Multi-objective Infill Criterion Driven Gaussian Process Assisted Particle Swarm Optimization of High-dimensional Expensive Problems

Jie Tian, Ying Tan, Jianchao Zeng, Chaoli Sun, *Member, IEEE*, Yaochu Jin, *Fellow, IEEE*

***Abstract*—Model management plays an essential role in surrogate-assisted evolutionary optimization of expensive problems, since the strategy for selecting individuals for fitness evaluation using the real objective function has substantial influences on the final performance. Among many others, infill criterion driven Gaussian process assisted evolutionary algorithms have been demonstrated competitive for optimization of problems with up to 50 decision variables. In this paper, a multi-objective infill criterion that considers the approximated fitness and the approximation uncertainty as two objectives is proposed for a Gaussian process assisted social learning particle swarm optimization algorithm. The multi-objective infill criterion uses non-dominated sorting for model management, thereby avoiding combining the approximated fitness and the approximation uncertainty into a scalar function, which is shown to be particularly important for high-dimensional problems, where the estimated uncertainty becomes less reliable. Empirical studies on 50- and 100-dimensional benchmark problems and a synthetic problem constructed from four real-world optimization problems demonstrate that the proposed multi-objective infill criterion is more effective than existing scalar infill criteria for Gaussian process assisted optimization given a limited computational budget.**

***Index Terms*—Expensive optimization, multi-objective infill criterion, Gaussian process, social learning particle swarm optimization.**

I. Introduction

EVOLUTIONARY algorithms (EAs) typically assume that analytic objective functions exist for calculating the fitness functions. However, many real-world optimization problems, such as antenna design [1], power system design [2], trauma system design [3], blast furnace optimization [4], and aerodynamic wing design [5] usually involve computationally intensive numerical simulations or costly experiments to evaluate the performance of candidate solutions [6]. For example, high-fidelity crashworthiness analysis in automotive industry may take several days for one analysis and therefore more than ten years are needed to complete 1000 analyses [7]. Thus, EAs will be prohibited from being used to solve these computationally expensive optimization problems because a large number of fitness evaluations are typically required before they locate a sub-optimal solution. In recent years, fitness approximation techniques with the help of computationally efficient surrogate models (also called meta-models) have received increasing attention in evolutionary computation for solving expensive or data-driven optimization problems [3] [8] [9] [10] [11]. Commonly used surrogate models include the polynomial repression models [12] [13], Kriging models [14] [15] [16], which are also known as Gaussian processes [17] [18] [19] [20] [21] or Bayesian optimization [22], artificial neural networks [23] [24], radial basis functions [25] [26] [27] [28] [29], and support vector machines [30] [31].

The surrogate model and the model management, i.e., the strategy for selecting candidate individuals to be evaluated using the expensive real objective function, are critical for the success of surrogate-assisted evolutionary algorithms [8] [10]. In the present work, we adopt the Gaussian process (GP) model as the surrogate mainly for its capability of providing an estimate of the fitness approximation uncertainty together with the approximated fitness value. The benefits of using GP model have been shown empirically in many papers, readers can refer to [14] [32] [33] [34] [22]. However, the model management strategy for selecting individuals to be evaluated using the real objective function, which is often called infill criterion in Gaussian process assisted evolutionary algorithms, remains the most important issue to be studied [35] [36]. Generally speaking, there are two main criteria for determining which individuals are to be evaluated using the real objective function. One commonly used criterion is to evaluate the individuals that have the global or local best approximated fitness. For example, the best individual in each cluster (which can be seen as local best individual) is suggested to be evaluated using the real objective function in [37] [38]. In [28] and [39], it is proposed to evaluate both the global and personal best particles. The second widely used criterion is

Manuscript received September 22, 2017; revised June 26, 2018; accepted September 04, 2018.This work was supported in part by the National Natural Science Foundation of China under the grant Nos. 61472269, 61403272 and 61403271, the Fund Program for the Scientific Activities of Selected Returned Overseas Professionals in Shanxi Province, and the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, China. (*Corresponding authors: Yaochu Jin and Chaoli Sun*)

Jie Tian is with the College of Mechanical Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China. She is also with School of Data and Computer Science, Shandong Women's University, Jinan, 250300 China. E-mail: tianjie1023@outlook.com.

Ying Tan is with the Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China. E-mail:tanying1965@gmail.com.

Jianchao Zeng is with the College of Mechanical Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China. He is also with the School of Computer Science and Control Engineering, North University of China, Taiyuan 030051, China. E-mail: zengjianchao@263.net.

Chaoli Sun is with the Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China. E-mail: chaoli.sun.cn@gmail.com.

Yaochu Jin is with the Department of Computer Science, University of Surrey, Guildford, GU2 7XH, UK. He is also with the Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China. E-mail: yaochu.jin@surrey.ac.uk.

to select the individual whose predicted fitness value has a large amount of uncertainty for fitness evaluation using the real fitness function. Selection of individuals whose approximated fitness has a large degree of uncertainty is out of the following two reasons. First, a large degree of uncertainty in fitness approximation of the individuals indicates that the fitness landscape around these individuals has not been well explored and therefore evaluation of these individuals are very likely to find a good solution [40]. Second, evaluating the individuals with a large degree of uncertainty can most effectively improve the accuracy of the surrogate model [10]. Many methods have been proposed for calculating the degree of uncertainty. For example, the degree of uncertainty is defined to be inversely proportional to the average distance between the individual and training samples which are used for constructing the surrogate [40]. In [37] [41] [42], authors proposed to use multiple surrogates or surrogate ensembles for fitness approximation so as to obtain a degree of uncertainty according to the different fitness values approximated by multiple models.

Several infill criteria have been proposed for model management when Gaussian processes are adopted for both single- and multi-objective surrogate-assisted optimization [35]. Commonly used infill criteria include the lower confidence bound (LCB) [15] [43], the expected improvement (EI) [44] [45] and the probability of improvement (PI) [26] [46]. Emmerich et al. [47] compared the above three criteria in surrogate-assisted optimization of low-dimensional (20-dimensional) problems[1], which showed that the EI criterion yields the best performance but requires the highest computing cost. In [45] and [48], EI is applied to surrogate-assisted multiobjective optimization and surrogate-assisted optimization of multiple problems, respectively. Liu et al [21] proposed to choose the individual with the smallest LCB value for fitness evaluation at each generation. Although the individual with the smallest LCB value might not be the best estimated solution, but it is a reasonably good solution that is able to balance the considerations of searching the promising region and less explored region.

In recent years, surrogate-assisted evolutionary algorithms have received increasing attention in solving expensive multi- and many- objective optimization problems. Namura et al. [49] proposed a new infill criterion, in which the EI of the Penalty-based Boundary Intersection (PBI) and the inverted PBI are utilized for selecting additional sample points to update the Kriging model. In [50], Zhan et al. suggested to aggregate the elements of the EI matrix into a scalar function. Chugh et al. [51] suggested a GP-assisted reference vector guided evolutionary algorithm for solving computationally expensive many-objective optimization, in which the individuals that have either the maximum amount of uncertainty or the minimum angle penalized distance are evaluated using the real objective functions. Wang et al. [52] proposed an ensemble-based model management strategy for surrogate-assisted evolutionary algorithm, in which both uncertainty and performance based infill criteria were utilized.

To the best of our knowledge, although both the approximated fitness and the uncertainty of the approximated fitness have been considered in model management, most of them either use one criterion or aggregate multiple criteria into a scalar one to select the individuals to be evaluated using the real computationally expensive objective function. In this paper, a multi-objective infill criterion, MIC for short, is proposed. Different from the existing infill criteria, the proposed MIC considers the minimization of the approximated fitness (for maximization problems, the maximize the fitness is, the better) and the maximization of the uncertainty of the fitness approximation as two separate objectives and a non-dominated sorting strategy is used to determine which individuals are to be chosen for fitness evaluation using the real expensive fitness function. The MIC has two main advantages for model management, particularly for optimization of high-dimensional problems. First, the proposed MIC sorts the population using non-dominated sorting and selects the completely dominated individuals (in terms of the approximated fitness and the degree of uncertainty) for fitness evaluation, thereby taking into account both exploration of the search space and enhancing the model quality. Second, MIC avoids pre-specifying a problem-specific hyper-parameter to linearly or nonlinearly combine the estimated fitness and the uncertainty, as done in LCB and EI, significantly enhancing the algorithm's ability to strike a good balance between performance and uncertainty. We will show that avoiding specifying the hyper-parameter becomes extremely important when the search dimension increases due to the fact that the estimated uncertainty becomes less reliable when only a small number of training samples is allowed to construct the Gaussian process model.

The rest of this paper is organized as follows. Section II provides a brief overview of the related techniques used in the proposed algorithm. A multi-objective infill criterion together with a Gaussian process assisted social learning particle swarm optimization is then presented in Section III. The proposed algorithm is empirically evaluated and compared with a few state-of-the-art algorithms on six 50- and 100-dimensional benchmark problems and a synthesized real-world problem in Section IV. Finally, Section V summarizes the paper and discusses future work.

II. Related Techniques

Without loss of generality, the optimization problem considered in this work can be formulated as follows:

$$\begin{aligned} &\min\ f(\mathbf{x}) \\ &\text{s. t. } \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^D$ is the feasible solution set, denotes the dimensionality of the search space, $f(\mathbf{x})$ is the objective function, $\mathbf{x}_l$ and $\mathbf{x}_u$ are the lower and upper bounds of the decision variables. In the following, we give a concise description of the Gaussian process and the social learning particle swarm optimization algorithm [53], which is adopted in this work as the base optimizer, as it was shown to perform robustly on medium to high-dimensional optimization problems [29] [53].

[1] In this paper, we call optimization problems with up to 30, from 30 to 50, and more than 50 decision variables, respectively, low-dimensional, medium-dimensional, and high-dimensional problems.

A. Gaussian process

The utilization of Gaussian process model can be traced back to 1970s, when it was used to solve regression problems in geostatistics, which was known as Kriging [54]. It became popular in 1990s, when Bayesian neural networks were investigated for machine learning [55]. As a kernel method developed on the basis of statistical learning and Bayesion theorem, theoretically sound methods are available for managing GP models for surrogate-assisted optimization [10], [22], [56]. Therefore, in this paper, the GP model, which can provide both fitness approximation uncertainty and the approximated fitness value as well, is employed as surrogate.

A Gaussian process is a collection of random variables that have a joint multivariate Gaussian distribution. Suppose DB is a training set that includes n samples $(\mathbf{x}_i, y_i), i = 1, 2, \ldots, n$. For any new input $\mathbf{x}$ (which corresponds to the decision variables of a candidate solution when the GP is used as a surrogate), its output y (which is the estimated fitness of the candidate solution) is a sample of $\mu + \varepsilon(\mathbf{x})$, where μ is the prediction of a regression model and $\varepsilon(\mathbf{x})$ is a normal distribution ($(N(0, \sigma^2)$) of zero mean and variance σ^2.

$$\mu{=}\mathbf{k}(\mathbf{x})\mathbf{K}^{-1}\mathbf{y} \tag{2}$$

$$\sigma^2{=}\kappa(\mathbf{x}) - \mathbf{k}(\mathbf{x})^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}) \tag{3}$$

where $\mathbf{K}$ is a matrix and each element in $\mathbf{K}$, $K_{ij} = C(\mathbf{x}_i, \mathbf{x}_j)$, describes the correlation between y_i and y_j related to the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. $\mathbf{k}(\mathbf{x}){=}[C(\mathbf{x}, \mathbf{x}_1), \cdots C(\mathbf{x}, \mathbf{x}_n)]^T$ is an n×1 vector of covariance between $\mathbf{x}$ and $\mathbf{X}$, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is the inputs of the training set, and $\kappa(\mathbf{x}){=}C(\mathbf{x}, \mathbf{x})$ is the covariance between $\mathbf{x}$ itself. The covariance function $C(\cdot,\cdot)$ can be any function that generates a positive semi-definite covariance matrix. According to the discussions and the empirical results comparing the performance and computational efficiency of various covariance functions provided in Section I (A) and Section II (A) of the Supplementary material, we adopt Matérn32 as the covariance function in this work, which is given as follows:

$$C(\mathbf{x}, \mathbf{x}_i) = k_{Matern32}(r) = \sigma_f^2(1 + \frac{\sqrt{3}r}{\sigma_l})exp(-\frac{\sqrt{3}r}{\sigma_l}) \tag{4}$$

where r is the Euclidean distance $r = \sqrt{(\mathbf{x} - \mathbf{x}_i)^T(\mathbf{x} - \mathbf{x}_i)}$, σ_l is the characteristic length scale, and σ_f is the standard deviation. Both σ_l and σ_f can be defined by a parameterization vector $\theta = (\theta_1, \theta_2)$, where $\theta_1 = \log \sigma_l$ and $\theta_2 = \log \sigma_f$. The standard deviation σ_f defines the differentiability of the covariance functions, and σ_l, which are characteristic length scales, describe the relevance between different inputs. To reduce the run time and memory complexity, we use sequential quadratic programming [57] to optimize the hyper-parameters, which iteratively estimates the Hessian matrix of the Lagrangian using the Broyden−Fletcher−Goldfarb−Shanno algorithm [58]. The reader is referred to Section I (B) of the Supplementary material for more discussions.

The prediction and uncertainty at $\mathbf{x}$ are expressed as below:

$$\hat{y}(\mathbf{x}) = \mu + \mathbf{k}(\mathbf{x})\mathbf{K}^{-1}(\mathbf{y} - \mathbf{I}\mu) \tag{5}$$

$$s^2(\mathbf{x}) = \sigma^2(1{+}\mathbf{k}^T(\mathbf{x})\mathbf{K}\mathbf{k}(\mathbf{x}) + \frac{\left(1 - \mathbf{I}^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x})\right)^2}{\mathbf{I}^T\mathbf{K}^{-1}\mathbf{I}}) \tag{6}$$

where $\mathbf{I}$ is an n × 1 vector with each element being equal to 1. s=$\sqrt{s^2(\mathbf{x})}$ is the estimated standard deviation (ESD) for measuring the uncertainty of the predicted fitness $\hat{y}(\mathbf{x})$.

B. Social Learning Particle Swarm Optimization

The social learning particle swarm optimization (SL-PSO) algorithm [53] was demonstrated to perform well in surrogate-assisted optimization of high-dimensional problems [29]. Thus, we use SL-PSO as the base optimizer for developing the multi-objective infill criterion driven Gaussian process assisted SL-PSO, called MGP-SLPSO. The main difference between SL-PSO and the canonical PSO is that during learning, SL-PSO does not learn from the global and personal best individuals. Instead, the particles in the swarm will be sorted according to their fitness in an ascending order and all particles, except for the best one, will learn from a randomly chosen better particle in the current swarm (termed demonstrators) starting from the worst particle. Specifically, the position of the j-th particle in SL-PSO at generation $t + 1$ is updated according to the following equations:

$$x_{jd}(t+1) = \begin{cases} x_{jd}(t) + \Delta x_{jd}(t+1) & \text{if } p_j(t) \le P_j^L \\ x_{jd}(t) & \text{otherwise} \end{cases} \tag{7}$$

$$\begin{aligned} \Delta x_{jd}(t+1) = r_1 \cdot \Delta x_{jd}(t) + r_2 \cdot (x_{kd}(t) - x_{jd}(t)) \\ + r_3 \cdot \varepsilon \cdot (\bar{x}_d(t) - x_{jd}(t)) \end{aligned} \tag{8}$$

In the above equations, $1 < j < m$, particle k is a randomly chosen demonstrator for particle j, so $j < \ k \ \le m$, m is the swarm size, and $x_{kd}(t)$ is the d-th element ($1 \le d \le D$, D denotes the dimension of the search space) of particle k. Note that the demonstrator should be re-chosen for each element of particle j. P_j^L is a learning probability inversely proportional to the fitness of particle j, $p_j(t)$ is a randomly generated probability for particle j, r_1, r_2 and r_3 are random numbers in the range of $[0, 1]$, $\bar{x}_d(t)$ denotes the mean position of the d-th variable in generation t, and constant ε, which is also known as the social influence factor, regulates the influence of $\bar{x}_d(t)$.

III. A Multi-objective Infill Criterion Driven Gaussian Process Assisted SL-PSO

A. Overall Framework of MGP-SLPSO

Fig. 1 shows the overall framework of MGP-SLPSO, which uses SL-PSO as the base optimizer and the GP model as the surrogate. The pseudo code for the main components of MGP-SLPSO is presented in Algorithm 1. The swarm consists of m particles (known as the swarm size), and the positions of the particles (i.e., the decision variables) of the initial swarm are generated using Latin hypercube sampling (LHS) [59]. All particles in the initial swarm are evaluated using the real objective function and the positions and their corresponding fitness values are stored in archive A. The position and velocity of all particles are then updated according to Eqs. (7) and (8) . Note that all new particles in the second iteration will also be evaluated using the real objective function and the generated data are stored in archive A. Starting from the third iteration, the data in archive A will be used to train the GP model, which is used to estimate the fitness and the ESD of all particles. Then, the MIC will be employed to select a few particles to be re-evaluated using the real objective function. These newly evaluated particles will be stored in archive A until the archive is full. When the number of training data is larger than the archive size, only the most recently evaluated data are stored.

From the above descriptions, we see that the main difference between MGP-SLPSO and SL-PSO is the model management strategy, including the selection of the particles for fitness evaluation using the real objective function (i.e., infill criterion), the update of the GP model, and the storage of the training data. Note that the best-so-far solution found by the algorithm is maintained and updated at the end of each iteration.

Although GP models have been successfully used as surrogates to assist algorithms for optimizing computationally expensive problems, they also suffer from serious limitations, in particular for high-dimensional problems. First, constructing a GP model may become computationally very intensive as the number of training data increases [51] [60]. Second, as indicated in [61] and also confirmed in this work, the difference between the ESDs of different solutions will vanish for high-dimensional problems, reducing the effectiveness of existing infill criteria. These issues can be partly alleviated by choosing the right covariance matrix and the right method for optimizing the hyper parameters in GP models [62].

This work takes a solid step forward by proposing a multi-objective infill criterion to address the issue of vanishing ESD information in GP-assisted evolutionary optimization of high-dimensional problems. In the following, we present in detail the proposed MIC for GP-assisted particle swarm optimization of high-dimensional systems.

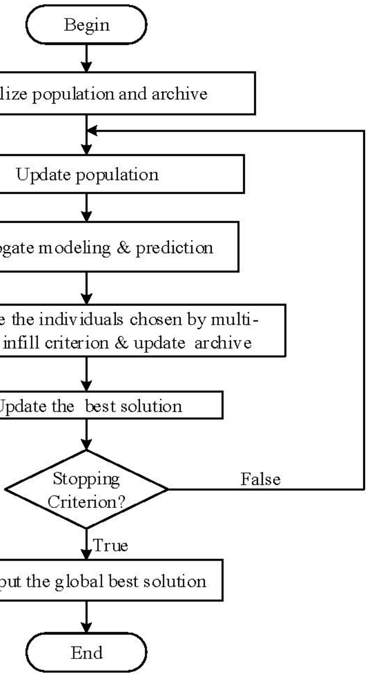
Begin
Intialize population and archive
Update population
Surrogate modeling & prediction
Re-evaluate the individuals chosen by multi-objective infill criterion & update archive
Update the best solution
Stopping Criterion?
False
True
Output the global best solution
End

Fig. 1. A diagram of the proposed MGP-SLPSO

Algorithm 1: The pseudo code of MGP-SLPSO

```
begin
    $t = 0$;
    Use LHS to generate the initial swarm;
    while the computational budget is not exhausted do
        if the amount of data in archive A does not exceed the pre-defined threshold then
            Evaluate the solutions using the real expensive objective function and update the global best position;
            Save all exact evaluations in $A$;
            $t = t + 1$;
            Update the particles according to (7) and (8);
        else
            Train a GP model using the latest n data in archive $A$;
            Update all particles according to (7) and (8);
            Estimate the fitness of all particles using GP model;
            Choose the particles according to the multi-objective infill criterion (see Algorithm 2) for fitness evaluation using the real objective function and update the global best position;
            Update archive $A$;
            $t = t + 1$;
        end if
    end while
end
```

B. Multi-objective infill criterion

Various infill criteria have been proposed for GP-assisted optimization algorithms [35], [36]. However, most infill criteria have only been applied to low-dimensional optimization problems, and little research has been reported on the effectiveness of the existing infill criteria optimization problems with more than 50 decision variables. Most existing infill criteria make use of the ESD that indicates the amount of uncertainty of the approximated fitness. These infill criteria have been shown to work well on low-dimensional problems; however, little research has been reported on the effectiveness of the infill criteria as the dimension of the optimization problems increases. One important phenomenon we have observed in our empirical studies is that the ESDs of different particles become extremely similar when the GP is used for approximating high-dimensional objective functions, in particular when the number of training samples is small.

Figs. 2 and 3 give examples of the ESDs of 100 particles in approximating the 10- and 50-dimesnional Rosenbrock functions, respectively. In each figure, we present the ESD of each particle at iterations 3, 10, 20, and 70 with different sizes of training data (denoted by τ). From Fig. 2, we can find that at iteration 3, i.e.,t=3, all ESD values are large but nearly the same, making it hardly possible to distinguish the degrees of uncertainty of the predicted fitness of different particles. As the amount of available training data increases over the

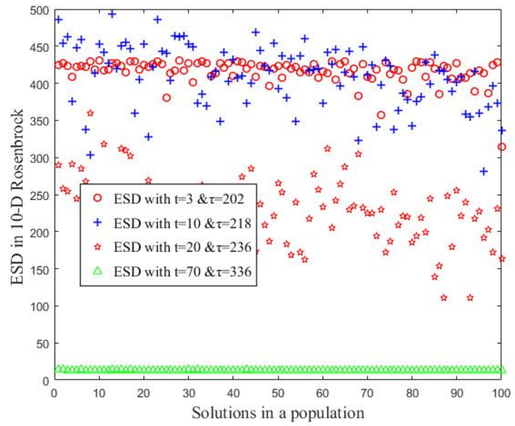

Fig. 2. ESD for the 10-dimensional Rosenbrock function

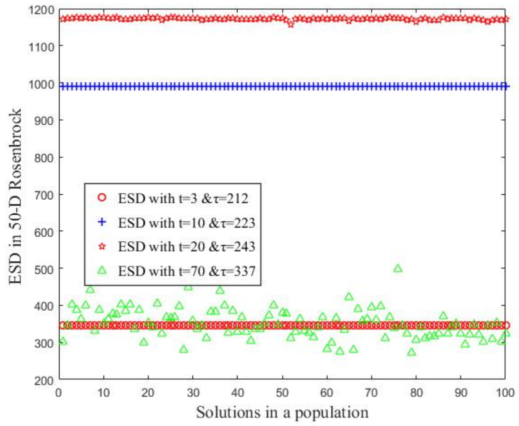

Fig. 3. ESD for the 50-dimensional Rosenbrock function

iterations (e.g., iterations 10 and 20), the ESDs become clearly distinguishable. But as the optimization converges (e.g. at iteration 70), the ESD values of different particles become indistinguishable again. The above issue becomes more serious for 50-dimensional problems, as shown in Fig. 3, where we can find that the ESD values of all particles are nearly the same at iterations 3, 10 and 20, and they can become distinguishable only around iteration 70. The above empirical results can be explained by analyzing Eq.(6), from which we can see that the ESD of a new individual $\mathbf{x}$ is determined by the correlation between $\mathbf{x}$ and the training samples, which is eventually determined by the covariance function and the distance between $\mathbf{x}$ and the training samples. As a result, all candidate solutions in a new population will have extremely similar ESDs, as all solutions may be equally far from the small number of training samples in a high-dimensional decision space.

From the above analysis, we recognize that for high-dimensional problems, the ESD values of different solutions will become hardly distinguishable, making those infill criteria based on linear or nonlinear combinations of the approximated fitness and the uncertainty of the approximated fitness (i.e., the ESD) less effective for selecting the right individuals for evaluation using the expensive objective function.

To address the above issue, we propose a new infill criterion, called multi-objective infill criterion (MIC) for managing the GP model, which considers the approximated fitness and the approximation uncertainty as two separate objectives. Consequently, non-dominated sorting is utilized to determine which individuals are to be evaluated using the real objective functions, rather than a scalar value that linearly or nonlinearly aggregates the approximated fitness and approximation uncertainty.

For a minimization problem, we provide below a mathematical description of the proposed MIC, which can be formulated by:

$$\begin{aligned} &\min_{\mathbf{x}} \ \mathbf{g}(\mathbf{x}) = (\mathbf{g}_1(\mathbf{x}), \mathbf{g}_2(\mathbf{x})) \\ &\text{s. t. } \mathbf{x} \in \mathbb{S} \subset \mathbb{R}^D \end{aligned} \tag{9}$$

$$\mathbf{g_1}(\mathbf{x}) = \hat{y}(\mathbf{x}) \tag{10}$$

$$\mathbf{g_2}(\mathbf{x}) = \sqrt{s^2(\mathbf{x})} \tag{11}$$

where the search space $\mathbb{S}$ is defined as $\mathbb{S} := [\mathbf{x}_{min}, \mathbf{x}_{max}]$, $\mathbf{x}_{min}$ and $\mathbf{x}_{max}$ are user-defined lower and upper bounds, $\hat{y}(\mathbf{x})$ is the approximated fitness according to (5) and $\sqrt{s^2(\mathbf{x})}$ is ESD according to (6) for measuring the estimated uncertainty.

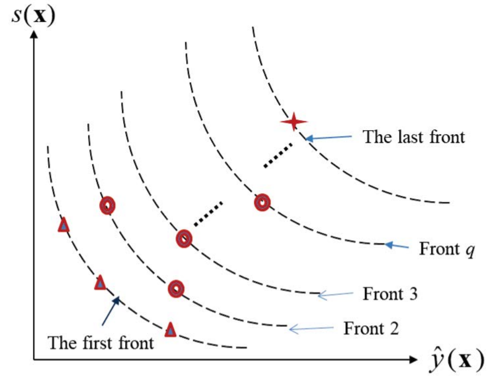

Fig. 4. An illustration of the non-dominated fronts sorted according to the approximated fitness and the estimated uncertainty. The fitness value of the articles on both the first and last non-dominated fronts will be evaluated using the expensive objective function.

In MGP-SLPSO, once the fitness value of all particles in the present generation is estimated using the trained GP model, the fast non-dominated sorting algorithm presented in [63] is employed to sort the particles into different non-dominated fronts. In the proposed MIC, the particles on the first front consisting of the non-dominated particles in the swarm will be evaluated using the expensive objective function. Then particles on the last front composed of those that do not dominate any others will also be selected for fitness evaluation, thereby emphasizing exploration. Fig.4 gives an illustration of selected particles on the first and last fronts that will be evaluated using the real objective function.

Algorithm 2 lists the pseudo code of MIC. All particles that are evaluated using the real expensive objective function are then stored in archive A.

Algorithm 2: Multi-objective infill criterion (MIC)

```
Input  : the approximated fitness value $\hat{y}$ and the
         approximation uncertainty $s$ of all particles.
Output: the particles to be evaluated using the real
         fitness function.
begin
    Non-dominated sorting: using the fast non-dominated
      sorting approach to sort the swarm into different
      non-dominated fronts according to $\hat{y}$ and $s$;
    Evaluate the particles both on the first non-dominated
      front and the last non-dominated front;
    Store all exact evaluations in archive $A$;
end
```

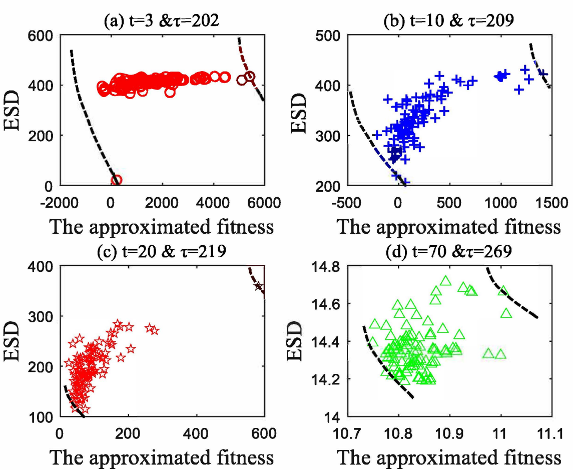

Fig. 5. The approximated values and ESD for 10-dimensional Rosenbrock function

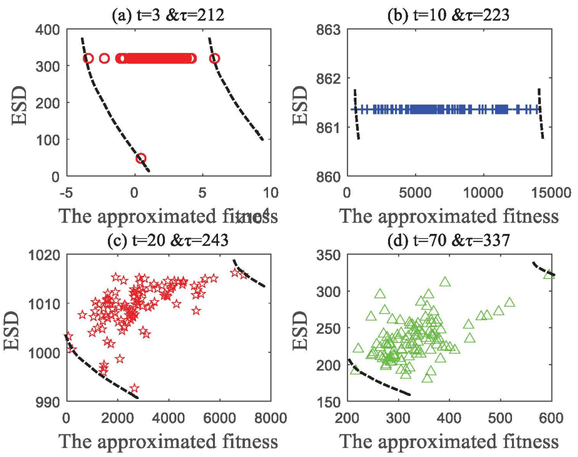

Fig. 6. The approximated values and ESD for 50-dimensional Rosenbrock function

To provide a first flavor about the difference between the proposed MIC and existing infill criteria, we present the results of the approximated fitness and ESD of each individual at iterations 3, 10, 20 and 70 in Figs. 5 and 6, respectively, when solving the 10- and 50-dimensional Rosenbrock function. From Figs. 5 and 6, we can see that compared from the existing scalar infill criteria, MIC is able to better distinguish the solutions under different situations, thereby making it easier to select individuals for fitness evaluation using the real objective function. Even in special cases when the ESD of different solutions is same, as shown in Figs. 6(a) and 6(b), different solutions can be easily distinguished by using the approximated fitness.

C. Empirical studies on the multi-objective infill criterion

To better understand why the proposed MIC is beneficial, we first examine the influence of the solutions in the first and last fronts on the population diversity. To this end, we compare three infill criteria in terms of the resulting population diversity in the decision space using a diversity metric called pure diversity (PD) proposed in [64] in optimizing the 100-D Rosenbrock and Ackley functions. PD is adopted here to assess the population diversity is mainly because PD is able to more accurately reflect diversity in high-dimensional spaces [64]. In the PD metric, an Lp-norm-based ($p < 1$) distance is adopted to measure the dissimilarity of solutions. The higher the PD value, the more diverse the population is. The results averaged over 30 independent runs are provided in Fig. 7, from which we can see that the diversity of the population using the multi-objective infill criterion (denoted First front + Last front) remains to be the largest compared to the EI and the one selecting the individuals in the first front only. This implies that MIC encourages stronger explorative search for high-dimensional problems.

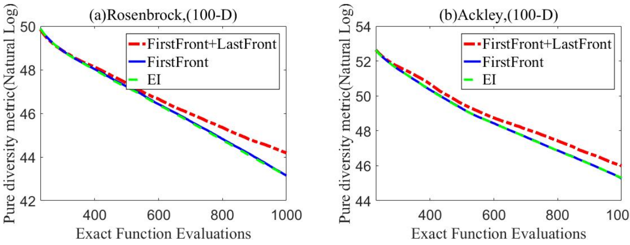

Fig. 7. Pure diversity metric on 100-D Rosenbrock and Ackley functions

In the following, we examine the number of solutions selected by MIC and their contribution to the improvement of the best fitness during the optimization. Fig. 8 plots number of solutions in the first and last fronts, respectively. As we can see, MIC often selects more individuals in the first front than the last, and it selects more than one individual in most iterations. Recall that existing single-objective infill criteria usually select only one individual and selecting multiple solutions for fitness re-evaluation with the real expensive objective function may be beneficial, as can also be seen from Fig. 6.

To investigate the contributions made by the solutions selected from the first and last fronts in MIC, we calculate the

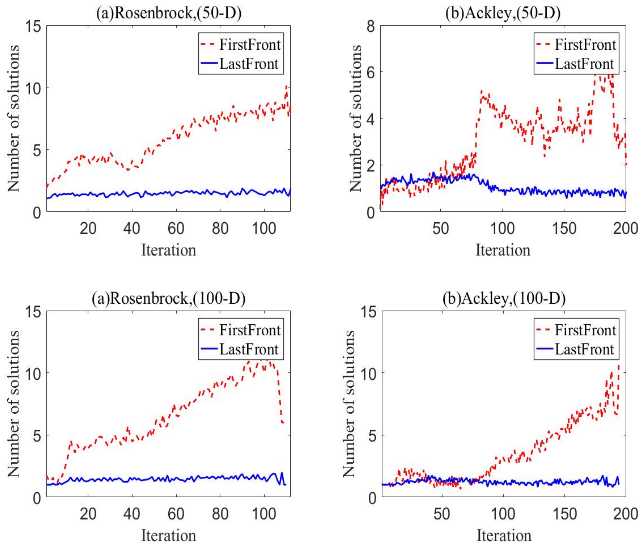

Fig. 8. Number of solutions in the first and last fronts over the generations on 50- and 100-dimensional Rosenbrock and Ackley functions.

percentage of the solutions in the first and last fronts in MGP-SLPSO that have contributed to the improvement of the global best fitness. Table I lists the average total number of solutions in the first and last fronts versus the number of solutions in the first and last fronts that have successfully improved the fitness in optimizing the 50-D and 100-D Rosenbrock and Ackley functions. From Table I, we can find that the percentage of the solutions that contribute to the successful improvement of the global best fitness, either on the first or last front, increases as the dimension of the problem becomes higher. In particular, it can be clearly seen that the solutions on the last front play an increasingly important role in improving the fitness of higher dimensional problems, confirming that the proposed MIC is effective for surrogate-assisted optimization of high-dimensional problems.

The empirical studies and analyses help understand the motivation and benefit of the proposed MIC compared against existing ones that linearly or nonlinearly aggregate the predicted fitness and the ESD. In the following, we verify the advantage of MIC when it is adopted in GP-assisted SLPSO.

IV. Experimental Studies

To examine the effectiveness of the proposed infill criterion on high-dimensional expensive optimization problems, we conduct a set of empirical studies on six 50- and 100-dimensional benchmark problems and a synthetic real-world problem by comparing it with some state-of-the-art surrogate-assisted evolutionary algorithms [29]. A description of the benchmark problems are listed in Table II.

As mentioned above, most GP-assisted optimization algorithms have been shown promising mainly on the low-dimension problems, typically less than 15 decision variables [51] and not higher than 30 [15]. Most recently, Sun et al. [29] proposed a surrogate-assisted cooperative swarm optimization algorithm (called SA-COSO) for solving high-dimensional optimization problems, in which a fitness estimation strategy assisted particle swarm optimization algorithm cooperates with an radial-basis-function network assisted SL-PSO algorithm. It was shown that the SA-COSO algorithm can obtain better results than GPEME (GP+DR) proposed in [15] on 50-dimensional problems and that it is successful in locating promising solutions on 100-dimensional problems given a limited computational budget. In our comparative studies, the performance of the proposed MGP-SLPSO algorithm is compared with SL-PSO (without surrogates), and three variants of GP-assisted SL-PSO driven by different infill criteria, namely, GP-assisted SL-PSO based on the approximated fitness only (GP-Fit for short), the lower confidence bound (GP-LCB), and the expected improvement (GP-EI) in order to investigate whether the proposed MIC has any advantage over the scalar infill criteria. All settings of GP-Fit, GP-LCB and GP-EI are the same as MGP-SLPSO, with the infill criterion being the only difference. In GP-Fit, the particles in the current population are sorted according to their approximated fitness. The best individual is then chosen to be evaluated using the real objective function. In addition, MGP-SLPSO is compared with SA-COSO algorithm [29] on the six benchmark problems. All experiments are implemented on a computer with a 2.10GHz processor and 32GB in RAM. Experimental results are obtained over 30 independent runs in Matlab $^{®}$R2017A.

A. *Parameter Settings*

The parameters of the SL-PSO algorithm in MGP-SLPSO, GP-Fit, GP-LCB and GP-EI are set same as recommended in [53]. The maximum number of fitness evaluations is set to 1,000. The number of training data n in GP-Fit, GP-LCB GP-EI and MGP-SLPSO is defined to be $2*m \leq n \leq 4*m$, where m is the swarm size, which is determined by the dimension of the search space as $m = 100 + \left\lfloor \frac{D}{10} \right\rfloor$, $2*m$ is the minimum number of training data and $4*m$ is the maximum number of training data. If the number of data in archive is larger than $4*m$, only the most recent $4*m$ data will be used to train the GP model. The number of training data is set due to the following reasons. First, as the computational complexity of GP model is $O(n^3)$ [51], where n is the number of training data, the cost of training time will increase rapidly with the number of training data. Thus, it is necessary to limit the number of training data. Second, it was suggested that the minimum number of training data should be the double of the dimension [47], therefore, $2*m$ is set to be the lower bound of the training data size, which is slightly larger than $2*D$, D is the dimension of the search space as defined in Section II (B). Third, as discussed in Section III, The estimated uncertainty will be less reliable for high-dimensional systems, especially when the number of training data is limited.

B. *Experimental results on high-dimensional problems*

Tables III and IV present the statistical results of the compared algorithms, including the t-test results calculated at a significance level of $\alpha = 0.05$. All algorithms perform $2*m$ real fitness evaluations (FEs) to train the GP model before they start optimization and terminate once 1000 exact FEs are exhausted. In the tables, $+$ indicates that the proposed

TABLE I
AVERAGE NUMBER OF SOLUTIONS IN FIRST FRONT AND LAST FRONT VERSUS THE AVERAGE NUMBER OF SUCCESSFUL SOLUTIONS IN FIRST FRONT AND LAST FRONT ON ROSENBROCK AND ACKLEY FUNCTIONS

Dimention	Function	Front	Average no. of solutions	Average no. of successful solutions	Percentage of the no. of successful solutions
50	Rosenbrock	First Front	635	28	4.41%
		Last Front	160	0	0.00%
	Ackley	First Front	580	47	8.10%
		Last Front	216	24	11.11%
100	Rosenbrock	First Front	630	40	6.35%
		Last Front	150	16	10.67%
	Ackley	First Front	449	45	10.02%
		Last Front	331	58	17.52%

TABLE II
DESCRIPTION OF THE SIX BENCHMARK FUNCTIONS

Function No.	**Function name**	**No. of variables**	**Global optimum**	**Property**
F1	Ellipsoid	50/100	0	Unimodal
F2	Rosenbrock	50/100	0	Multimodal with narrow valley
F3	Ackley	50/100	0	Multimodal,Local optimas number is huge
F4	Griewank	50/100	0	Multimodal,Local optimas number is huge
F5	Shifted Rotated Rastrign	50/100	-330	Very complicated multimodal
F6	Rotated Hybrid Composition Function(F19 in [41])	50/100	10	Very complicated multimodal

MGP-SLPSO algorithm, statistically significantly outperforms a compared algorithm, while $\approx$ and $-$ indicates that MGP-SLPSO performs comparably or significantly worse than the compared algorithm, respectively. In addition, the best results obtained for each function are highlighted in bold. From Table III and Table IV, we can find that all GP-assisted SL-PSO algorithms obtained better results than the SL-PSO without GP assistance, which confirms that the surrogate indeed helps accelerate the convergence of the SL-PSO. Compared to GP-Fit and GP-EI, MGP-SLPSO obtained better results on all 50- and 100-dimensional benchmark problems except for F2 with 50 dimensions. Note, however, that the results of MGP-SLPSO on F2 with 100 dimension are much better than those of GP-LCB and GP-Fit. These results showed that the proposed MIC is much more effective than the scalar infill criteria for high-dimensional problems. We see that MGP-SLPSO also outperforms SA-COSO on six benchmark functions.

To further demonstrate the competitive performance of MGP-SLPSO, the convergence profiles of the compared algorithms are plotted in Figs. 9∼20. From these figures, we can see that MGP-SLPSO is better than the compared algorithms with several orders of magnitude on the Ellipsoid and Griewank functions given 1000 FEs. When the allowed computational budget is reduced to 400-600 FEs, the performance of MGP-SLPSO remains to be clearly superior to the compared infill criterion driven GP assisted SL-PSO algorithms, and is comparable to that of the SA-COSO on 100-dimensional F6. Taking a closer look at the convergence profiles of GP-Fit, GP-EI and GP-LCB, we can see that they are similar on most benchmark problems, especially when the

TABLE III
COMPARATIVE RESULTS ON $50 - D$ BENCHMARK FUNCTIONS

	Approach	Best	Worst	Mean	Std.	
F1	SL-PSO	1.10E+03	2.09E+03	1.78E+03	2.74E+02	+
	SA-COSO	2.38E+01	8.38E+01	4.93E+01	1.60E+01	+
	GP-Fit	8.50E-08	1.03E+01	7.71E-01	2.33E+00	+
	GP-EI	1.76E-07	4.93E+00	4.01E-01	9.80E-01	+
	GP-LCB	2.10E-07	4.89E+00	4.34E-01	1.00E+00	+
	MGP-SLPSO	**2.90E-16**	**4.94E-15**	**9.88E-16**	**1.02E-15**	
F2	SL-PSO	1.82E+03	3.20E+03	2.37E+03	4.02E+02	+
	SA-COSO	1.54E+02	3.64E+02	2.49E+02	5.43E+01	+
	GP-Fit	6.22E+01	2.28E+02	1.07E+02	3.40E+01	$-$
	GP-EI	**5.82E+01**	2.19E+02	1.21E+02	3.66E+01	$\approx$
	GP-LCB	6.48E+01	1.82E+02	**1.05E+02**	3.30E+01	$-$
	MGP-SLPSO	8.84E+01	**1.65E+02**	1.20E+02	**1.87E+01**	
F3	SL-PSO	1.68E+01	1.86E+01	1.78E+01	4.26E-01	+
	SA-COSO	7.79E+00	1.26E+01	9.54E+00	1.21E+00	+
	GP-Fit	6.09E+00	1.43E+01	1.10E+01	2.53E+00	+
	GP-EI	6.66E+00	1.36E+01	9.95E+00	2.03E+00	+
	GP-LCB	6.62E+00	1.66E+01	1.04E+01	2.74E+00	+
	MGP-SLPSO	**7.77E+00**	**1.21E+01**	**9.31E+00**	**1.13E+00**	
F4	SL-PSO	2.16E+02	3.87E+02	2.86E+02	4.04E+01	+
	SA-COSO	3.69E+00	7.61E+00	5.54E+00	1.04E+00	+
	GP-Fit	2.22E-01	1.38E+00	7.56E-01	3.03E-01	+
	GP-EI	1.16E-01	1.68E+00	7.42E-01	3.36E-01	+
	GP-LCB	2.16E-01	2.37E+00	6.48E-01	4.38E-01	+
	MGP-SLPSO	**3.74E-02**	**6.14E-01**	**1.54E-01**	**1.30E-01**	
F5	SL-PSO	2.86E+02	5.03E+02	4.09E+02	5.20E+01	+
	SA-COSO	1.48E+02	2.95E+02	2.14E+02	3.33E+01	+
	GP-Fit	-1.11E+01	2.66E+02	1.03E+02	7.90E+01	+
	GP-EI	-6.37E+01	1.54E+02	5.62E+01	5.38E+01	+
	GP-LCB	**-9.80E+01**	1.91E+02	4.67E+01	5.52E+01	+
	MGP-SLPSO	-4.34E+01	**8.84E+01**	**3.30E+01**	**3.61E+01**	
F6	SL-PSO	1.13E+03	1.26E+03	1.20E+03	2.79E+01	+
	SA-COSO	**1.00E+03**	1.16E+03	1.08E+03	3.66E+01	+
	GP-Fit	1.09E+03	1.28E+03	1.20E+03	5.12E+01	+
	GP-EI	1.09E+03	1.21E+03	1.15E+03	3.48E+01	+
	GP-LCB	1.08E+03	1.22E+03	1.13E+03	3.55E+01	+
	MGP-SLPSO	1.03E+03	**1.11E+03**	**1.06E+03**	2.14E+01	

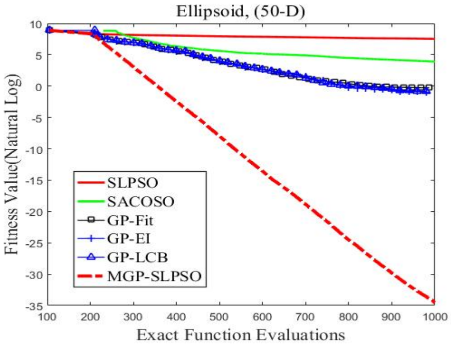

Fig. 9. The convergence profiles on 50-dimensional F1

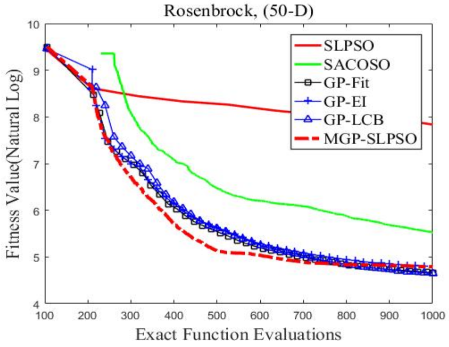

Fig. 10. The convergence profiles on 50-dimensional F2

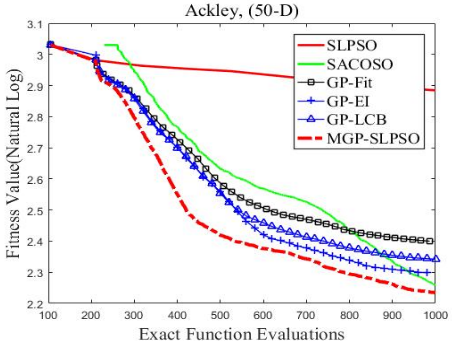

Fig. 11. The convergence profiles on 50-dimensional F3

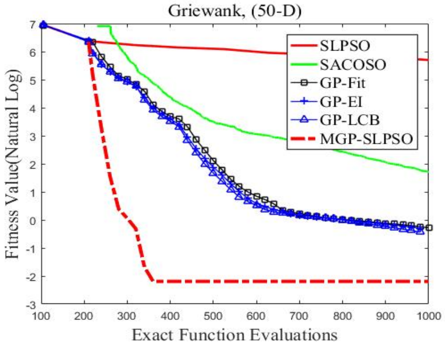

Fig. 12. The convergence profiles on 50-dimensional F4

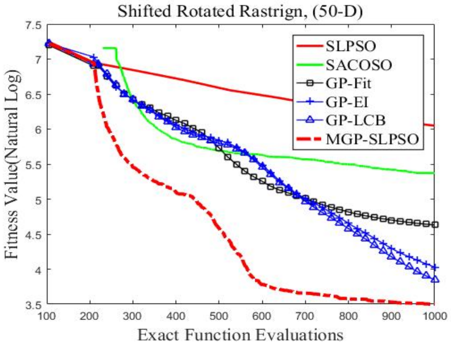

Fig. 13. The convergence profiles on 50-dimensional F5

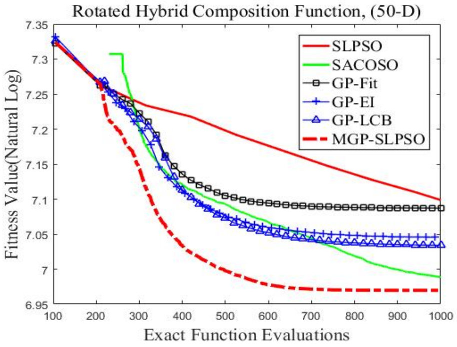

Fig. 14. The convergence profiles on 50-dimensional F6

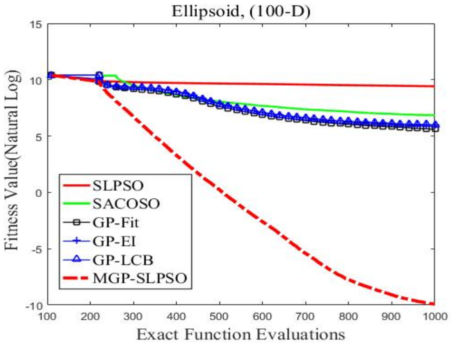

Fig. 15. The convergence profiles on 100-dimensional F1

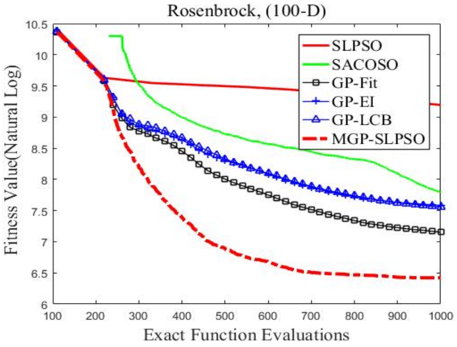

Fig. 16. The convergence profiles on 100-dimensional F2

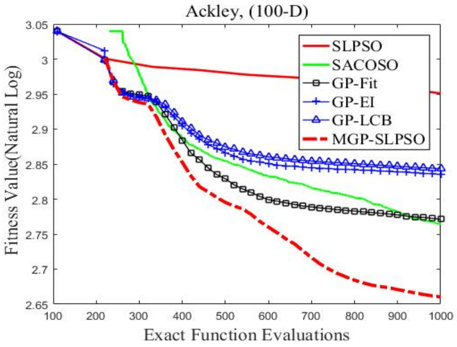

Fig. 17. The convergence profiles on 100-dimensional F3

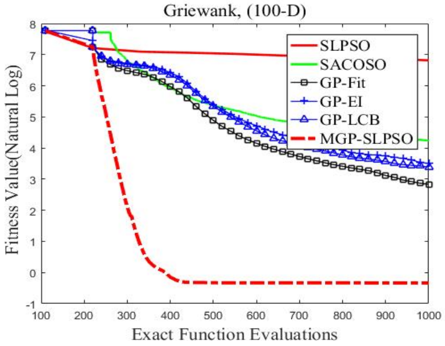

Fig. 18. The convergence profiles on 100-dimensional F4

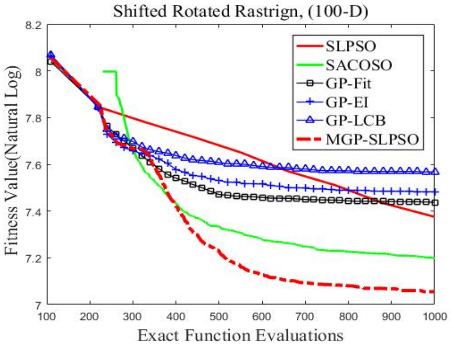

Fig. 19. The convergence profiles on 100-dimensional F5

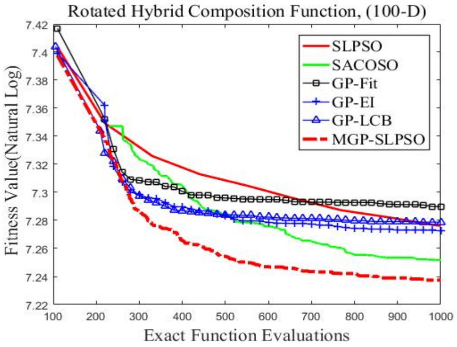

Fig. 20. The convergence profiles on 100-dimensional F6

TABLE IV
COMPARATIVE RESULTS ON $100 - D$ BENCHMARK FUNCTIONS

	Approach	Best	Worst	Mean	Std.	
F1	SL-PSO	9.00E+03	1.32E+04	1.17E+04	9.60E+02	+
	SA-COSO	5.94E+02	1.39E+03	9.29E+02	2.36E+02	+
	GP-Fit	5.46E+01	4.77E+02	2.72E+02	1.21E+02	+
	GP-EI	1.48E+02	1.02E+03	3.91E+02	2.15E+02	+
	GP-LCB	5.55E+01	1.46E+03	4.03E+02	3.06E+02	+
	MGP-SLPSO	**1.37E-16**	**1.03E-03**	**4.93E-05**	**1.95E-04**	
F2	SL-PSO	6.61E+03	1.10E+04	9.16E+03	1.12E+03	+
	SA-COSO	1.46E+03	4.63E+03	2.41E+03	7.99E+02	+
	GP-Fit	6.77E+02	2.42E+03	1.28E+03	3.81E+02	+
	GP-EI	1.17E+03	4.12E+03	1.96E+03	6.48E+02	+
	GP-LCB	1.20E+03	3.08E+03	1.92E+03	4.66E+02	+
	MGP-SLPSO	**4.55E+02**	**7.33E+02**	**6.12E+02**	**6.79E+01**	
F3	SL-PSO	1.85E+01	1.95E+01	1.90E+01	**2.43E-01**	+
	SA-COSO	1.42E+01	1.72E+01	1.59E+01	7.44E-01	+
	GP-Fit	1.44E+01	1.72E+01	1.60E+01	7.74E-01	+
	GP-EI	1.56E+01	1.84E+01	1.70E+01	7.62E-01	+
	GP-LCB	1.59E+01	1.88E+01	1.72E+01	6.25E-01	+
	MGP-SLPSO	**1.34E+01**	**1.57E+01**	**1.43E+01**	6.21E-01	
F4	SL-PSO	7.10E+02	1.05E+03	8.74E+02	8.76E+01	+
	SA-COSO	4.14E+01	1.06E+02	6.90E+01	1.50E+01	+
	GP-Fit	1.93E+00	5.01E+01	1.70E+01	1.26E+01	+
	GP-EI	3.43E+00	6.53E+01	3.34E+01	1.51E+01	+
	GP-LCB	5.05E+00	6.74E+01	2.97E+01	1.47E+01	+
	MGP-SLPSO	**4.78E-01**	**8.47E-01**	**7.15E-01**	**7.24E-01**	
F5	SL-PSO	1.37E+03	1.65E+03	1.52E+03	**8.53E+01**	+
	SA-COSO	1.10E+03	1.60E+03	1.34E+03	1.13E+02	**+**
	GP-Fit	1.44E+03	2.09E+03	1.70E+03	1.44E+02	+
	GP-EI	1.44E+03	2.08E+03	1.78E+03	1.66E+02	+
	GP-LCB	1.63E+03	2.38E+03	1.93E+03	1.96E+02	+
	MGP-SLPSO	**8.77E+02**	**1.16E+03**	**8.85E+02**	1.17E+03	
F6	SL-PSO	1.39E+03	**1.49E+03**	1.44E+03	**2.52E+01**	+
	SA-COSO	1.35E+03	1.52E+03	1.41E+03	3.80E+01	+
	GP-Fit	1.40E+03	1.56E+03	1.47E+03	4.14E+01	+
	GP-EI	1.39E+03	1.50E+03	1.44E+03	2.98E+01	+
	GP-LCB	1.39E+03	1.50E+03	1.45E+03	2.63E+01	+
	MGP-SLPSO	**1.33E+03**	**1.49E+03**	**1.39E+03**	4.77E+01	

dimension is high. This agrees with our observations in Section III that the ESD values of different particles in the swarm are too similar to distinguish for selecting the most uncertain particles to be evaluated using the real objective function.

To demonstrate the influence of the variance functions, we performed additional experiments comparing the performance of MGP-SLPSO using different covariance functions. The results are presented in Fig. S2 and Table S-II in the Supplementary material. We found that Matérn32 shows better performance on F2, F3 and F6, and similar performance on F4 and F5.

To further demonstrate the benefit of using a multi-objective infill criterion, we also show the results when the two objectives in Eq. (9), the estimated fitness and ESD, are replaced with two scalar infill criteria, such as fitness and LCB (denoted by Fit+LCB), or fitness and EI (denoted by Fit+LCB). The results on 100-dimensional functions are provided in Table S-III and Fig. S3 in the Supplementary materials. We can see that the combination of Fit+ESD (the proposed algorithm) converges faster than other combinations in the whole optimization process on F1, F2, F3 and F6.

C. Experimental results on 30-dimensional problems

To examine the performance of MGP-SLPSO on lower dimensional problems with limited computational budget, we also compare MGP-SLPSO with GPEME [15] and CAL-SAPSO [52] on 30-dimensional test problems. Table V presents the statistical results with 1000 FEs and Table VI shows the results compared to CAL-SAPSO with 330 FEs. From Table V, we can see that MGP-SLPSO obtains better results on all six problems, which indicates that MGP-SLPSO has better performance than GPEME on the lower-dimensional problems. We can also find from Table VI that when the number of FEs is reduced to 330, MGP-SLPSO remains to perform better on all benchmark problems except for F2 (Rosenbrock function), which has a very narrow valley from local optimum to the global optimum, which may pose challenges for MGP-SLPSO that uses a global GP model only. Note that a local search strategy is utilized in the CAL-SAPSO, which could have helped achieve better results on the Rosenbrock function.

TABLE V
COMPARATIVE RESULTS OBTAINED BY GPEME AND MGP-SLPSO ON $30D$ PROBLEMS WITH 1000 FES

	Approach	Mean	Std.
F1	GPEME	2.21E+02	8.16E+01
	MGP-SLPSO	**1.65E-22**	**4.19E-23**
F2	GPEME	2.58E+02	8.02E+02
	MGP-SLPSO	**1.00E+02**	**2.23E+01**
F3	GPEME	1.32E+01	1.58E+00
	MGP-SLPSO	**6.58E+00**	**2.60E+00**
F4	GPEME	3.66E+01	1.32E+01
	MGP-SLPSO	**1.30E-02**	**5.00E-03**
F5	GPEME	-2.19E+01	3.64E+01
	MGP-SLPSO	**-2.22E+02**	**1.96E+01**
F6	GPEME	9.59E+02	2.57E+01
	MGP-SLPSO	**9.52E+02**	**1.90E+01**

TABLE VI
COMPARATIVE RESULTS OBTAINED BY CAL-SAPSO AND MGP-SLPSO ON $30D$ PROBLEMS WITH 330 FES

	Approach	Mean	Std.
F1	CAL-SAPSO	4.02E+00	1.08E+00
	MGP-SLPSO	**4.17E-05**	**1.79E-05**
F2	CAL-SAPSO	**1.76E+00**	**3.96E-01**
	MGP-SLPSO	1.60E+02	2.40E+01
F3	CAL-SAPSO	1.62E+01	**4.13E-01**
	MGP-SLPSO	**9.55E+00**	2.92E+00
F4	CAL-SAPSO	9.95E-01	3.99E-02
	MGP-SLPSO	**1.50E-02**	**5.07E-03**
F5	CAL-SAPSO	2.49E +02	2.44E+01
	MGP-SLPSO	**-7.02E+01**	**1.65E+01**
F6	CAL-SAPSO	1.03E+03	4.27E+01
	MGP-SLPSO	**1.02E+03**	**3.01E+01**

D. Experimental results on a synthetic real-world problem

To further evaluate the effectiveness of MGP-SLPOS, we apply it to a complex 31-dimensional optimization problem [65], which was created by summing up the scaling outputs of four real-world functions, namely, Borehole, Wing Weight, OTL Circuit and Piston Simulation functions:

$$\begin{aligned} f(\mathbf{x}) &= y_1^*\left(x_1, \ldots, x_8\right) + y_2^*\left(x_9, \ldots, x_{18}\right) \\ &+ y_3^*\left(x_{19}, \ldots, x_{24}\right) + y_4^*\left(x_{25}, \ldots, x_{31}\right) \end{aligned} \tag{12}$$

where, $y_i^* = \frac{y_i - \min(y_i)}{\max(y_i) - min(y_i)}$, $i = 1,\ 2,\ 3,\ 4$, are the outputs from the Borehole, Wing Weight, OTL Circuit and Piston Simulation functions, respectively. The Borehole function models water flow rate (y_1) through a borehole in m^3 per year. The function of wing weight (y_2) is a mathematical model of light aircraft wings, which was used by Forrester et al. [66] to estimate the weight of a light aircraft wing. The OTL Circuit function [67] models the output of a transformerless push-pull circuit, which is the midpoint voltage (y_3). Finally, the Piston Simulation Function [67] models the circular motion of a piston within a cylinder (y_4), which is the cycle time (the time it takes to complete one cycle) in seconds. More details of the four real-world problems are Section II(C) of the Supplementary material. MGP-SLPSO, as well as other three different infill criteria, including GP-fit, GP-LCB and GP-EI are employed to optimize the synthetic real-world problem. Like the setting in [52] and several other references on surrogate-assisted evolutionary optimization, the computational budget is set to $11D$ FEs, where D is the search dimension. This means that the allowed maximum number of fitness evaluations is 11 times the search dimension. Each algorithm is run independently for 30 times. The convergence curves of the algorithms under comparison are given in Fig. 21. From the figure, we can see that MGP-SLPSO exhibits a clear advantage over the compared algorithms. Table VII lists the comparative results of the five GP-assisted SLPSO variants together with three traditional algorithms, including Interior Point Method (IP), Active Set Method (AS) and SQP. Again, MGP-SLPSO outperforms all compared algorithms, confirming its competitive performance on the synthetic real-world problem.

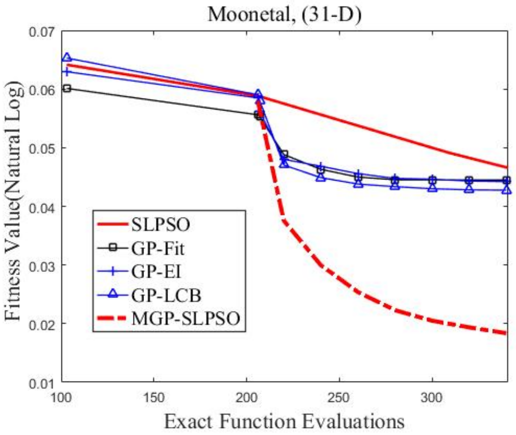

Fig. 21. The convergence profiles on the synthetic real-world problem

TABLE VII
RESULTS ON THE SYNTHETIC REAL-WORLD PROBLEM

Approach	Best	Mean	Std.
IP	1.0603E+00	1.0603E+00	2.2584E-16
SQP	1.0533E+00	1.0533E+00	4.5168E-16
AS	1.0500E+00	1.0500E+00	9.0336E-16
SL-PSO	1.0435E+00	1.0505E+00	4.5120E-03
GP-Fit	1.0313E+00	1.0455E+00	7.8564E-03
GP-EI	1.0317E+00	1.0452E+00	7.2391E-03
GP-LCB	1.0368E+00	1.0436E+00	5.3707E-03
MGP-SLPSO	**1.0166E+00**	**1.0185E+00**	**9.5653E-04**

E. Empirical Studies of the Computational Complexity

The computational complexity of MGP-SLPSO is determined by the computation time for fitness evaluations, for training the GP model, as well as for performing the non-dominated sorting in MIC. The computational complexity of training GP modeling is $O(n^3)$ [22], and the computational complexity of non-dominated sorting is $O(2m^2)$. In the following, we empirically examine the computation time required by the compared algorithms for solving 50-D and 100-D test problems.

Table VIII lists the average computation time that each algorithm has consumed. From Table VIII, we can see that SL-PSO needs the least computation time, which is understandable as its computation time is mainly consumed for fitness evaluations. For convenience, we use the time that the SL-PSO algorithm used on a fixed budget of 1000 FEs as the baseline for comparison. From Table VIII, we can see that the proposed MGP-SLPSO requires less time than the compared surrogate-assisted optimization algorithms, which indicates that the fast non-dominated sorting takes less time than training the surrogate model. Compared with most computationally intensive fitness evaluations in many real-world applications, where each fitness evaluation may take minutes to hours or even days, it is reasonable to consider that this amount of increase in computation time for training surrogates and sorting solutions is acceptable.

V. CONCLUSION AND FUTURE WORK

This paper proposes a multi-objective infill criterion for model management in Gaussian process assisted particle swarm optimization of high-dimensional problems. The main idea is to consider the approximated fitness and the approximation uncertainty of a GP surrogate as two separate objectives so that non-dominated sorting is employed to sort the swarm into a number of non-dominated fronts. The solutions in the first and the last non-dominated fronts are evaluated using the real objective function, thereby striking a good balance between the exploitation and exploration. Our analyses of the proposed multi-objective infill criterion indicate that it works more effectively than existing scalar infill criteria in selecting solutions to be evaluated using the expensive objective function, especially for high-dimensional problems. The benefit of the multi-objective infill criterion is further confirmed by comparing MGP-SLPSO with the state-of-the-art surrogate-assisted evolutionary algorithms on 30D to 100D benchmark problems and a synthetic real-world problem given a limited number of expensive fitness evaluations.

There is much room for improvement although MGP-SLPSO has shown promising performance on the six benchmark problems up to a dimension of 100. For example, further research is desirable to improve the performance of MGP-SLPSO on optimization problems whose fitness landscape has a narrow basin or a plateau around the global optimum. It is also essential to develop new algorithms that is able to train a GP model more efficiently given a limited amount of training data for high-dimensional problems. Therefore, in the

TABLE VIII
THE AVERAGE COMPUTATION TIME (IN SECONDS) USED BY THE ALGORITHMS UNDER COMPARISON FOR 50-D AND 100-D TEST INSTANCES USING 1000 FITNESS EVALUATIONS

Function	Approach	50D	100D
F1	SL-PSO	1.41E-02	2.19E-02
	SA-COSO	1.18E+03	1.66E+03
	GP-Fit	1.51E+02	2.48E+02
	GP-EI	2.36E+02	4.72E+02
	GP-LCB	1.53E+02	5.26E+02
	MGP-SLPSO	1.32E+02	2.22E+02
F2	SL-PSO	2.29E-02	4.53E-02
	SA-COSO	1.15E+03	1.72E+03
	GP-Fit	2.73E+03	2.86E+03
	GP-EI	3.92E+03	2.75E+03
	GP-LCB	5.85E+03	4.53E+03
	MGP-SLPSO	3.42E+02	3.68E+02
F3	SL-PSO	2.29E-02	3.49E-02
	SA-COSO	1.41E+03	1.92E+03
	GP-Fit	2.24E+03	2.63E+03
	GP-EI	2.24E+03	2.63E+03
	GP-LCB	2.24E+03	2.63E+03
	MGP-SLPSO	2.55E+02	2.33E+02
F4	SL-PSO	1.77E-02	3.85E-02
	SA-COSO	1.36E+03	1.51E+03
	GP-Fit	2.23E+02	2.49E+02
	GP-EI	2.38E+02	4.92E+02
	GP-LCB	2.42E+02	4.85E+02
	MGP-SLPSO	1.08E+02	1.60E+02
F5	SL-PSO	1.86E+01	1.77E+01
	SA-COSO	1.10E+03	1.55E+03
	GP-Fit	3.66E+03	5.04E+03
	GP-EI	4.72E+03	3.82E+03
	GP-LCB	5.03E+03	3.13E+03
	MGP-SLPSO	1.03E+03	8.12E+02
F6	SL-PSO	2.14E+01	2.49E+01
	SA-COSO	1.34E+03	2.13E+03
	GP-Fit	2.69E+03	2.97E+03
	GP-EI	5.19E+03	4.19E+03
	GP-LCB	2.92E+03	2.88E+03
	MGP-SLPSO	4.51E+02	2.65E+03

future, we will resort to more advanced techniques in machine learning, such as active learning and semi-supervise learning for dealing with high-dimensional problems. Finally, it is of great interest to extend the multi-criterion infill criterion to Gaussian process assisted multi-objective optimization.

REFERENCES

[1] S. Soltani and R. D. Murch, “A compact planar printed mimo antenna design,” *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 3, pp. 1–1, 2015.

[2] Y. Del Valle, G. K. Venayagamoorthy, S. Mohagheghi, and J. C. Hernandez, “Particle swarm optimization: Basic concepts, variants and applications in power systems,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171–195, 2008.

[3] H. Wang, Y. Jin, and J. O. Jansen, “Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 939–952, 2016.

[4] T. Chugh, N. Chakraborti, K. Sindhya, and Y. Jin, “A data-driven surrogate-assisted evolutionary algorithm applied to a many-objective blast furnace optimization problem,” *Materials and Manufacturing Processes*, vol. 32, no. 10, pp. 1172–1178, 2017.

[5] Y. S. Ong, P. B. Nair, and A. J. Keane, “Evolutionary optimization of computationally expensive problems via surrogate modeling,” *Aiaa Journal*, vol. 41, no. 4, pp. 687–696, 2003.

[6] S. Shan and G. G. Wang, “Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions,” *Structural and Multidisciplinary Optimization*, vol. 41, no. 2, pp. 219–241, 2010.

[7] L. Gu, “A comparison of polynomial based regression models in vehicle safety analysis,” in *ASME Design Engineering Technical Conferences*, vol. 33, 2001, Conference Proceedings, pp. 3444–3448.

[8] Y. Jin, “A comprehensive survey of fitness approximation in evolutionary computation,” *Soft computing*, vol. 9, no. 1, pp. 3–12, 2005.

[9] L. Shi and K. Rasheed, “A survey of fitness approximation methods applied in evolutionary algorithms,” *Computational intelligence in expensive optimization problems*, pp. 3–28, 2010.

[10] Y. Jin, “Surrogate-assisted evolutionary computation: Recent advances and future challenges,” *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.

[11] A. Kattan and Y.-S. Ong, “Surrogate genetic programming: A semantic aware evolutionary search,” *Information Sciences*, vol. 296, pp. 345–359, 2015.

[12] I. Paenke, J. Branke, and Y. Jin, “Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 405–420, 2006.

[13] Y. Lian and M. S. Liou, “Multiobjective optimization using coupled response surface model and evolutionary algorithm,” *European Radiology*, vol. 43, no. 6, pp. 1316–1325, 2012.

[14] L. Willmes, T. Back, J. Yaochu, and B. Sendhoff, “Comparing neural networks and kriging for fitness approximation in evolutionary optimization,” in *IEEE Congress on Evolutionary Computation*, 2003, Conference Proceedings, pp. 663–670.

[15] B. Liu, Q. Zhang, and G. G. E. Gielen, “A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180–192, 2014.

[16] J.-W. Klinkenberg, M. T. Emmerich, A. H. Deutz, O. M. Shir, and T. Bck, *A reduced-cost SMS-EMOA using kriging, self-adaptation, and parallelization*. Springer, 2010, pp. 301–311.

[17] M. A. El-Beltagy and A. J. Keane, “Evolutionary optimization for computationally expensive problems using gaussian processes,” in *International Conference on Artificial Intelligence*, vol. 1, 2001, Conference Proceedings, pp. 708–714.

[18] A. Ratle, “Kriging as a surrogate fitness landscape in evolutionary optimization,” *Artificial Intelligence for Engineering Design Analysis and Manufacturing*, vol. 15, no. 1, pp. 37–49, 2001.

[19] A. Kattan, A. Agapitos, Y.-S. Ong, A. A. Alghamedi, and M. ONeill, “Gp made faster with semantic surrogate modelling,” *Information Sciences*, vol. 355, pp. 169–185, 2016.

[20] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, “Expensive multiobjective optimization by moea/d with gaussian process model,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 456–474, 2010.

[21] B. Liu, H. Yang, and M. J. Lancaster, “Global optimization of microwave filters based on a surrogate model-assisted evolutionary algorithm,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 6, pp. 1976–1985, 2017.

[22] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. d. Freitas, “Taking the human out of the loop: A review of bayesian optimization,” *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[23] S. Ferrari and R. F. Stengel, “Smooth function approximation using neural networks,” *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 24–38, 2005.

[24] J. Yaochu, M. Olhofer, and B. Sendhoff, “A framework for evolutionary optimization with approximate fitness functions,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 481–494, 2002.

[25] R. G. Regis, “Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 326–347, 2014.

[26] Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum, “Combining global and local surrogate models to accelerate evolutionary optimization,” *IEEE Transactions on Systems Man and Cybernetics Part C*, vol. 37, no. 1, pp. 66–76, 2006.

[27] J. Gonzalez, I. Rojas, J. Ortega, H. Pomares, F. J. Fernandez, and A. F. Diaz, “Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation,” *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1478–1495, 2003.

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

[29] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 644–660, 2017.

[30] X. Lu, K. Tang, and X. Yao, "Classification-assisted differential evolution for computationally expensive problems," in *2011 IEEE Congress of Evolutionary Computation (CEC)*, 2011, Conference Proceedings, pp. 1986–1993.

[31] I. Loshchilov, M. Schoenauer, and M. Sebag, "Comparison-based optimizers need comparison-based surrogates," in *International Conference on Parallel Problem Solving From Nature*, 2010, Conference Proceedings, pp. 364–373.

[32] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. 3, pp. 235–284, 2008.

[33] C. E. Rasmussen and C. K. Williams, *Gaussian Process for Machine Learning*. MIT Press, 2006, vol. 1.

[34] C. K. I. Williams, "Prediction with gaussian processes: From linear regression to linear prediction and beyond," *Nato Asi*, vol. 89, pp. 599–621, 2015.

[35] D. Horn, T. Wagner, D. Biermann, C. Weihs, and B. Bischl, "Model-based multi-objective optimization: Taxonomy, multi-point proposal, toolbox and benchmark," in *Evolutionary Multi-Criterion Optimization: 8th International Conference, EMO*, 2015, Conference Proceedings, pp. 64–78.

[36] W. Ponweiser, T. Wagner, and M. Vincze, "Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models." in *IEEE Congress on Evolutionary Computation*, 2008, Conference Proceedings, pp. 3515–3522.

[37] Y. Jin and B. Sendhoff, "Reducing fitness evaluations using clustering techniques and neural network ensembles," in *Genetic and Evolutionary Computation*. Springer, 2004, Conference Proceedings, pp. 688–699.

[38] L. Grning, Y. Jin, and B. Sendhoff, "Individual-based management of meta-models for evolutionary optimization with application to three-dimensional blade optimization," *Evolutionary computation in dynamic and uncertain environments*, pp. 225–250, 2007.

[39] T. Jie, T. Ying, S. Chaoli, Z. Jianchao, and Y. Jin, "A self-adaptive similarity-based fitness approximation for evolutionary optimization," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016, Conference Proceedings, pp. 1–8.

[40] J. Branke and C. Schmidt, "Faster convergence by means of fitness estimation," *Soft Computing*, vol. 9, no. 1, pp. 13–20, 2005.

[41] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 329–355, 2010.

[42] T. Goel, R. T. Haftka, S. Wei, and N. V. Queipo, "Ensemble of surrogates," *Structural and Multidisciplinary Optimization*, vol. 33, no. 3, pp. 199–216, 2007.

[43] J. Dennis and V. Torczon, "Managing approximation models in optimization," *Multidisciplinary design optimization: State-of-the-art*, pp. 330–347, 1997.

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

[45] J. Luo, A. Gupta, Y. S. Ong, and Z. Wang, "Evolutionary optimization of expensive multiobjective problems with co-sub-pareto front gaussian process surrogates," *IEEE Transactions on Cybernetics*, pp. 1–14, 2018.

[46] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization*, vol. 21, no. 4, pp. 345–383, 2001.

[47] M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 421–439, 2006.

[48] A. T. W. Min, Y. S. Ong, A. Gupta, and C. K. Goh, "Multi-problem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2017.

[49] N. Namura, K. Shimoyama, and S. Obayashi, "Expected improvement of penalty-based boundary intersection for expensive multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 6, pp. 898–913, 2017.

[50] D. Zhan, Y. Cheng, and J. Liu, "Expected improvement matrix-based infill criteria for expensive multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, to be published. DOI:10.1109/TEVC.2017.2697503.

[51] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 129–142, 2018.

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

[53] R. Cheng and Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Information Sciences*, vol. 291, no. 6, pp. 43–60, 2015.

[54] M. Stein, *Statistical Interpolation of Spatial Data: Some Theory for Kriging*. New York: Springer, 1999.

[55] R. M. Neal, *Bayesian learning for neural networks*. Springer,, 1996.

[56] C. E. Rasmussen and C. K. Williams, "Gaussian processes in machine learning," *Lecture notes in computer science*, vol. 3176, pp. 63–71, 2004.

[57] J. Nocedal and S. J. Wright, *Numerical Optimization*. SPRINGER, 2006.

[58] J. J. E. Dennis and J. J. Mor, "Quasi-newton methods, motivation and theory," *SIAM Review*, vol. 19, no. 1, pp. 46–89, 1977. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1019005

[59] M. D. McKay, R. J. Beckman, and W. J. Conover, "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[60] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble based infill criterion for evolutionary multi-objective optimization of expensive problems," *IEEE Transactions on Cybernetics*, 2018(accepted).

[61] H. Wang, "Uncertainty in surrogate models," in *Genetic and Evolutionary Computation Conference Companion*, 2016, Conference Proceedings, pp. 1279–1279.

[62] J. Tian, Y. Tan, C. Sun, and J. Z. andY. Jin, "Comparisons of different kernels in kriging-assisted evolutionary expensive optimization," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, Conference Proceedings.

[63] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[64] H. Wang, Y. Jin, and X. Yao, "Diversity assessment in many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 6, pp. 1510–1522, 2017.

[65] H. Moon, A. M. Dean, and T. J. Santner, "Two-stage sensitivity-based group screening in computer experiments," *Technometrics*, vol. 54, no. 4, pp. 376–387, 2012.

[66] A. Forrester and A. Keane, *Engineering design via surrogate modelling: a practical guide*. John Wiley and Sons, 2008.

[67] E. Ben-Ari and D. Steinberg, "Modeling data from computer experiments: An empirical comparison of kriging with mars and projection pursuit regression," *Quality Engineering*, vol. 19, no. 4, pp. 327–338, 2007.

Jie Tian received the B.Sc. degree in Computer Science and Technology from Shandong Agricultural University, Taian, China in 2005, M.Sc. degree in Computer Software and Theory from Taiyuan University of Science and Technology, Taiyuan, China, in 2008, where she is currently pursuing the Ph.D. degree. She is also a Lecturer with Department of Computer Science and Technology, Taiyuan University of Science and Technology. Her current research interests include computational intelligence, surrogate-assisted evolutionary optimization, and engineering design optimization. She has served as a Reviewer for several journals, including the IEEE Transactions on Emerging Topics in Computational Intelligence, Soft Computing, and Complex & Intelligent Systems.

Ying Tan received the B.Sc. degree in Industrial Automation from Taiyuan University of Science and Technology in 1986, and the M.Sc. degree in Control Theory and Application from Tsinghua University in 1994. She is a Professor with the Department of Computer Science and Technology, Taiyuan University of Science and Technology. Her current research interests include intelligent computing, cloud computing and distributed network security.

Jianchao Zeng received the B.Sc. degree from Taiyuan Heavy Machinery Institute in 1982, the M.Sc. and Ph.D. degrees from Xian Jiaotong University in 1985 and 1990, respectively. He is a Professor with the School of Computer Science and Control Engineering, North University of China, Taiyuan, China. His current research focuses on modelling and control of complex systems, intelligent computation, swarm intelligence and swarm robotics. He has published more than 200 international journal and conference papers. Dr. Zeng is a Vice President of North University of China, and a Director of China Simulation Federation. He is also a Vice President of Technical Committee on System Simulation in Chinese Association of Automation.

Chaoli Sun (M'15) received the B.Sc. and M.Sc. degrees from Hohai University, Nanjing, China, in 2000 and 2003, respectively, and the Ph.D. degree from the Taiyuan University of Science and Technology, Taiyuan, China, in 2011.

She is a Professor with the Department of Computer Science and Technology, Taiyuan University of Science and Technology. Her current research interests include swarm intelligence, particle swarm optimization, surrogate-assisted evolutionary optimization, and their applications to practical engineering problems.

Prof. Sun is an Associate Editor of Soft Computing and Complex & Intelligent Systems, the Chair of the Task Force on Data-Driven Evolutionary Optimization of Expensive Problems within the Evolutionary Computation Technical Committee of the IEEE Computational Intelligence Society. She has also served as a Reviewer for journals, including the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, Information Sciences, Soft Computing, Natural Computing, and Complex & Intelligent Systems.

Yaochu Jin (M'98-SM'02-F'16) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996, respectively, and the Dr.-Ing. degree from Ruhr University Bochum, Germany, in 2001.

He is a Professor in Computational Intelligence, Department of Computer Science, University of Surrey, Guildford, U.K., where he heads the Nature Inspired Computing and Engineering Group. He is also a Finland Distinguished Professor funded by the Finnish Funding Agency for Innovation (Tekes) and a Changjiang Distinguished Visiting Professor appointed by the Ministry of Education, China. His research interests lie primarily in the cross-disciplinary areas of computational intelligence, computational neuroscience, and computational systems biology. He is also particularly interested in the application of nature-inspired algorithms to solving real-world optimization, learning and self-organization problems. He has (co)authored over 300 peer-reviewed journal and conference papers and been granted eight patents on evolutionary optimization.

Dr Jin is the Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and Complex & Intelligent Systems. He is an IEEE Distinguished Lecturer (2017-2019) and was the Vice President for Technical Activities of the IEEE Computational Intelligence Society (2014-2015). He is a recipient of the 2014 and 2016 IEEE Computational Intelligence Magazine Outstanding Paper Award, and the 2017 IEEE Transactions on Evolutionary Computation Outstanding Paper Award. He is a Fellow of IEEE.