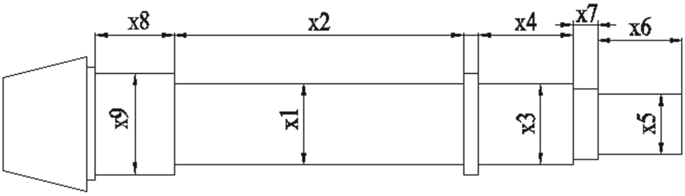


Fig. 13 The design variables of the driving axle

tionally expensive. For this optimization problem, we use penalty function method to deal with the fatigue life constraint, thus the problem can be transformed into an unconstrained computationally expensive single objective optimization problem. Then, the proposed TASEA was used in this problem. Considering that the affordable number of simulations is limited, the maximum number of simulations is set to 300. The same initial population is generated by using Latin hypercube sampling, and all the other settings are the same as those in the original paper.

Table 7 and Fig. 14 demonstrate the results obtained by TASEA on this optimization problem. It can be seen that the mass of the axle is reduced by about 42.2 kg, almost 7.7% of the Initial mass. Therefore, the proposed TASEA can provide an efficient way for solving computationally expensive engineering optimization problems.

Table 7 The results obtained by TASEA

Methods	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	Mass	FF
Initial	179.8	862.3	219.8	271.1	184.7	269.8	88.5	225.8	222.2	550.1	37.8
TASEA	167.7	856.0	200.1	281.1	177.7	256.2	87.5	226.9	229.1	507.9	23.3

Initial means the best solution from the initial population. And FF means the fatigue life of the driving axle

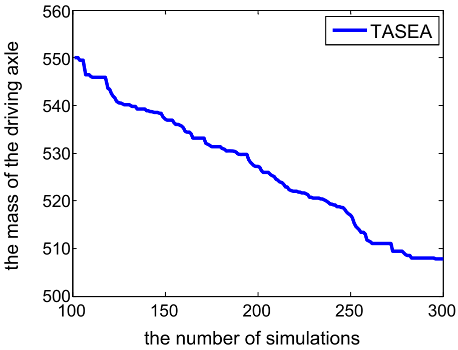


Fig. 14 The convergence profiles from TASEA and GPEME

5 Conclusions

A Two-layer Adaptive Surrogate-assisted Evolutionary Algorithm (TASEA) is proposed in this paper to reduce the computational cost for solving high-dimensional computationally expensive optimization problems. Firstly, the feedback mechanism of TASEA can adjust the search strategy of the algorithm promptly while the optimization is trapped into local optimum. Then the offspring produced by the DE/current-to-best/1 strategy can be effectively pre-screened by the global GP model with EI pre-screening strategy for fast convergence speed, and the DE/current-to-randbest/1 strategy can effectively guide the global GP model to locate promising regions when the feedback information reaches the presetting threshold. Moreover, the local GP model which is built by using individuals closest to the current best individual can intensively exploit the promising regions for fast convergence. The TASEA also uses Sammon mapping for dimension reduction on the original search space for high-dimensional problems, so that the computational overhead for modeling and pre-screening can be reduced significantly and the model quality can be maintained with an affordable number of training data points. Experimental results on twelve 50-dimensional and eight 100-dimensional benchmark problems demonstrate that the efficiency and effectiveness of the proposed TASEA algorithm when compared with three other state-of-art algorithms.

Despite the proposed TASEA has shown promising performance on typical high-dimensional problems, future work is still required to further reduce computational cost and to improve the algorithm performance on high-dimensional multi-modal problems. There are some further research directions that can be promising in the future.

1. Investigation on the use of other cheaper while accurate surrogate modeling methods and other dimension reduction methods for TASEA.
2. Construct new adaptive feedback mechanism that can handle some complicated multi-modal problems, such as the Ackley function whose fitness landscape is nearly a plateau in most of the region close to the global optimum.

3. Combine TASEA with some other evolutionary algorithms, such as PSO, ACO and so on.
4. Generalization of TASEA to computationally expensive objective function with expensive constraints and multi-objective computationally expensive constrained optimization problems.

Acknowledgements This research is supported by the National Natural Science Foundation of China under Grant Nos. 51675198, 51721092, the National Natural Science Foundation for Distinguished Young Scholars of China under Grant No. 51825502, and the Program for HUST Academic Frontier Youth Team.

References

1. El-Ela, A.A., Fetouh, T., Bishr, M., Saleh, R.: Power systems operation using particle swarm optimization technique. Electr. Power Syst. Res. **78**(11), 1906–1913 (2008)
2. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Glob. Optim. **13**(4), 455–492 (1998)
3. Nguyen, S., Zhang, M., Johnston, M., Tan, K.C.: Automatic programming via iterated local search for dynamic job shop scheduling. IEEE Trans. Cybernet. **45**(1), 1–14 (2015)
4. Yoon, Y., Kim, Y.-H.: An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks. IEEE Trans. Cybernet. **43**(5), 1473–1483 (2013)
5. Wu, T.-Y., Lin, C.-H.: Low-SAR path discovery by particle swarm optimization algorithm in wireless body area networks. IEEE Sens. J. **15**(2), 928–936 (2015)
6. He, S., Prempain, E., Wu, Q.: An improved particle swarm optimizer for mechanical design optimization problems. Eng. Optim. **36**(5), 585–605 (2004)
7. Lim, D., Jin, Y., Ong, Y.-S., Sendhoff, B.: Generalizing surrogate-assisted evolutionary computation. IEEE Trans. Evol. Comput. **14**(3), 329–355 (2010)
8. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. Soft Comput. A Fusion Found. Methodol. Appl. **9**(1), 3–12 (2005)
9. Gaspar-Cunha, A., Vieira, A.: A Hybrid Multi-objective evolutionary algorithm using an inverse neural network. In: Hybrid Metaheuristics, pp. 25–30 (2004)
10. Gaspar-Cunha, A., Vieira, A.: A multi-objective evolutionary algorithm using neural networks to approximate fitness evaluations. Int. J. Comput. Syst. Signal **6**(1), 18–36 (2005)
11. Lian, Y., Liou, M.-S.: Multiobjective optimization using coupled response surface model and evolutionary algorithm. AIAA J. **43**(6), 1316–1325 (2005)
12. Loshchilov, I., Schoenauer, M., Sebag, M.: A mono surrogate for multiobjective optimization. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, pp. 471–478. ACM (2010)
13. Herrera, M., Guglielmetti, A., Xiao, M., Coelho, R.F.: Metamodel-assisted optimization based on multiple kernel regression for mixed variables. Struct. Multidiscip. Optim. **49**(6), 979–991 (2014)
14. Isaacs, A., Ray, T., Smith, W.: An evolutionary algorithm with spatially distributed surrogates for multiobjective optimization. In: Australian Conference on Artificial Life, pp. 257–268. Springer (2007)
15. Zapotecas Martínez, S., Coello Coello, C.A.: MOEA/D assisted by RBF networks for expensive multiobjective optimization problems. In: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, pp. 1405–1412. ACM (2013)
16. Knowles, J.: ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans. Evol. Comput. **10**(1), 50–66 (2006)
17. Ponweiser, W., Wagner, T., Biermann, D., Vincze, M.: Multiobjective optimization on a limited budget of evaluations using model-assisted S-metric selection. In: International Conference on Parallel Problem Solving from Nature, pp. 784–794. Springer (2008)
18. Zhang, Q., Liu, W., Tsang, E., Virginas, B.: Expensive multiobjective optimization by MOEA/D with Gaussian process model. IEEE Trans. Evol. Comput. **14**(3), 456–474 (2010)
19. Ahmed, M., Qin, N.: Surrogate-based multi-objective aerothermodynamic design optimization of hypersonic spiked bodies. AIAA J. **50**(4), 797–810 (2012)
20. Ratle, A.: Kriging as a surrogate fitness landscape in evolutionary optimization. AI EDAM **15**(01), 37–49 (2001)
21. Jin, Y., Olhofer, M., Sendhoff, B.: A framework for evolutionary optimization with approximate fitness functions. IEEE Trans. Evol. Comput. **6**(5), 481–494 (2002)

22. Ulmer, H., Streichert, F., Zell, A.: Evolution strategies assisted by Gaussian processes with improved preselection criterion. In: Evolutionary Computation. CEC'03. The 2003 Congress on 2003, pp. 692–699. IEEE (2003)
23. Karakasis, M., Giannakoglou, K.: On the use of metamodel-assisted, multi-objective evolutionary algorithms. Eng. Optim. **38**(8), 941–957 (2006)
24. Parno, M.D., Fowler, K.R., Hemker, T.: Framework for particle swarm optimization with surrogate functions. Darmstadt Technical University, Darmstadt (2009)
25. Jin, Y.: Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm Evolut. Comput. **1**(2), 61–70 (2011)
26. Di Nuovo, A., Ascia, G., Catania, V.: A study on evolutionary multi-objective optimization with fuzzy approximation for computational expensive problems. In: Parallel Problem Solving from Nature-PPSN XII, pp. 102–111 (2012)
27. Liu, B., Zhang, Q., Gielen, G.G.: A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. IEEE Trans. Evol. Comput. **18**(2), 180–192 (2014)
28. Gong, W., Zhou, A., Cai, Z.: A multioperator search strategy based on cheap surrogate models for evolutionary optimization. IEEE Trans. Evol. Comput. **19**(5), 746–758 (2015)
29. Ong, Y.S., Nair, P.B., Keane, A.J.: Evolutionary optimization of computationally expensive problems via surrogate modeling. AIAA J. **41**(4), 687–696 (2003)
30. Smith, R.E., Dike, B.A., Stegmann, S.: Fitness inheritance in genetic algorithms. In: Proceedings of the 1995 ACM Symposium on Applied Computing, pp. 345–350. ACM (1995)
31. Hendtlass, T.: Fitness estimation and the particle swarm optimisation algorithm. In: Evolutionary Computation. CEC 2007. IEEE Congress on 2007, pp. 4266–4272. IEEE (2007)
32. Sun, C., Zeng, J., Pan, J., Xue, S., Jin, Y.: A new fitness estimation strategy for particle swarm optimization. Inf. Sci. **221**, 355–370 (2013)
33. Zhou, Z., Ong, Y.S., Nguyen, M.H., Lim, D.: A study on polynomial regression and Gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm. In: Evolutionary Computation. The 2005 IEEE Congress on 2005, pp. 2832–2839. IEEE (2005)
34. Tenne, Y., Armfield, S.W.: A framework for memetic optimization using variable global and local surrogate models. Soft Comput. A Fusion Found. Methodol. Appl. **13**(8), 781–793 (2009)
35. Müller, J., Shoemaker, C.A.: Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems. J. Glob. Optim. **60**(2), 123–144 (2014)
36. Sun, C., Jin, Y., Zeng, J., Yu, Y.: A two-layer surrogate-assisted particle swarm optimization algorithm. Soft. Comput. **19**(6), 1461–1475 (2015)
37. Bouhlel, M.A., Bartoli, N., Otsmane, A., Morlier, J.: Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction. Struct. Multidiscip. Optim. **53**(5), 935–952 (2016)
38. Regis, R.G.: Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. Eng. Optim. **46**(2), 218–243 (2014)
39. Regis, R.G.: Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions. IEEE Trans. Evol. Comput. **18**(3), 326–347 (2014)
40. Liu, B., Koziel, S., Zhang, Q.: A multi-fidelity surrogate-model-assisted evolutionary algorithm for computationally expensive optimization problems. J. Comput. Sci. **12**, 28–37 (2016)
41. Jin, C., Qin, A.K., Tang, K.: Local ensemble surrogate assisted crowding differential evolution. In: Evolutionary Computation (CEC), IEEE Congress on 2015, pp. 433–440. IEEE (2015)
42. Awad, N.H., Ali, M.Z., Mallipeddi, R., Suganthan, P.N.: An improved differential evolution algorithm using efficient adapted surrogate model for numerical optimization. Inf. Sci. **451**, 326–347 (2018)
43. Elsayed, S.M., Ray, T., Sarker, R.A.: A surrogate-assisted differential evolution algorithm with dynamic parameters selection for solving expensive optimization problems. In: Evolutionary Computation (CEC), IEEE Congress on 2014, pp. 1062–1068. IEEE (2014)
44. Mallipeddi, R., Lee, M.: An evolving surrogate model-based differential evolution algorithm. Appl. Soft Comput. **34**, 770–787 (2015)
45. Dennis, J., Torczon, V.: Managing approximation models in optimization. In: Multidisciplinary Design Optimization: State-of-the-Art, pp. 330–347 (1997)
46. Forrester, A., Sobester, A., Keane, A.: Engineering Design via Surrogate Modelling: A Practical Guide. Wiley, New York (2008)
47. Viana, F.A., Haftka, R.T., Watson, L.T.: Efficient global optimization algorithm assisted by multiple surrogate techniques. J. Glob. Optim. **56**(2), 669–689 (2013)
48. Rasmussen, C.E.: Gaussian processes in machine learning. In: Advanced Lectures on Machine Learning, pp. 63–71. Springer (2004)

49. Lophaven, S.N., Nielsen, H.B., Søndergaard, J.: DACE-A Matlab Kriging toolbox, version 2.0. In. (2002)
50. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. Stat. Sci. **4**, 409–423 (1989)
51. Van Der Maaten, L., Postma, E., Van den Herik, J.: Dimensionality reduction: a comparative. J. Mach. Learn. Res. **10**, 66–71 (2009)
52. Sammon, J.W.: A nonlinear mapping for data structure analysis. IEEE Trans. Comput. **100**(5), 401–409 (1969)
53. Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J.: SOM Toolbox for Matlab 5. Helsinki University of Technology, Espoo (2000)
54. Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**(4), 341–359 (1997)
55. Zhang, J., Sanderson, A.C.: JADE: adaptive differential evolution with optional external archive. IEEE Trans. Evol. Comput. **13**(5), 945–958 (2009)
56. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution—A Practical Approach to Global Optimization. Natural Computing Series. Springer, Berlin (2005)
57. Barbosa, H.J., Sá, A.: On adaptive operator probabilities in real coded genetic algorithms. In: XX International Conference of the Chilean Computer Science Society (2000)
58. Thierens, D.: An adaptive pursuit strategy for allocating operator probabilities. In: Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, pp. 1539–1546. ACM (2005)
59. Gong, W., Fialho, Á., Cai, Z., Li, H.: Adaptive strategy selection in differential evolution for numerical optimization: an empirical study. Inf. Sci. **181**(24), 5364–5386 (2011)
60. Liu, J., Lampinen, J.: A fuzzy adaptive differential evolution algorithm. In: TENCON'02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, pp. 606–611. IEEE (2002)
61. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: Evolutionary Computation. The 2005 IEEE Congress on 2005, pp. 1785–1791. IEEE (2005)
62. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans. Evol. Comput. **10**(6), 646–657 (2006)
63. Regis, R.G., Shoemaker, C.A.: Improved strategies for radial basis function methods for global optimization. J. Glob. Optim. **37**(1), 113–135 (2007)
64. Regis, R.G., Shoemaker, C.A.: Constrained global optimization of expensive black box functions using radial basis functions. J. Glob. Optim. **31**(1), 153–171 (2005)
65. Holmström, K.: An adaptive radial basis algorithm (ARBF) for expensive black-box global optimization. J. Glob. Optim. **41**(3), 447–464 (2008)
66. Regis, R.G.: Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. Comput. Oper. Res. **38**(5), 837–853 (2011)
67. Regis, R.G., Shoemaker, C.A.: Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. Eng. Optim. **45**(5), 529–555 (2013)
68. Liang, J., Qu, B., Suganthan, P., Hernández-Díaz, A.G.: Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. In: Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report 201212 (2013)
69. Awad, N., Ali, M., Liang, J., Qu, B., Suganthan, P.: Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization (2016)
70. Regis, R.G.: An initialization strategy for high-dimensional surrogate-based expensive black-box optimization. In: Modeling and Optimization: Theory and Applications, pp. 51–85. Springer (2013)
71. Sun, C., Jin, Y., Cheng, R., Ding, J., Zeng, J.: Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. IEEE Trans. Evol. Comput. **21**(4), 644–660 (2017)