Surrogate-Assisted Evolutionary Multi-Objective Optimization of Medium-Scale Problems by Random Grouping and Sparse Gaussian Modeling

Haofeng Wu, Yaochu Jin, *Fellow, IEEE*, Kailai Gao, Jinliang Ding, *Senior Member, IEEE*, and Ran Cheng, *Senior Member, IEEE*

***Abstract*—Gaussian processes (GPs) are widely employed in surrogate-assisted evolutionary algorithms (SAEAs) because they can estimate the level of uncertainty in their predictions. However, the computational complexity of GPs grows cubically with the number of training samples, the time required for constructing a GP becomes excessively long. Additionally, in SAEAs, the GP is updated using the new data sampled in each round, which significantly impairs its efficiency in addressing medium-scale optimization problems. This issue is exacerbated in multi-objective scenarios where multiple GP models are needed. To address this challenge, we propose a fast SAEA using sparse GPs for medium-scale expensive multi-objective optimization problems. We construct a sparse GP for each objective on randomly selected sub-decision spaces and optimize a multi-objective acquisition function using a multi-objective evolutionary algorithm. The resulting population is combined with the previously evaluated solutions, and k-means is used for clustering to obtain candidate solutions. Before real function evaluations, the candidate solutions in the subspace are completed with the values of the knee point in the archive. Experimental results on three benchmark test suites up to 80 decision variables demonstrate the algorithm's computational efficiency and competitive performance compared to state-of-the-art methods. Additionally, we verify its performance on a real-world optimization problem.**

***Index Terms*—Multi-objective optimization, medium-scale expensive optimization, sparse Gaussian process, random grouping.**

I. Introduction

EXPENSIVE optimization problems are those whose function evaluations (FEs) are time-consuming or costly. In real-world scenarios, multiple expensive and conflicting objective functions are often to be optimized simultaneously. Specifically, this paper addresses expensive unconstrained multi-objective optimization problems (EMOPs) [1]:

$$\min \quad \mathbf{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots f_m(\boldsymbol{x})), \boldsymbol{x} \in \mathcal{X}, \tag{1}$$

where $\mathcal{X} \subset \mathbb{R}^d$ is the decision space and $\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in \mathcal{X}$, $\boldsymbol{x}$ is a decision vector, $\mathbf{f}(\mathcal{X}) \subset \mathbb{R}^m$ is the objective space, where $\mathbf{f}(\boldsymbol{x})$ is the objective vector. The objectives often conflict with each other, and any improvement on one objective will lead to the deterioration of at least another objective, resulting in a set of optimal trade-off solutions. A solution $\boldsymbol{x_1}$ is said to dominate another solution $\boldsymbol{x_2}$ (denoted as $\boldsymbol{x_1} \succ \boldsymbol{x_2}$) iff

$$\begin{Bmatrix} \forall_i \in 1, 2, \ldots, m : f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}_2) \\ \exists_i \in 1, 2, \ldots, m : f_i(\boldsymbol{x}_1) < f_i(\boldsymbol{x}_2) \end{Bmatrix}. \tag{2}$$

A solution that is not dominated by any other feasible solutions is called a Pareto optimal solution. The collection of all Pareto optimal solutions is called the Pareto optimal set (PS) in the decision space, and the Pareto optimal solution set in the objective space is known as the Pareto optimal front (PF).

GPs are widely used for surrogate-assisted optimization of expensive problems because they can estimate the objective values and a level of uncertainty of the estimated values. These two pieces of information can be used to construct an infill criterion that guides the sampling of new data, which is then used to update the GPs, thereby achieving a balance between exploration and exploitation in the search. The process of sampling new data and updating surrogate models is known as model management [2] in surrogate-assisted evolutionary algorithms (SAEAs) [3], [4], [5]. In particular, an SAEA based on the infill criterion comprises two main components: 1) a GP is built based on the given training data to model the probability distribution of the objective function to be optimized; 2) an infill criterion is designed and optimized to select the next sample to be evaluated. Infill

Manuscript received 1 March 2023; revised 9 November 2023; accepted 29 January 2024. This work was supported in part by National Key R&D Plan Project under Grant 2022YFB3304700, in part by the National Natural Science Foundation of China under Grant 61988101, in part by 111 Project 2.0 under Grant B08015, in part by the Xinliao Talent Program of Liaoning Province under Grant XLYC2202002, in part by Liaoning Province Central Leading Local Science and Technology Development Special Project under Grant 2022JH6/100100055. *(Corresponding authors: Yaochu Jin; Jinliang Ding.)*

Haofeng Wu, Kailai Gao, and Jinliang Ding are with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China (e-mail: 1710283@stu.neu.edu.cn; jlding@mail.neu.edu.cn).

Yaochu Jin is with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China, and also with the School of Engineering, Westlake University, Hangzhou 310030, China (e-mail: jinyaochu@westlake.edu.cn).

Ran Cheng is with the Shenzhen Key Laboratory of Computational Intelligence, University Key Laboratory of Evolving Intelligent Systems of Guangdong Province, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: chengr@sustech.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TETCI.2024.3372378, provided by the authors.

Recommended for acceptance by Prof. Y. Wang.

Digital Object Identifier 10.1109/TETCI.2024.3372378

criterion driven SAEAs are originated from multiple related disciplines, including efficient global optimization (EGO) [6] and Bayesian optimization (BO) [7], [8]. Note that infill criteria are also called acquisition functions in BO. Commonly used acquisition functions include lower confidence bound (LCB) [9], upper confidence bound (UCB) [10], expected improvement (EI) [11], probability of improvement (PI) [12], entropy search (ES) [13], predictive entropy search (PES) [14], and max-value entropy search [15]. In [16], an evolutionary algorithm used to optimize the acquisition function is called evolutionary BO (EBO). To avoid confusion, we use acquisition functions to refer to infill criteria hereafter.

Acquisition function-driven SAEAs assisted by GPs have been successful in low-dimensional single- and multi-objective optimization. However, there are two main challenges in scaling up SAEAs for solving medium-scale EMOPs. First, the computational complexity of the GP is $O(N^3)$, where N is the number of training points. Extending the conventional single-objective SAEAs to multi-objective usually requires multiple GP models. Thus, multi-objective SAEAs become intractable when the number of training samples is large. Second, for single-objective optimization problems, as the search space increases exponentially with the increase of the search dimension, it is more challenging for the acquisition function to efficiently guide the search towards the global optimum [17]. Moreover, the GP tends to overestimate the uncertainty as the dimension increases, leading to an imbalance between exploration and exploitation based on the conventional acquisition function [18]. Therefore, for multi-objective SAEAs, these problems are further magnified when dealing with multiple high-dimensional acquisition functions. Although the issue of the high computational complexity of multi-objective SAEAs can be partially alleviated by replacing the GP with an ensemble surrogate [19] or a dropout deep neural network [20], efficient estimation of the uncertainty of the predictions remains a challenge. On the one hand, the performance of multi-objective SAEAs heavily relies on the quality of uncertainty estimation, which plays a key role in the acquisition function. On the other hand, an accurate estimation of the uncertainty based on an ensemble surrogate depends on the number of base learners in the ensemble. Clearly, it is unrealistic to build an ensemble with a very large number of base learners. Similarly, the estimate of the uncertainty depends on the number of dropouts in prediction, requiring increasing computation time for better estimation.

Interestingly, many GP variants that rely on a small set of *inducing points* to represent the full GP have been reported to reduce the computational complexity for modeling Big Data [21]. For example, the sparse Gaussian process using Pseudo-inputs (SPGP) is an efficient GP model for approximating the exact GP [22]. SPGP retains the favorable properties of GP, but consumes a lower computational cost ($O(M^2N)$) and memory ($O(MN)$), where M is the number of *inducing points*. Sparse GPs have received limited attention for improving the performance of medium-scale EMOPs due to two key challenges. First, sparse GPs are computationally inefficient for approximating high-dimensional objective functions. Second, as the dimensionality increases, the search space grows exponentially, leading to a significant deterioration in the model performance. Consequently, aiding the search algorithm in achieving fast convergence and finding high-quality solutions to update the surrogate model becomes challenging.

We propose a sparse GP-assisted evolutionary algorithm (SPGP-SAEA) in response to these challenges. It includes an efficient subspace SPGP model and a corresponding model management strategy, enabling it to address medium-scale EMOPs efficiently. In summary, this paper makes the following key contributions.

- The random grouping technique, which is widely used in large-scale optimization [23], is introduced to build a SPGP model for each objective based on a randomly chosen subspace, thereby further improving the computational efficiency of the sparse GP models and enhancing the search efficiency in optimizing the acquisition function.
- To select solutions for evaluation, we propose a new model management strategy for SPGP models with randomly selected subspaces. We use NSGA-II to optimize the acquisition function in a randomly selected subspace. The optimized population is then merged with the corresponding solutions from the randomly selected subspace in the archive. We apply the k-means algorithm to obtain cluster centers of the merged solutions within the selected subspace. Finally, we consider the value corresponding to the unselected subspace of the knee solution in the archive as an infill point. Each cluster center in the selected subspace is completed by the infill point to create complete solutions for evaluations.
- The proposed algorithm is assessed on three benchmark test suites in comparison with the state-of-the-art surrogate-assisted multi-objective evolutionary algorithms (MOEAs) and further verified on a real-world operational optimization problem of crude oil distillation unit (CODU).

The remainder of this paper is organized as follows. Section II reviews some related work. Section III details the the proposed SPGP-SAEA. Experimental results on three test suites are presented and discussed in Section IV, followed by a description of the comparative results on the operational optimization problem of CODU in Section V. Section VI gives a conclusion and future work.

II. Related Work and Motivations

A. Related Work

Several single-objective SAEAs for solving high-dimension problems have been reported in the literature. In [24], Tian et al. found that the candidate solution is far from the training data in high-dimensional space. The uncertainty level estimated by the GP becomes less accurate. Therefore, a new multi-objective infill criterion (MIC) is proposed to address this issue and select superior candidate solutions. However, the cubic time complexity of the GP makes the SAEA inefficient. To address the computational cost issue of GP within SAEAs, Zhan et al. propose an incremental method for updating the GP model [25]. This method involves updating the model based on the equations from the previously trained model rather than constructing a

new model from scratch with new data. As a result, the time complexity of updating GP is reduced to quadratic. However, the method in [25] only considers the influence of the surrogate models on the efficiency of SAEAs. A large search space also diminishes the optimization efficiency of the acquisition function. Therefore, In [17], Kandasamy et al. propose to tackle high-dimensional problems by assuming a latent additive structure with disjoint dimensions, then the acquisition function can be optimized independently in multiple low-dimensional subspaces. Guo et al. develop a surrogate-assisted differential evolution algorithm using dimensionality reduction techniques to reduce the number of variables by random projection [26]. Li et al. [27] adopted a dropout technique in BO that drops the decision variables at a pre-specified probability to obtain a small subset of effective decision variables. Fu et al. develope an SAEA for solving single-objective problems up to 200 dimensions that sequentially optimize randomly selected subproblems within a population and fill in the decision variables with those of the best solution found so far [28]. Using the cooperative swarm optimization algorithm, Sun et al. propose a surrogate-assisted social learning particle swarm optimization algorithm [29]. Liu et al. propose a surrogate-assisted multi-population particle swarm optimize algorithm (SA-MPSO) [30]. In SA-MPSO, MPSO and a trust-region local search technology work together to enhance the search efficiency.

Thanks to the population based nature, MOEAs have inherent advantages in solving multi-objective optimization problems (MOPs) [1]. Generally, existing MOEAs can be divided into four categories. The first category is known as dominance-based approaches, e.g., NSGA-II [31], SPEA2 [32], among many others. They use dominance-based strategies to perform environmental selections. The second category includes decomposition-based MOEAs, such as MOEA/D [33] and RVEA [34]. They decompose a complex multi-objective problem into multiple simple sub-problems, and then solve them collaboratively. The performance indicator based approaches belong to the third category, e.g., IBEA [35], HypE [36]. The fourth category of algorithms combines dominance-based and decomposition-based approaches. These algorithms are often referred to as hybrid algorithms [37], [38]. Recently, research on MOEAs have also made remarkable progresses in scaling up the decision space [39], [40]. Most existing MOEAs, however, require a large number of FEs to obtain satisfactory solutions, making them less practical for EMOPs, which are widely seen in the real-world [41].

Facing the challenges of EMOPs, it is intuitive to extend traditional single-objective SAEAs to multi-objectives by combining the advantages of MOEAs and SAEAs. Knowles et al. propose a MOEA assisted by efficient global optimization (ParEGO), which linearly combines the objective functions with several random weights and maximizes EI [42]. This method adopts a single-point acquisition function and selects only one candidate solution in each iteration and is verified on low-dimensional problems only. Zhang et al. apply the decomposition-based multi-objective algorithm MOEA/D [33] to the SAEAs framework and proposed (MOEA/D-EGO) [43]. While its performance is better compared with ParEGO, it suffers relatively high computational costs by optimizing the hyperparameters of the GP using differential evolution and building multiple local GP models. Chugh et al. propose a Kriging-assisted reference vector guided many-objective optimization algorithm (K-RVEA) [44]. Despite its promising performance, building a GP model for each objective is time-consuming. To improve the performance of GP, an ensemble SAEA framework is proposed by Lin et al., in which each base learner is a GP model [45]. Song et al. propose to use one influential point-insensitive GP model to assist a two-archive evolutionary algorithm (KTA2) [46]. Considering the non-dominance relation of the solutions produced by the optimization, Hao et al. propose a relational predictive model-assisted MOEA [47]. Lin et al. adopte an adaptive dropout mechanism to select the necessary dimensions to model the RBF network and assist the evolutionary algorithm search [48].

B. Sparse Gaussian Process Using Pseudo-Inputs

Given a training dataset $\mathcal{D}$ of size N, the dimension of the input (decision variable) $\boldsymbol{X}$ is D, and the corresponding function (objective) value is $\boldsymbol{y}$. GP is defined as the following distribution [49]:

$$f(\boldsymbol{x}) \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_{N,N}), \tag{3}$$

where $[\boldsymbol{K}_{N,N}]_{nn'} = k(x^n, x^{n'})$, $\boldsymbol{K}_{(,)}$ is the covariance matrix, $k_{(,)}$ is the kernel function. A Gaussian kernel in the form of automatic relevance determination (ARD) is used in the covariance matrix:

$$k(\boldsymbol{x}^n, \boldsymbol{x}^{n'}) = \sigma_f \exp\left[-\frac{1}{2}\sum\nolimits_d^D l^d (x_d^n - x_d^{n'})^2\right], \boldsymbol{\theta} = \{\sigma_f, \boldsymbol{l}\}, \tag{4}$$

where $\boldsymbol{\theta}$ is the hyperparameter of kernel. It comprises amplitude (σ_f) and length scale ($\boldsymbol{l}$), where σ_f determines the vertical scale over which the function changes and $\boldsymbol{l}$ determines its variation rate. The predicted distribution of an unknown input $\boldsymbol{z}^*$ is:

$$\begin{aligned} p(f_* \mid \boldsymbol{y}) = \mathcal{N}\Big(&\boldsymbol{k}_{*,N}(\boldsymbol{K}_{N,N} + \sigma_{noise}^2 \boldsymbol{I})^{-1}\boldsymbol{y}, \\ &K_{*,*} - \boldsymbol{K}_{*,N}(\boldsymbol{K}_{N,N} + \sigma_{noise}^2 \mathbf{I})^{-1}\boldsymbol{K}_{N,*}\Big), \end{aligned} \tag{5}$$

where $[\boldsymbol{k}_{*,N}] = k(\boldsymbol{z}^*, \boldsymbol{x}^n)$, $K_{*,*} = K(\boldsymbol{z}^*, \boldsymbol{z}^*)$. The predicted distribution depends on the hyperparameter $\boldsymbol{\theta}$ and σ_{noise}.

Even though the GP has been successfully used in surrogate-assisted evolutionary optimization, it suffers from the curse of dimensionality as GP modeling requires the inverse operation of an $N \times N$ matrix $\boldsymbol{K}_{N,N}$, typically $O(N^3)$ for training and $O(N^2)$ for prediction, where N is the number of training samples. The most straightforward idea is to find a sparse approximation of the original covariance matrix, i.e., SPGP [22]. SPGP uses a pseudo-data set $\bar{D} = \{\bar{\boldsymbol{X}}, \bar{\boldsymbol{f}}\}$ to represent the real training data set, where $\bar{\boldsymbol{X}} = \{\bar{\boldsymbol{x}}_m\}_{m=1}^M$ is called pseudo-input points or inducing points, they represent the entire training inputs [50]. An illustrative example of the training data and $\bar{\boldsymbol{X}}$ are plotted in Fig. S1.[1] The pseudo-input set M is much smaller than the

[1]Fig. Sxxx represents a figure in the Supplementary material with a serial number of Sxxx.

training set N, i.e., $M \ll N$. The predicted distribution of an unknown input is:

$$p(f_*|\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{k}_{*,M}\sum \boldsymbol{K}_{M,N}\boldsymbol{\Lambda}^{-1}\boldsymbol{y}, K_{*,*} - Q_{*,*} + \boldsymbol{k}_{*,M}\sum \boldsymbol{k}_{M,*}\right), \quad (6)$$

where $\sum = (\boldsymbol{K}_{M,M} + \boldsymbol{K}_{M,N}\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{M,N})^{-1}$, $\boldsymbol{\Lambda} = diag[\boldsymbol{K}_{N,N} - \boldsymbol{Q}_{N,N} + \sigma^2_{noise}\mathbf{I}]$ and $[\boldsymbol{k}_{*,M}] = k(\boldsymbol{z}^*, \bar{\boldsymbol{x}}_m)$. $Q_{a,b} \triangleq \boldsymbol{K}_{a,M}\boldsymbol{K}^{-1}_{M,M}\boldsymbol{K}_{M,b}$. The previous methods have restricted the selection of pseudo-input points from the training and test sets, which is a combinatorial optimization problem. SPGP relaxes this constraint and converts the selection of pseudo-input points into a continuous optimization problem by computing the maximum marginal likelihood:

$$p(\boldsymbol{y}|\bar{\boldsymbol{X}}, \boldsymbol{\theta}) = -\frac{1}{2}|\boldsymbol{Q}_{N,N} + \boldsymbol{\Lambda}| - \frac{1}{2}\boldsymbol{y}^T(\boldsymbol{Q}_{N,N} + \boldsymbol{\Lambda})^{-1}\boldsymbol{y} - \frac{n}{2}\log(2\pi). \quad (7)$$

This work uses the gradient ascent method to train all the parameters $\Theta = \{\boldsymbol{\theta}, \sigma_{noise}, \bar{X}\}$, where the number of parameters is $Md + |\boldsymbol{\theta}| + 1$, and d and M are the numbers of dimensions and pseudo-input points, respectively. The computational cost of the marginal likelihood of SPGP is $O(M^2\,N)$, which is because an inverse operation of matrix $\boldsymbol{Q}_{N,N} + \boldsymbol{\Lambda}$ instead of the matrix $\boldsymbol{K}_{N,N}$ of the standard GP is performed. More details about SPGP can be seen in the Supplementary material.

C. Acquisition Functions

Given an expensive objective function $f : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$, the analytical form of f is unknown, the derivative cannot be obtained, and the process of obtaining observation samples is expensive. In this scenario, to obtain an optimal input $\boldsymbol{x}^* \subseteq \mathcal{X}$, it is only meaningful to limit the number of evaluations. SAEAs can obtain satisfactory results with a few FEs. As the most commonly used surrogate model in SAEAs, GP and SPGP can provide a prediction (i.e. the mean) together with an estimated prediction uncertainty (i.e. the variance). Then, considering the predicted function value and estimation uncertainty, an acquisition function is constructed and optimized to sample a new solution. Specifically, the LCB acquisition function is as follows:

$$LCB(\boldsymbol{x}) = \hat{y}(\boldsymbol{x}) - \omega\hat{s}(\boldsymbol{x}), \quad (8)$$

where $\hat{y}(\boldsymbol{x})$ and $\hat{s}(\boldsymbol{x})$ are the mean and variance predicted by GP, and ω is a positive constant that balances $\hat{y}(\boldsymbol{x})$ and $\hat{s}(\boldsymbol{x})$. In most GP-based SAEAs, EI is used. The expression is as follows:

$$EI(\boldsymbol{x}) = (y_{\min} - \hat{y}(\boldsymbol{x}))\Phi\left(\frac{y_{\min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) + \hat{s}(\boldsymbol{x})\phi\left(\frac{y_{\min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right), \quad (9)$$

where Φ and ϕ are the standard normal distribution and the probability density function, respectively. $y_{\min}$ is the minimum objective value.

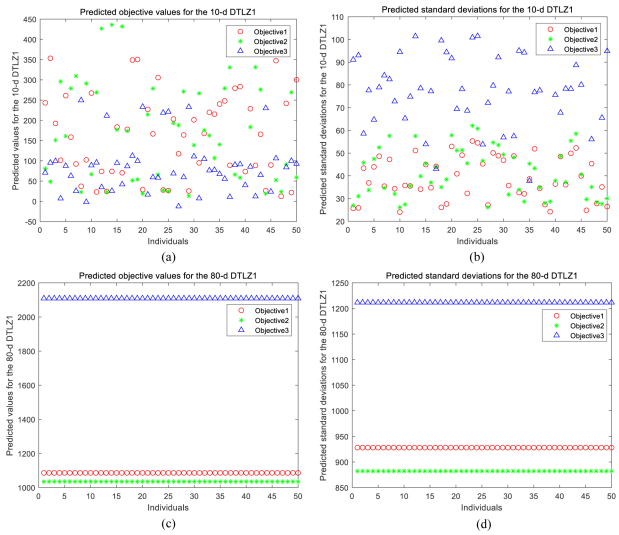


Fig. 1. Predicted objective values and standard deviations on DTLZ1.

D. Motivations

As previously analyzed, a full GP model is unsuitable for medium-scale EMOPs due to its high computational complexity. In addition, with a limited number of training samples, it is impractical to build one global GP to properly capture the overall fitness landscape for high-dimensional multimodal problems. The acquisition function may not work without proper predictions of the objective values and a sensible estimation of the uncertainty of their prediction. To empirically illustrate this, an empirical analysis of the above problem is presented as follows: we plot in Fig. 1 the predictions and their standard deviations of 50 individuals in a population when a GP-assisted evolutionary algorithm is used to optimize the 10-dimensional and 80-dimensional DTLZ1 problems [51], respectively. From Fig. 1(a) and (b), we can observe that the 50 predicted values and their estimated uncertainty for the 10-dimensional function using the full GP clearly distinguish themselves, implying the GP model works properly in this case. However, the predictions and their estimated uncertainty become indistinguishable for the 80-dimensional function, as shown in Fig. 1(c) and (d), respectively. These observations are consistent with those in [24], [45].

Although replacing GP with sparse GP as a surrogate model to reduce the computational complexity appears straightforward [52], [53], several issues must be clarified. First, it has been shown that it is highly time-consuming to determine the pseudo-input points for high-dimensional problems. To resolve this problem, McIntire et al. [54] proposed a weighted updating online GP to replace the induced point selection scheme of sparse GP, making it more suited for BO. Thus, it is unpractical to directly apply sparse GP to BO of high-dimensional problems. Secondly, in [55], Yang et al. have found that the sparse GP may provide over-confident estimations of the confidence levels, which affects the balance of the acquisition function for exploration and exploitation. To address this issue, they derive a new regularized marginal likelihood to find the optimal frequency

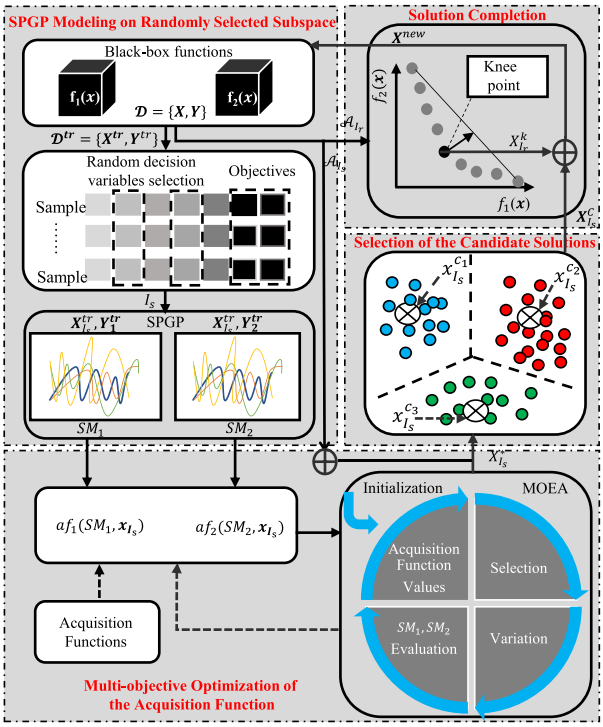


Fig. 2. Proposed SPGP-SAEA framework for an MOP.

for a sparse spectrum Gaussian process (SSGP) based surrogate model. Finally, it should be pointed out that sparse GP achieves high computational efficiency at the cost of a decreased approximation accuracy. This does not prevent us from using sparse surrogates as it has been indicated in [56], [57] that one might even benefit from the approximation errors in the surrogates, provided that the surrogates are properly management. To sum up, sparse GP models are promising for effective and efficient surrogate-assisted optimization if we use them in a relatively low-dimensional space together with an appropriate surrogate management strategy, which is exactly the motivation of this work.

III. Proposed Algorithm

A. Overview of SPGP-SAEA

SPGP-SAEA is an iterative algorithm with four steps in each round, as illustrated in Fig. 2 using bi-objective optimization as an example. Firstly, we construct an SPGP model on subspace for each objective function. Next, a bi-objective acquisition function is formulated within this subspace, and we use MOEA to optimize it. Upon obtaining an optimized population, we select three candidate solutions: $\boldsymbol{x}^1_{I_s}$, $\boldsymbol{x}^2_{I_s}$, and $\boldsymbol{x}^3_{I_s}$. Finally, we employ the Knee solution to complete the values of the remaining decision space for $\boldsymbol{x}^1_{I_s}$, $\boldsymbol{x}^2_{I_s}$, and $\boldsymbol{x}^3_{I_s}$ for the expensive objective FEs.

The procedure of SPGP-SAEA is detailed in Algorithm 1. Initialization is required before the loop starts (Lines 1 and 2). In which, to generate the initial data $\{\boldsymbol{X}, \boldsymbol{Y}\} = \{\boldsymbol{x}_i, f_{1:m}(\boldsymbol{x}_i)\}_{i=1}^N$, Latin hypercube sampling (LHS) [58] is used to obtain the N points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ from the decision space $\mathcal{X} \subseteq \mathbb{R}^d$ and to evaluate them with the real expensive objective functions $\mathbf{f}(\boldsymbol{x})$, all the data that have been evaluated are saved in database $\mathcal{D}$. After that, the number of FE is equals to N. The loop phase follows this, we update the archive $\mathcal{A}$, the non-dominated solutions, $\{\boldsymbol{X}^{nd}, \boldsymbol{Y}^{nd}\}$, that have been evaluated are stored in $\mathcal{A}$ (Line 4). Then, we select a part of the data from $\mathcal{D}$ as the training data $\{\boldsymbol{X}^{tr}, \boldsymbol{Y}^{tr}\}$, which are stored in the training database $\mathcal{D}^{tr}$ (Line 5). We divide the decision space into two groups and select one to build a surrogate model and to perform optimization of the acquisition function. The index corresponding to the selected group of decision space is denoted by I_s (Line 6). Thus, the surrogate in the randomly selected subspace is used to approximate each objective function $\boldsymbol{f} : \mathcal{X} \to R^m$ to replace the expensive objective FEs (Lines 8-10). After updating the SPGP model, we choose the same acquisition function for each objective, which forms an MOP whose objectives are the acquisition functions. Any MOEA can be applied to solve the acquisition function to obtain the last generation population $X^*_{I_s}$ (Line 11). After that, we use $X^*_{I_s}$ combined with archived non-dominated solutions $X^{nd}_{I_s}$ to select the most promising candidate solutions for evaluations (Lines 12-14). Finally, we use the value of knee point in the archive to fill in the remaining decision variables of the selected solutions to generate complete solutions $\mathbf{X}^{new} \in \mathbb{R}^d$ for evaluations using the expensive objective functions (Line 15). After sampling these new data $\{\boldsymbol{X}^{new}, \mathbf{Y}^{new}\}$, SPGP-SAEA updates $\mathcal{D}$ and the number of FE, then goes to the next iteration (Lines 16-17).

B. SPGP Modeling on Randomly Selected Subspace

SPGP approximation based on a small set of M pseudo-input points ($M \ll N$) is able to reduce the computational complexity from $O(N^3)$ to $O(M^2N)$. For the predicted distribution of SPGP shown in equation (6), we can get $\boldsymbol{K}_{M,N} = \boldsymbol{K}_{N,N} = \boldsymbol{Q}_{N,N}$ when $M = N$ and if we assume $\bar{\boldsymbol{X}} = \boldsymbol{X}$, in this case, $\boldsymbol{\Lambda} = \sigma^2_{noise}\mathbf{I}$ and the predicted distribution of SPGP is

$$p(f_* \mid \boldsymbol{y}) = \mathcal{N}\Big(\boldsymbol{K}_{*,N}(\boldsymbol{K}_{N,N} + \sigma^2_{noise}\boldsymbol{I})^{-1}\boldsymbol{y}\,,$$
$$\boldsymbol{K}_{*,*} - \boldsymbol{K}_{*,N}(\boldsymbol{K}_{N,N} + \sigma^2_{noise}\mathbf{I})^{-1}\boldsymbol{K}_{N,*}\Big). \quad (10)$$

It can be seen from (5) and (10) that SPGP becomes the full GP. Snelson treats pseudo-input points as additional parameters of the model [22] and obtains the optimal solutions by optimizing the likelihood function. In SPGP, the number of parameters is $Md + |\boldsymbol{\theta}| + 1$, i.e., SPGP has Md additional parameters than the standard GP, since a large number of parameters may lead to overfitting, unless more training data is available, which, however, increases computational complexity. To reduce the required number of pseudo-input points (M), we first select a subset of the most relevant training data and then construct the SPGPs on a lower dimensional space rather than the original decision space.

1) Training Data Selection: At the initial stage of optimization, all data $\{\boldsymbol{X}, \boldsymbol{Y}\}$ in $\mathcal{D}$ are selected as the training data $\{\boldsymbol{X}^{tr}, \boldsymbol{Y}^{tr}\}$, they are stored in $\mathcal{D}^{tr}$. At this stage, we assume the number of data in $\mathcal{D}$ is still smaller than $N_{\max}$ (capacity

Algorithm 1: SPGP-SAEA.

Input: $\mathcal{X}$: decision space; $\mathbf{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), ..., f_m(\boldsymbol{x}))$, m: number of black-box objective functions; d: dimension of decision space; d_s: subspace size of the selected group; $maxFE$: maximum number of function evaluations; $N_{\max}$: the maximum number of training sets; C: batch size.
Output: A set of non-dominated solutions

```
/* Initialization                                                    */
```
Generate N initial sample points, $\boldsymbol{X}$, by LHS, $\boldsymbol{X} \leftarrow LHS(d, N)$, $\boldsymbol{Y} \leftarrow \{\mathbf{f}(\boldsymbol{x}_1), ..., \mathbf{f}(\boldsymbol{x}_N)\}$;
$\mathcal{D} \leftarrow \{\boldsymbol{X}, \boldsymbol{Y}\}$, $FE = N$;
while $FE < maxFE$ **do**
  $\mathcal{A} \leftarrow \{\boldsymbol{X}^{nd}, \boldsymbol{Y}^{nd}\}$, select Non-dominated solution from $\mathcal{D}$;
  `/* SPGP Modeling on Randomly Selected subspace */`
  $\mathcal{D}^{tr} \leftarrow$ `Training data selection` $(\mathcal{D}^{tr}, N, N_{max})$;
  $I_s \leftarrow rand(d, d_s)$;
  $I_r \leftarrow \{1, ..., d\} \setminus I_s$;
  for $j = 1$ *to* m **do**
    $SM_j \leftarrow$ `SPGP` $(\mathcal{D}^{tr}, I_s, j)$;
  end
  `/* Multi-objective Optimization of the Acquisition function */`
  $\boldsymbol{X}^*_{I_s} \leftarrow \arg\max_{\boldsymbol{x}_{I_s} \in \mathcal{X}_{I_s}} AF\ (\boldsymbol{x}_{I_s})$, refer to equation (11);
  `/* Selection of the Candidate Solutions */`
  $\boldsymbol{X}^{nd}_{I_s} \leftarrow$ Select the value corresponding to index I_s in $\mathcal{A}$;
  $P \leftarrow \boldsymbol{X}^*_{I_s} \cup \boldsymbol{X}^{nd}_{I_s}$;
  $\boldsymbol{X}^c_{I_s} \leftarrow kmeans\,(P)$;
  `/* Solution Completion */`
  $\boldsymbol{X}^{new} \leftarrow$ `Solution Completion` $(\boldsymbol{X}^c_{I_s}, I_r, \mathcal{A})$;
  $\boldsymbol{Y}^{new} \leftarrow \mathbf{f}(\boldsymbol{X}^{new})$, $\mathcal{D} \leftarrow \mathcal{D} \cup \{\boldsymbol{X}^{new}, \boldsymbol{Y}^{new}\}$;
  $FE = FE + |\boldsymbol{X}^{new}|$, $N = FE$;
end

Algorithm 2: `Training Data Selection`.

Input: $\mathcal{D}$, database; N, the number of data in $\mathcal{D}$; N_{max}, capacity of the training database.
Output: $\mathcal{D}^{tr}$

if $N \leq N_{max}$ **then**
  $\mathcal{D}^{tr} \leftarrow \{X, Y\}$;
else
  $\mathcal{D}^{tr} \leftarrow$ Select the first $N_{\max}/2$ data points according to the non-dominance rank and the crowding distance;
  $\mathcal{D}^{tr} \leftarrow \mathcal{D}^{tr} \cup$ Randomly select $N_{max}/2$ data from $(\mathcal{D} \setminus \mathcal{D}^{tr})$
end

Algorithm 3: `SPGP`.

Input: $\mathcal{D}^{tr}$, training database; I_s, indices of the decision space; j, indice of objective function.
Output: $\bar{\boldsymbol{X}}$, $\boldsymbol{\theta}$, σ_{noise}.

$\{\boldsymbol{X}^{tr}_{I_s}, \boldsymbol{Y}^{tr}_j\} \leftarrow$ Select the training inputs corresponding to I_s, and select the training outputs corresponding to j-th objective;
$\bar{\boldsymbol{X}}_0$, $\boldsymbol{\theta}_0$, $\sigma_0 \leftarrow$ Initialize hyperparameters, let pseudo-inputs to a random subset of training inputs;
$\bar{\boldsymbol{X}}$, $\boldsymbol{\theta}$, $\sigma_{noise} \leftarrow$Maximize the marginal likelihood according to equation (7) and then obtain the optimal solution of all hyperparameters by the gradients ascent.

of the training database). In each iteration of SPGP-SAEA, new samples will be stored in the database $\mathcal{D}$, i.e., $\mathcal{D} = \mathcal{D} \cup \{\boldsymbol{X}^{new}, \boldsymbol{Y}^{new}\}$. To reduce the computational complexity of the SPGP models while maintaining the approximation quality, When the number in $\mathcal{D}$ exceeds N_{max}, we select a subset from $\mathcal{D}$ for training the models. Similar to the approach reported in [42], we apply non-dominated sorting on the solutions in the archive and calculate the crowding distance as done in NSGA-II. The first $N_{\max}/2$ data points are selected according to the non-dominance rank and the crowding distance, which are most likely the newly evaluated data and the most important solutions for the multi-objective problem. In addition, the rest $N_{\max}/2$ data points are randomly selected from the remaining solutions in $\mathcal{D}$, which enables the updated surrogate to learn the global profile of the fitness landscape. `Training data selection` is provided in Algorithm 2.

2) Constructing a SPGP for Each Objective Function on a Randomly Selected Subspace: After selecting training data, we divide the decision space into two groups. One is a randomly chosen subset d_s of the decision space of the original optimization problem, and the corresponding indices of the decision space in this group are denoted as I_s. We then construct a SPGP model for j-th objective function on this subspace. The corresponding training inputs and outputs are denoted as $X^{tr}_{I_s}$, Y^{tr}_j, respectively. The rest of the decision variables are in the other group, whose indices are denoted as I_r, where $I_s \cup I_r = \{1, \ldots, d\}$ and $I_s \cap I_r = \emptyset$.

In SPGP, a crucial issue is the optimization of pseudo-input points, which is a non-convex optimization problem, as indicated by equation (7). In the proposed SPGP-SAEA framework, since SPGP frequently updates according to the new observations, we randomly initialize pseudo-input points $\bar{\boldsymbol{X}}_0$, $\boldsymbol{\theta}_0$ and σ_0 from the training inputs $\boldsymbol{X}^{tr}_{I_s}$, which is determined by the training data selection. After that, maximize the marginal likelihood according to equation (7) and then obtain the optimal solution of all hyperparameters $\bar{\boldsymbol{X}}$, $\boldsymbol{\theta}$ and σ_{noise} by the gradients ascent. At this point, the SPGP model SM_j for the j-th objective function has been trained, the process of train SPGP is provided in Algorithm 3. For an unknown input $\boldsymbol{x}_{I_s}$, the predicted value of its j-th objective function can be obtained from the posterior distribution of SM_j, where the posterior distribution is computed by equation (6) as $\mathcal{N}_1(\boldsymbol{\mu}_1(\boldsymbol{x}_{I_s}), \boldsymbol{\sigma}_1(\boldsymbol{x}_{I_s})), \ldots, \mathcal{N}_m(\boldsymbol{\mu}_m(\boldsymbol{x}_{I_s}), \boldsymbol{\sigma}_m(\boldsymbol{x}_{I_s}))$, respectively, where $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$, $i = 1, \ldots, m$ are mean and variance, respectively. By benefiting from the SPGP models built in a subspace of a reduced dimension, fewer pseudo-input points are needed, making optimizing in d_s-dimensional space easier. In addition, they can represent most of the information contained in the entire training set while reducing the size of the covariance matrix, further facilitating the use of the less computationally demanding SPGP models.

C. Multi-Objective Optimization of the Acquisition Function

After obtaining sparse models $SM_1, \ldots, SM_m$ on the selected subspace of the decision variables for each objective function $f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})$ as introduced in Section III-B, promising

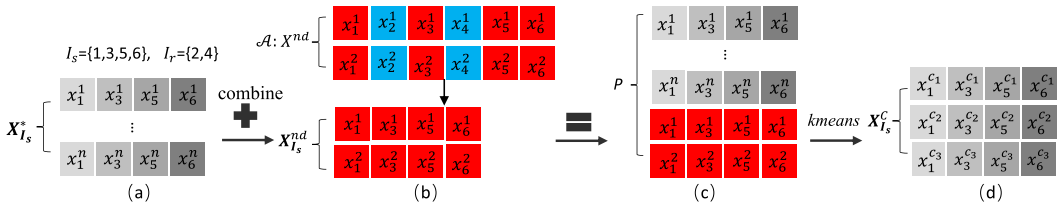


Fig. 3. Illustration of the selection of candidate solutions.

candidate solutions $\boldsymbol{X}^*_{I_s}$ are obtained by maximizing the acquisition function for each objective:

$$\arg\max_{\boldsymbol{x}_{I_s}\in\mathcal{X}_{I_s}} AF\left(\boldsymbol{x}_{I_s}\right) = \left(af_1(SM_1, \boldsymbol{x}_{I_s}), \ldots, af_m(SM_m, \boldsymbol{x}_{I_s})\right), \tag{11}$$

where $af_j(\cdot),\ j = 1, \ldots, M$ are the acquisition function. This paper mainly uses LCB as the acquisition function because it provides a good trade-off between exploration and exploitation. It achieves this balance through the parameter ω, and its exploration level can be regulated. Moreover, LCB is relatively easy to implement, and it is easy to compute. However, it is important to note that other acquisition functions, such as EI, can also be used in the algorithm presented in this paper. A comparison of the experimental results regarding LCB and EI is presented in the Supplementary material. NSGA-II is a dominance-based parameterless MOEA and is relatively insensitive to different multi-objective scales and different Pareto fronts. Belakaria et al. [59] have also demonstrated that it is effective to use MOEA to solve multi-objective acquisition function in BO. (see Section III of the Supplementary material).

D. Selection of the Candidate Solutions

Once a multi-objective acquisition function has been optimized, the last generation population ($\boldsymbol{X}^*_{I_s}$) is obtained, we first remove those that are too close to the already evaluated data points ($\boldsymbol{X}_{I_s}$) by predefining a threshold of 10^{-5} in Euclidean distance in the decision space. Then, we combine solutions $\boldsymbol{X}^{nd}_{I_s}$ in the archive and $\boldsymbol{X}^*_{I_s}$ to form the population P (Line 13 of Algorithm 1). Finally, to reduce the number of solutions to be sampled in each round, we need to choose a subset of representative solutions from these non-dominated solutions that have been obtained so far. To this end, we adopt k-means to select the solutions $\boldsymbol{X}^c_{I_s}$ (i.e., the cluster centers C) to promote diversity. In the process of clustering the final population, the non-dominated solutions that have been evaluated are added to enhance the diversity of solutions. This approach complements the acquisition functions, which strike a balance between exploration and exploitation in the search. The candidate solutions selection process is illustrated in Fig. 3. Consider a decision space with $d = 6$ for the original MOP and $I_s = 1, 3, 5, 6$. Assuming that the population size of NSGA-II is n, the final population obtained by solving the optimization problem (11) is denoted as $X^*_{I_s}$, as shown in Fig. 3(a). Currently, two nondominated solutions exist in $\mathcal{A}$, x^{nd}_1 and x^{nd}_2, which have values $x^{nd}_{1_{I_s}}$ and $x^{nd}_{2_{I_s}}$ on I_s, as depicted in Fig. 3(b). Then, these two solutions are combined with $X^*_{I_s}$ to form the population P, as illustrated in Fig. 3(c). Finally, when the number of cluster centers is set to 3, we obtain the cluster centers of the clustering algorithm for P, as shown in Fig. 3(d).

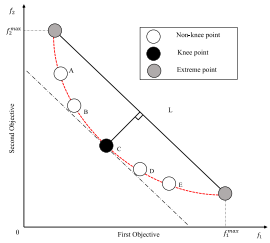


Fig. 4. Illustration of a knee point of a bi-objective optimization problem. The black circle C that is the farthest from the hyperplane L is the knee point.

E. Solution Completion

Note that the candidate solutions $\boldsymbol{X}^c_{I_s}$ obtained in the previous step are incomplete since they contain the randomly chosen subset of the decision variables only. Thus, we need to appropriately fill them with the remaining decision variables $\boldsymbol{X}_{I_r}$ to generate complete solutions $\boldsymbol{X}^{new} = [\boldsymbol{X}^c_{I_s}, \boldsymbol{X}_{I_r}]$ before they can be evaluated. The most straightforward idea is to use the best solutions $\boldsymbol{x}^{best}$ from the archive $\mathcal{A} = \{\boldsymbol{X}^{nd}, \boldsymbol{Y}^{nd}\}$ with the remaining decision variables $\boldsymbol{x}_{I_r}$. For multi-objective optimization, the quality of a non-dominated solution can be measured by a performance indicator such as the hypervolume [36], [60] (HV). Therefore, we can select the best solution in the archive that maximizes HV:

$$\boldsymbol{x}^{best} = \arg\max_{\boldsymbol{x}\in\mathcal{X}} [HV(\boldsymbol{X}^{nd}) - HV(\boldsymbol{X}^{nd}\backslash\{\boldsymbol{x}\})]. \tag{12}$$

Then we can fill in the remaining decision variables of the selected candidate solutions ($\boldsymbol{X}^c_{I_s}$) with those $\boldsymbol{x}^{best}_{I_r}$. Unfortunately, finding the non-dominated solution that maximizes the hypervolume is computationally expensive [61]. In multi-objective optimization, it has been found that knee points on the non-dominated front contribute most to the hypervolume [62]. Therefore, we identify the knee point $\boldsymbol{x}^k$ in the archive as the best solution and use its $\boldsymbol{x}^k_{I_r}$ to fill in $\boldsymbol{X}^c_{I_s}$. Specifically, we adopt the distance between non-dominated solutions and the hyperplane L to identify the knee point, as illustrated in Fig. 4, the hyperplane L is defined as follows:

$$Af_1\left(\boldsymbol{x}\right) + Bf_2\left(\boldsymbol{x}\right) + C = 0, \tag{13}$$

where the extreme points in the archive can determine the parameters A, B and C. Then the distance from a solution $\boldsymbol{x}^k$

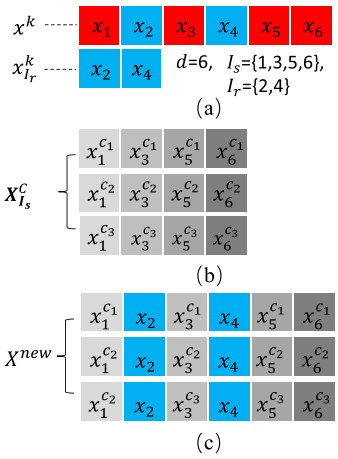


Fig. 5. Illustration of Solution Completion. $\boldsymbol{X}_{I_s}^c$ are the three cluster centers that need to be completed, $\boldsymbol{x}_{I_r}^k$ is the remaining decision variable of knee point. The complete solutions $\boldsymbol{X}^{new}$ are achieved by combining $\boldsymbol{X}_{I_s}^c$ with $\boldsymbol{x}_{I_r}^k$.

to L can be calculated as follows:

$$d\left(\boldsymbol{x}^k, L\right) = \begin{cases} \frac{|Af_1+Bf_2+C|}{\sqrt{A^2+B^2}} & if\, Af_1 + Bf_2 + C < 0 \\ -\frac{|Af_1+Bf_2+C|}{\sqrt{A^2+B^2}} & \text{otherwise.} \end{cases} \tag{14}$$

In the non-dominated solution set, the point with the largest distance to the hyperplane L is the knee point. It is worth noting that, rather than searching for the knee point in the remaining subspace, we find the knee point in the original space and use its value on I_r as the infill point to complete the candidate solution. Consider a decision space $d = 6$ of the original MOP to elaborate on completing the candidate solutions. The index corresponding to the randomly chosen subset of decision variables is $I_s = \{1, 3, 5, 6\}$ and the indices for the remaining group of decision variables are $I_r = \{2, 4\}$. Assuming that the knee point in the archive is $\boldsymbol{x}^k$, as shown in Fig. 5(a). Therefore $\boldsymbol{x}_{I_r}^k$ will be used to fill in the selected solutions. If the number of cluster centers is set to 3, i.e., $C = 3$, and $\boldsymbol{X}_{I_s}^c$ are three cluster centers obtained by the clustering algorithm, as illustrated in Fig. 5(b). Finally, the complete solutions $\boldsymbol{X}^{new}$ are achieved by combining $\boldsymbol{X}_{I_s}^c$ with $\boldsymbol{x}_{I_r}^k$, as shown in Fig. 5(c). These three complete candidate solutions will be evaluated by expensive objective functions and included in the archive.

Algorithm 4: `Solution Completion.`

Input: $\boldsymbol{X}_{I_s}^c$, candidate solutions; I_r, the remaining decision subspace index; $\mathcal{A}$, the archive.
Output: $\boldsymbol{X}^{new}$, completed solutions to be evaluated using the expensive objective functions.

1 $E \leftarrow$ Find the extreme points from $\mathcal{A}$;
2 $A, B, C \leftarrow$ Use extreme points to calculate the parameters of hyperplane;
3 $d\left(\boldsymbol{x}^k, L\right) \leftarrow$ Use equation (14) to calculate the distance between each solution in $\mathcal{A}$ and L;
4 $\boldsymbol{x}^k \leftarrow$ Find the knee point based on the distance;
5 $\boldsymbol{x}_{I_r}^k \leftarrow$ Obtain the value of the knee point on the I_r;
6 $\boldsymbol{X}^{new} \leftarrow$ Each cluster center $\boldsymbol{X}_{I_s}^c$ is completed with $\boldsymbol{x}_{I_r}^k$ to obtain the complete solutions ($\left[\boldsymbol{X}_{I_s}^c, \boldsymbol{x}_{I_r}^k\right]$).

IV. Experimental Studies

In this section, we first introduce the details of the experimental settings. Then, we perform ablation studies of each important component of SPGP-SAEA. After that, we compare the proposed algorithm with the other state-of-the-art algorithms on DTLZ [51], UF [63]. In addition, we compare the proposed algorithm with HeE-MOEA and EDN-ARMOEA on WFG [51]. Finally, we discuss the proposed algorithm.

A. Test Problems

Benchmark test suites DTLZ, UF and WFG are used to test the performance of the proposed SPGP-SAEA, and the properties of the seventeen test functions are summarized in Table SII.[2] The decision variables are set to 20, 40, and 80, respectively. As suggested in [64], DTLZ1a and DTLZ3a are modified by changing the cosine value in DTLZ1 and DTLZ3 from 20π to 2π. DTLZ4a changes the parameter α in DTLZ4 from 100 to 10.

B. Performance Metrics

We adopt hypervolume (HV) [36] and the inverted generational distance (IGD) [65] as performance indicators. Given the non-dominated set P_{op} obtained by an algorithm, the definitions of HV and IGD are as follows.

1) Hypervolume (HV):

$$HV\left(P_{op}, r_p\right) = volume\left(\bigcup_{f\in P_{op}} [f_1, r_{p1}] \times \ldots [f_m, r_{pm}]\right), \tag{15}$$

where $r_p = (r_{p1}, \ldots, r_{pm})^T$ is a reference point dominating all the objective vectors in P_{op}. The upper and lower bounds of all non-dominated solutions obtained by the compared algorithms can be used to determine the reference point as $r_m = \max_m + 0.01(\max_m - \min_m)$.

2) Inverted Generational Distance (IGD):

$$IGD\left(P_{op}, R_p^*\right) = \frac{\sum_{v\in R_p^*} d\left(v, P_{op}\right)}{\left|R_p^*\right|}, \tag{16}$$

where R_p^* is a set of reference points sampled from the true PF and $d(v, P_{op})$ is the minimum Euclidean distance from v to P_{op}. In our experiments, 1000 reference points are uniformly sampled from the true PF.

For both HV and IGD, the Wilcoxon rank sum test is used for significance tests at a significance level of 5%: '+', '−' and '≈', means that the solution set obtained by SPGP-SAEA is significantly better than, worse than, or similar to that obtained by the compared algorithm, respectively. The best mean result in each row is highlighted. Fridman's test is used to observe significant differences at the level of 5%. Moreover, the Nemenyi *post-hoc* distinguishes our algorithm from others. '†' represents

[2]Table Sxxx denotes a table with the serial number of Sxxx in the Supplementary material.

the significant difference from our algorithm after the Nemenyi test.

C. Parameter Settings

The number of independent runs is set to 20 in all experiments. To train reasonable surrogate models before the optimization starts, we use $11d-1$ FEs to collect offline samples. We use a batch size of $C=5$ in each iteration and run 24 iterations in total. Thus, the maximum numbers of FEs and training data points (N_{max}) are $11d-1+120$ and $11d-1+25$, respectively. The sensitivity analysis of C can be found in Fig. S10 and N_{max} can be found in Table SIV. We use NSGA-II as the baseline solver. Crossover and mutation probabilities are set to 0.9 and, 0.2 respectively. The experimental results obtained by varying FEs of NSGA-II are summarized in Section III-B of the Supplementary material.

The subspace size (d_s), i.e., the number of selected decision variables for GP modeling and evolutionary optimization, is set to 5, 10, 10, 100 for $d=20$, 40, 80, 200, respectively. Since the number of pseudo-input points M for the SPGP is related to the decision space size and training data size, we set M to a relatively small value, 10. A sensitivity analysis of the subspace size d_s and M are given in Section III-D and Section III-E of the Supplementary material.

Each time when k-means is started, we need to initialize C cluster centers $\boldsymbol{X}_{I_s}^1, \boldsymbol{X}_{I_s}^2, \ldots, \boldsymbol{X}_{I_s}^C$ ($C=5$), which are randomly selected from the union of the non-dominated solutions ($\boldsymbol{X}_{I_s}^*$) optimized by NSGA-II and the evaluated non-dominated solutions ($\boldsymbol{X}_{I_s}^a$). The clustering iteration stops when the predefined maximum number of iterations reaches 100 or the cluster center no longer changes; otherwise, the algorithm re-optimizes the acquisition function.

In the LCB acquisition function, ω is a parameter used to control the exploration strength. Based on the recommendation in [66], ω is set to be 2 in this work. A sensitivity analysis for ω is presented in Fig. S11.

In solution completion described in Section III-E, we only select one global knee point. An analysis of the effect of knee point selection on diversity is presented in Section III-H of the Supplementary material.

D. Ablation Studies

A series of ablation studies are designed on two test suites, DTLZ and UF, on 20, 40, and 80 dimensions to verify the effectiveness of each key part of the SPGP-SAEA.

1) Ablation Study for SPGP Modeling on Random Subspace Selection: To verify the effectiveness of SPGP modeling on the randomly selected subspace (Algorithm 1 lines 8-10), we compare the original SPGP-SAEA with its variant without dimension reduction via random grouping, denoted as Full-SPGP-SAEA. On DTLZ1a, DTLZ2, DTLZ3a and DTLZ5, d_s is set to 12 for $d=20$. On DTLZ5, d_s is set to 12 for $d=40$. The parameter settings for the remaining test instances are the same as in Section IV-C.

Table SVII presents the statistical results (mean IGD values and standard deviation) on the two test suites, DTLZ and UF, with up to 80 dimensions obtained by SPGP-SAEA and Full-SPGP-SAEA. SPGP-SAEA performs better than Full-SPGP-SAEA on 30 of the 51 test instances. The main reason is that dimension reduction reduces the search space. Although the accuracy of modeling in the subspace may be worse than that in the original space, it can also fit the trend of landscape degree and effectively guide the population to search for potential areas. In addition, the acquisition function is easier to solve in low dimensional space, and the subspace changes randomly in each iteration, which can improve the exploration performance. The strategy of SPGP modeling on the random subspace needs to cooperate with the strategy of completion to play a good role. The performance of SPGP-SAEA is worse than that of Full-SPGP-SAEA, mostly at $d=80$, mainly because in relatively high space, excessive space reduction leads to the inability to guide the search effectively. The average runtime and training time of each iteration on DTLZ1 with 20, 40, and 80 dimensions of 20 runs are given in Fig. 6. From Fig. 6, the CPU time of Full-SPGP-SAEA is about four times that of SPGP-SAEA. The above results indicate that GP modeling and evolutionary search on subspaces of the original problem can improve both the performance and computational efficiency of SPGP-SAEA.

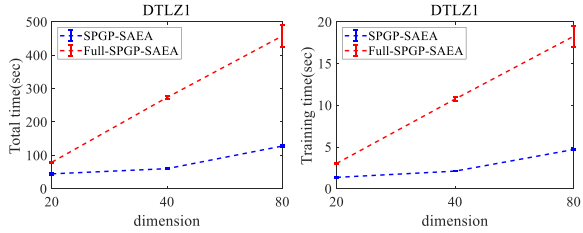


Fig. 6. Total runtime and training time of Full-SPGP-SAEA and SPGP-SAEA on DTLZ1 averaged over 20 independent runs with $d=20$, 40, 80, respectively.

2) Ablation Study on Decision Variable Fill-In Methods: To verify the effectiveness of the proposed decision variable fill-in method that uses the decision variables of the knee point in the archive (Algorithm 4), we compare it with the method using random sampling in the remaining decision space (SPGP-SAEA-R). After the candidate solutions $\boldsymbol{X}_{I_s}^c$ are obtained, we complete each candidate solution with a random value sampled from the remaining decision space $\mathcal{X}_{d_r}$ according to a uniform distribution. We also compare our fill-in method with one randomly selecting a non-dominated solution from the archive (SPGP-SAEA-A). We randomly select non-dominated solutions from the archive $\boldsymbol{X}_{I_s}^a$ and use corresponding decision variables to complete each candidate solution. d_s is set to 5, 10, 10 for $d=20$, 40, 80, respectively. As can be observed from the results in Table SVIII, SPGP-SAEA SPGP is better than SPGP-SAEA-A on 46 out of 51 instances and better than SPGP-SAEA-R on all instances. The proposed completion method based on knee points is more efficient than the others in SPGP-SAEA-A and SPGP-SAEA-R. The reason is straightforward: in a solution set, a bias towards the knee points among the non-dominated solutions is demonstrated in a solution set to approximate the bias towards a large hypervolume [62]. It enhances convergence performance.

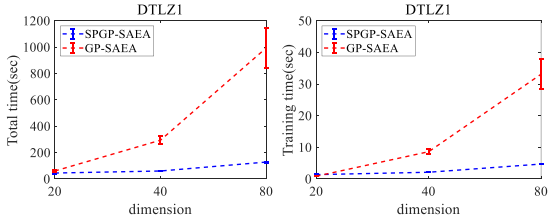


Fig. 7. Total runtime and training time of SPGP-SAEA and GP-SAEA on DTLZ1 averaged over 20 independent runs with $d = 20, 40, 80$, respectively.

3) Ablation Study on Candidate Solutions Selection Methods: To verify the effectiveness of the proposed candidate solutions selection method (Section III-D), we make comparisons with a variant of the proposed algorithm without using the solutions in the archive during the clustering (SPGP-SAEA-NA). As can be seen from the results in Table SVIII, SPGP-SAEA is better than SPGP-SAEA-NA on 37 out of 51 instances, confirming the effectiveness of considering the existing solutions in the archive in candidate solution selection. However, on DTLZ1, DTLZ3, and UF4, which are mostly concave and multimodal, and DTLZ6, which is degenerate, these problems are difficult to converge, especially being scaled up to higher dimensions. The solutions in the archive may be worse than the non-dominated solutions found by optimizing acquisition functions. Therefore, in these problems, considering the solutions in the archive when selecting solutions can reduce the convergence and diversity performance.

4) Ablation Study on SPGP: To verify the performance and computational efficiency of SPGP, we replace SPGP with a standard GP in the proposed optimization framework. The variant using the standard GP is denoted as GP-SAEA. Table SIX of the Supplementary material presents the IGD results of the two algorithms averaged over 20 runs for $d = 20, 40, 80$, respectively. Since SPGP is an approximation of GP, the approximation error of SPGP on high-dimensional problems may be greater than that of GP. This might enables SPGP to smooth out the landscape so that SPGP-assisted search may converge faster than GP-assisted search [2], [64], [57]. Fig. 7 shows the runtime of the two algorithms and the time for updating the models at each iteration on DTLZ1 with 20, 40, and 80 dimensions, averaged over 20 runs.

From Table SIX, the proposed SPGP-SAEA has obtained better results on 23 out of 51 cases and outperformed GP-SAEA on only one test instance.

From Fig. 7, it can be seen that the runtime of SPGP-SAEA increases almost linearly with the problem dimension, while the runtime of GP-SAEA increases more dramatically. Specifically, when the dimension increases to 80, the runtime of SPGP-SAEA and GP-SAEA are $1.28E+02\,s$ and $9.93E+02\,s$, respectively. From the above results, SPGP is computationally more scalable than the original GP.

E. Comparisons With Peers

To assess the performance of SPGP-SAEA, we compare it with eight representative peer algorithms, including three representative multi-objective SAEAs, i.e., MOEA/D-EGO [43], USeMO [59] and KRVEA [44]; four recently proposed large-scale multi-objective SAEAs, i.e., HeE-MOEA [19], EDN-ARMOEA [20], ReMO [67], [68], MCEAD [69], and a baseline algorithm, NSGA-II [31]. All compared algorithms are implemented on MATLAB R2019a and run a PC with Intel Core i7, 3.4-GHz CPU, and Microsoft Windows 10 operating system. MOEA/D-EGO and KRVEA are implemented in the PlatEMO toolbox [70], and the parameters are adapted to high-dimensional problems. For DTLZ1a, DTLZ3a and DTLZ5, d_s is set to 12 for $d = 20$. In particular, to further test the efficiency of SPGP-SAEA in high-dimensional space, we set $d = 200$ to compare SPGP-SAEA with EDN-ARMOEA on DTLZ and UF. The experimental results are presented in Section IV-P of the Supplementary material. In addition, more tests on HeE-MOEA and EDN-ARMOEA are presented in Section IV-P of the Supplementary material. To fully illustrate the competitiveness of our algorithm, we also compared our algorithm with ReMO and MCEAD according to the experimental setup of MCEAD. We take the settings in [68] for implementing of ReMO and the surrogate model. A more detailed description of the experimental setup and results are presented in Section III-Q of the Supplementary material. Note that the application of crude oil distillation units requires simulations on commercial package Aspen HYSYS [71].

About DTLZ, Table I summarizes the IGD results of the first five algorithms under comparison with $d = 20, 40, 80$, respectively, and the results of NSGA-II are summarized in Table SXII, the results of ReMO and MCEAD are summarized in Table SXV. As shown in Table I, SPGP-SAEA outperforms MOEA/D-EGO, KRVEA, UseMO, HeE-MOEA and EDN-ARMOEA on 22, 24, 24, 29 and 27 out of 30 test instances, respectively. Moreover, based on the Friedman test and Nemenyi post-hoc test, KRVEA, UseMO, HeE-MOEA, and EDN-ARMOEA are significantly different from our algorithms, respectively. However, MOEA/D-EGO is similar to our algorithm. From Table SXII, SPGP-SAEA outperforms NSGA-II on 27 out of 30 test instances. Table SXV summarizes that SPGP-SAEA outperforms ReMO and MCEAD on 22 and 22 out of 30 test instances, respectively.

For DTLZ1 and DTLZ3 at dimensions $d = 40$ and $d = 80$, MOEA/D-EGO is better than SPGP-SAEA, which can be attributed to the fact that DTLZ1 and DTLZ3 are multimodal, containing multiple local optima that hinder solution convergence and are difficult to model. MOEA/D-EGO constructs multiple local GP models for each objective function, improving model accuracy. In contrast, SPGP-SAEA builds SPGP models for each objective function on randomly selected subspaces. The former modeling method is more effective in guiding the population search for potential areas. Except for MOEA/D-EGO, SPGP-SAEA outperforms the other algorithms significantly on DTLZ1 and DTLZ3. Compared to DTLZ1, DTLZ1a is smoother. Similarly, compared to DTLZ3, DTLZ3a is smoother. SPGP-SAEA can achieve the best performance thanks to the modeling strategy of SPGP in a random subspace, which reduces search space and enhances convergence performance.

TABLE I
IGD VALUES OBTAINED BY MOEA/D-EGO, KRVEA, USEMO, HEE-MOEA, EDN-ARMOEA, SPGP-SAEA

Problem	d	MOEA/D-EGO	KRVEA	UseMO	HeE-MOEA	EDN-ARMOEA	SPGP-SAEA
DTLZ1	20	2.1320E+2(2.48E+1) ≈	3.1608E+2(5.01E+1) +	3.3699E+2(4.43E+1) +	3.5320E+2(3.83E+1) +	3.4391E+2(3.96E+1) +	2.1374E+2(5.27E+1)
	40	4.8989E+2(3.24E+1) −	8.8395E+2(4.28E-1) +	9.2363E+2(4.51E+1) +	8.9024E+2(5.55E+1) +	8.6622E+2(3.16E+1) +	6.3996E+2(8.72E+1)
	80	1.0231E+3(8.73E+1) −	1.9637E+3(9.22E+1) +	2.0459E+3(8.76E+1) +	2.0188E+3(8.35E+1) +	1.9581E+3(7.73E+1) +	1.5520E+3(1.11E+1)
DTLZ1a	20	2.4850E+2(3.45E+1) +	1.2164E+2(3.29E+1) +	2.7107E+2(5.97E+1) +	5.3726E+1(1.74E+1) +	3.1859E+2(4.25E+1) +	3.5327E+1(1.04E+1)
	40	6.6200E+2(1.43E+2) +	8.4958E+2(4.76E+1) +	8.6992E+2(4.91E+1) +	3.0389E+2(4.77E+1) +	8.5530E+2(4.29E+1) +	9.1678E+1(2.01E+1)
	80	1.4023E+3(1.71E+2) +	1.9669E+3(7.99E+1) +	1.9999E+3(6.20E+1) +	9.7200E+2(1.74E+2) +	1.9767E+3(6.87E+1) +	3.9847E+2(7.81E+1)
DTLZ2	20	7.8780E-1(5.69E-2) +	7.6850E-1(9.18E-2) +	9.0890E-1(3.74E-2) +	3.7140E-1(3.31E-2) ≈	8.5600E-1(4.72E-2) +	3.5450E-1(6.18E-2)
	40	1.6360E+0(2.56E-1) +	2.1027E+0(8.98E-2) +	2.1206E+0(8.90E-2) +	7.9750E-1(5.92E-2) +	2.0782E+0(1.26E-1) +	4.9450E-1(9.81E-2)
	80	2.7968E+0(0.47E-1) +	4.7891E+0(1.31E-1) +	4.8121E+0(1.45E-1) +	1.9735E+0(2.93E-1) +	4.7767E+0(1.31E-1) +	1.3019E+0(3.29E-1)
DTLZ3	20	6.1560E+2(1.21E+2) +	9.6875E+2(1.16E+2) +	8.1840E+2(1.16E+2) +	1.0287E+3(1.39E+2) +	1.0295E+3(9.92E+1) +	5.3527E+2(8.75E+1)
	40	1.5566E+3(2.07E+2) −	2.7014E+3(2.03E+2) +	2.4512E+3(1.63E+2) +	2.7945E+3(1.40E+2) +	2.7435E+3(1.54E+2) +	1.8931E+3(2.71E+2)
	80	3.7167E+3(4.82E+2) −	6.4174E+3(2.31E+2) +	6.1254E+3(2.53E+2) +	6.3925E+3(2.25E+2) +	6.3990E+3(2.26E+2) +	5.4025E+3(3.62E+2)
DTLZ3a	20	8.5175E+2(2.01E+2) +	8.3908E+2(2.24E+2) +	1.0562E+3(1.25E+2) +	1.7283E+2(4.08E+1) +	9.9193E+2(1.22E+2) +	1.4648E+2(4.80E+1)
	40	1.9431E+3(2.69E+2) +	2.6791E+3(1.73E+2) +	2.7405E+3(1.20E+2) +	9.8240E+2(1.44E+2) +	2.7274E+3(1.26E+2) +	4.3110E+2(1.76E+2)
	80	4.3209E+3(6.47E+2) +	6.3847E+3(1.35E+2) +	6.3963E+3(1.43E+2) +	3.0788E+3(4.55E+2) +	6.3955E+3(1.42E+2) +	1.3540E+3(4.13E+2)
DTLZ4	20	1.0755E+0(7.45E-2) +	1.0297E+0(1.68E-1) +	1.2366E+0(8.51E-2) +	9.4410E-1(4.07E-2) +	9.8330E-1(1.11E-1) +	5.8100E-1(9.40E-2)
	40	1.9378E+0(2.01E-1) +	2.3422E+0(1.76E-1) +	2.4849E+0(1.31E-1) +	1.2093E+0(6.98E-2) +	2.3407E+0(1.85E-1) +	9.1520E-1(1.06E-1)
	80	3.2713E+0(5.25E-1) +	5.0746E+0(1.96E-1) +	5.1340E+0(1.86E-1) +	2.5084E+0(2.31E-1) +	5.0827E+0(1.66E-1) +	1.1493E+0(1.03E-1)
DTLZ4a	20	8.9480E-1(6.36E-2) +	8.8877E-1(7.44E-2) +	1.0251E+0(6.44E-2) +	8.5000E-1(6.41E-2) +	8.9220E-1(5.90E-2) +	5.6210E-1(7.80E-2)
	40	1.8582E+0(1.94E-1) +	2.2113E+0(1.09E-1) +	2.2682E+0(1.17E-1) +	1.1975E+0(7.67E-2) +	2.1916E+0(1.12E-1) +	6.6581E-1(1.18E-1)
	80	2.9924E+0(4.81E-1) +	4.9475E+0(1.23E-1) +	4.9731E+0(1.32E-1) +	2.4574E+0(2.00E-1) +	4.9223E+0(1.29E-1) +	1.0157E+0(1.09E-1)
DTLZ5	20	6.7500E-1(8.47E-2) +	6.6000E-1(1.27E-1) +	8.2930E-1(4.59E-2) +	2.3900E-1(9.47E-2) +	7.7759E-1(5.47E-2) +	1.8810E-1(4.71E-2)
	40	1.4415E+0(2.81E-1) +	2.0198E+0(1.02E-1) +	2.0490E+0(1.06E-1) +	6.4030E-1(6.77E-2) +	2.0167E+0(1.16E-1) +	4.4140E-1(1.34E-1)
	80	2.9394E+0(3.64E-1) +	4.6580E+0(1.51E-1) +	4.7521E+0(1.64E-1) +	2.0205E+0(2.27E-1) +	4.7286E+0(1.56E-1) +	1.1859E+0(3.48E-1)
DTLZ6	20	8.5008E+0(1.14E+0) −	1.1495E+1(7.19E-1) −	1.1041E+1(8.83E-1) −	1.4784E+1(5.77E-1) +	1.3980E+1(5.00E-1) ≈	1.4585E+1(4.93E-1)
	40	2.2062E+1(2.17E+0) −	3.0056E+1(7.94E-1) −	2.8429E+1(5.15E-1) −	3.2301E+1(7.14E-1) +	3.1299E+1(6.12E-1) ≈	3.1440E+1(9.49E-1)
	80	4.6359E+1(4.01E+0) −	6.5689E+1(6.81E-1) −	6.3441E+1(9.40E-1) −	6.7540E+1(6.86E-1) +	6.6605E+1(4.51E-1) ≈	6.6573E+1(1.04E+0)
DTLZ7	20	5.8987E+0(9.23E-1) +	3.0720E-1(5.76E-2) −	1.1591E+0(1.89E-1) −	4.3057E+0(7.03E-1) +	6.1016E+0(6.91E-1) +	2.8279E+0(1.06E+0)
	40	8.3063E+0(6.68E-1) +	7.6210E-1(1.65E-1) −	3.4023E+0(3.53E-1) −	6.7107E+0(5.31E-1) +	7.9400E+0(4.55E-1) +	4.8052E+0(1.08E+0)
	80	9.5221E+0(4.68E-1) +	2.2383E+0(3.25E-1) −	5.6277E+0(2.53E-1) −	8.5819E+0(4.81E-1) +	9.3009E+0(2.90E-1) +	7.9801E+0(1.06E+0)
Average ranking		3	3.6667†	4.8†	3.2333†	4.6†	1.7
+ / − / ≈		22/7/1	24/6/0	24/6/0	29/0/1	27/0/3	

DTLZ2, DTLZ4, and DTLZ4a have relatively simple landscape characteristics, making it easier for an MOEA to converge to the Pareto front. However, achieving good performance on these problems requires a strong focus on diversity within the algorithm. SPGP-SAEA excels in these scenarios because it employs k-means to select candidate solutions, thus maintaining a high solution diversity. Moreover, it considers archive information during clustering to further enhance solution diversity. Consequently, the strategy of selection of the candidate solutions can effectively choose well-distributed solutions.

DTLZ5 and DTLZ6 have degenerate PFs. Additionally, the PF of DTLZ6 is discontinuous. SPGP-SAEA achieves the best performance on DTLZ5. However, on DTLZ6, SPGP-SAEA falls short compared to MOEA/D-EGO and KRVEA. Because MOEA/D, utilized in MOEA/D-EGO, and RVEA, used in KRVEA, exhibit stronger search capabilities than NSGA-II in SPGP-SAEA when dealing with discontinuous problems. DTLZ7 has a discontinuous PF, and KRVEA outperforms SPGP-SAEA because KRVEA uses RVEA and a solution selection method based on reference vectors, which improves diversity performance.

From Table SXI, SPGP-SAEA outperforms MOEA/D-EGO, KRVEA, UseMO, HeE-MOEA and EDN-ARMOEA on 19, 17, 21, 19 and 21 of 21 test instances on UF, respectively. Moreover, based on the Friedman test and Nemenyi post-hoc test, MOEA/D-EGO, KRVEA, UseMO, and EDN-ARMOEA are significantly different from our algorithms, respectively. However, HeE-MOEA is similar to our algorithm. From Table SXII, SPGP-SAEA outperforms NSGA-II on 18 out of 21 test cases. From Table SXV, SPGP-SAEA outperforms ReMO and MCEAD on 19 and 13 test instances, respectively. Our algorithm is less effective for the multimodal problems UF2 and UF3 than HeE-MOEA and MCEAD. The biggest difference lies in the modeling techniques used in these algorithms. HeE-MOEA utilizes an ensemble model, MCEAD employs classifiers, and the surrogate model of SPGP-SAEA is SPGP built on random subspaces. This discrepancy may be the primary reason why SPGP-SAEA performs worse than them. For UF4 with $d = 80$, SPGP-SAEA performs worse than MCEAD. However, apart from the three test cases mentioned above, our algorithm performs relatively well.

We enlarge the decision variable dimension to 200 on DTLZ and UF to compare SPGP-SAEA with EDN-ARMOEA. Table SXIII shows the IGD results, from which we can see that SPGP-SAEA outperforms EDN-ARMOEA on all test instances except for DTLZ6 and DTLZ7. And, SPGP-SAEA is significantly different from EDN-ARMOEA.

In the WFG test suite, we compare SPGP-SAEA with HeE-MOEA and EDN-ARMOEA. As shown in Table SXIV, HeE-MOEA outperforms SPGP-SAEA on WFG4 with $d = 20$,

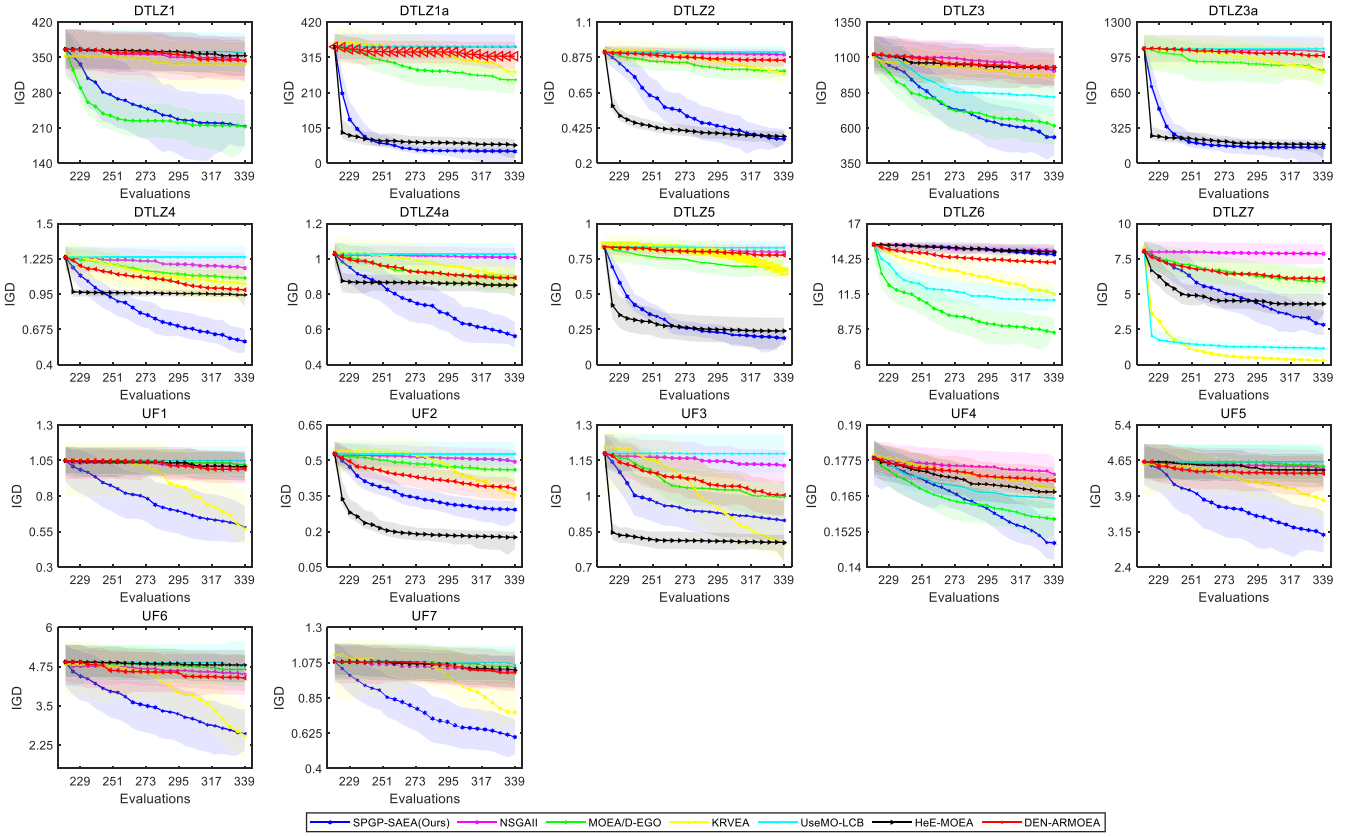


Fig. 8. IGD values over the number of function evaluations on all test suites with a batch size of 5 when $d = 20$.

TABLE II
AVERAGE RUNTIME OF MOEA/D-EGO, KRVEA, USEMO, HEE-MOEA, EDN-ARMOEA AND SPGP-SAEA ON DTLZ1 WITH $d = 20, 40, 80, 200$

	20	40	80	200
MOEA/D-EGO	1.4746E+2(2.36E+0)	8.0137E+2(8.88E+1)	3.4426E+4(1.12E+3)	-
KRVEA	1.3587E+2(5.39E+0)	1.7319E+3(5.93E+1)	1.7071E+4(5.16E+2)	-
UseMO	2.0732E+2(1.15E+1)	1.2157E+3(2.95E+1)	1.3729E+4(6.97E+2)	-
HeE-MOEA	1.2298E+2(6.76E+0)	5.1591E+2(1.50E+1)	7.0087E+3(2.80E+2)	-
EDN-ARMOEA	1.2829E+2(1.77E+0)	2.5177E+2(4.30E+0)	2.7735E+2(7.45E-1)	7.8665E+2(3.95E+0)
SPGP-SAEA	4.4604E+1(3.92E+0)	6.0156E+1(1.17E+0)	1.2789E+2(3.28E+0)	3.9678E+2(5.10E+1)

WFG6 with $d = 80$, as well as WFG8 with $d = 20$, 40, and 80. EDN-ARMOEA outperformed SPGP-SAEA only on WFG4. However, aside from the test cases mentioned above, SPGP-SAEA achieved the best performance.

Fig. 8 shows the mean IGD of each compared algorithm over the number of FEs for $d = 20$. It can be seen that the mean IGD of SPGP-SAEA converges much faster than other algorithms on almost all problems except for DTLZ1, DTLZ6, DTLZ7, UF2 and UF3, indicating its fast convergence.

The runtime of each compared algorithm on DTLZ1 with $d = 20, 40, 80, 200$ is shown in Table II, where the proposed SPGP-SAEA is still the fastest in the category of the regression model as a surrogate model. By contrast, MOEA/D-EGO, KRVEA, and UseMO are relatively time-consuming due to the expensive training costs of GP; HeE-MOEA adopts feature selection to increase the prediction diversity, leading to high computational costs.

F. Discussion

We compare the proposed algorithm with three variants of representative existing multi-objective SAEAs, i.e., MOEA/D-EGO, KRVEA and UseMO, in which we replace the conventional GP with the sparse GP. These variants are called MOEA/D-EGO (SPGP), KRVEA (SPGP) and UseMO (SPGP). The parameter settings of these algorithms are the same as those in Section IV-C. The experimental results are presented in Table SXVI. As can be seen from Table SXVI, SPGP-SAEA outperforms MOEA/D-EGO (SPGP), KRVEA (SPGP) and UseMO (SPGP) on 39, 44, 45 out of 51 test instances, respectively. These results confirm the importance of developing appropriate model management strategies to ensure sparse GPs to guide the search in surrogate-assisted optimization effectively.

From comparative results (regarding the IGD) in Table SXVI and ablation studies, we can see that an existing GP-assisted evolutionary algorithm may fail to work properly if we replace

the standard GP with the sparse GP to reduce the computational complexity, which might be attributed to the fact that neither the standard GP nor SPGP can properly estimate the level of uncertainty for its predictions for high-dimensional problems, making the acquisition functions less effective in model management. Research findings in [17], [72] have confirmed the effectiveness of dimension reduction in BO. In the proposed algorithm, we take advantage of the random grouping technique, which has shown to be powerful for dimension reduction in both surrogate-assisted high-dimensional optimization [28], cooperative co-evolutionary optimization of large-scale optimization [23]. In addition, the proposed algorithm selects the knee point to generate complete solutions since it has been shown that knee solution usually contributes to a better HV performance [62]. Ablation studies have confirmed the effectiveness of selecting the knee point. In most existing GP-assisted optimization algorithms, only the diversity in the current population is considered when performing clustering to select candidate solutions to be sampled. By contrast, the proposed algorithm includes the solutions in the archive into the non-dominated solutions in the current population. Then, it performs clustering, which can effectively enhance the diversity of the solutions of the entire solution set while striking a balance between exploration and exploitation, which our ablation studies have confirmed.

However, our empirical results on the COCO test suite [73] presented in Section IV-R of the Supplementary material also indicate that the proposed algorithm becomes less effective on ill-conditioned test instances. In addition, the proposed strategy for solution completion based on the knee point may fail to work when the objectives do not present a clear tradeoff relationship during the early stage of the search.

V. A Real-World Application

Crude oil distillation is an essential process in the oil refining industry. The design of the operating parameters of the distillation unit is a challenging task. We can adjust the operating parameters in the atmospheric distillation system to maximize the product profit. However, it also affects the heat recovery effect of the heat exchanger network. Therefore, optimizing the operating parameters of the atmospheric distillation unit and heat exchanger network is a typical MOP. The designed optimizer aims to tune the operating parameters involved in each process to meet the given requirements. The problem can be formulated as follows:

$$\begin{aligned}
&\max f_1 \\
&= \sum_{i=1}^{4} Pr_{prod,i} Fr_{prod,i} - Pr_{co} Fr_{co} - Pr_{ss} \sum_{j=1}^{3} Fr_{ss,j} \\
&\min f_2 = Pr_{fuel} U_{fuel} + Pr_{cw} U_{cw} \\
&\text{s.t. } T95_i^{lb} \le T95_i \le T95_i^{up} \\
&CC(Fr_{prod}, Fr_{ss}, Fr_{PA}, Q_{PA}, T_f) = 1
\end{aligned} \tag{17}$$

where Pr and Fr represent prices and product flow rates, respectively; subscripts $prod$ represents products, co represents crude oil, ss represents stripping steam, cw represents cooling water; U represents the lower bound of each indicator, PA represents pump-around; The grand composite curve (GCC) is used to compute $U_{fuel} = GCC(Fr_{PA}, Fr_{ss}, Q_{PA}, T_f, Fr_{prod})$ and $U_{cw} = GCC(Fr_{PA}, Fr_{ss}, Q_{PA}, T_f, Fr_{prod})$; $T95$ represents the temperature constraint of the product; $CC(\cdot) = 1$ indicates that the system specifications must satisfy the convergence criterion. f_1 and f_2 are the product profit and energy consumption, respectively. More information is provided in Section IV of the Supplementary material.

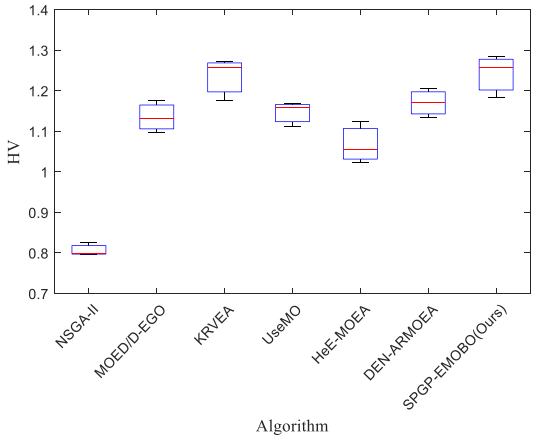


Fig. 9. Boxplot of HV indicator for each algorithm on a operational optimization problem of CODU.

SPGP-SAEA and the other six algorithms under comparison are applied to the operational optimization problem of CODU, which aims to improve performance and energy efficiency by assigning operational conditions. Fig. 9 shows the HV values achieved by each algorithm after 273 evaluations or maximum time budgets. Since the simulation of the problem is computationally very intensive, each algorithm has run independently three times. From these results, we see that the performance of SPGP-SAEA is comparable with KRVEA and significantly better than NSGA-II.

VI. Conclusion

We propose to replace the Gaussian process in SAEAs with a sparse Gaussian process model to reduce the computational complexity so that SAEAs can be scaled to medium-scale EMOPs. To make this possible, the sparse Gaussian process models are built with random grouping technology, and a new sampling strategy is suggested, together with a solution completion method based on knee solutions. Empirical results on two suites of multi-objective benchmark problems with up to 80 design variables and an operational optimization problem of CODU demonstrate the promising scalability of the proposed algorithm in comparison with several state-of-the-art algorithms.

Although the proposed algorithm has proven effective in solving medium-scale EMOPs, we have also observed that sparse Gaussian process models tend to underestimate the variance of their predictions, impacting the effectiveness of the acquisition function. Therefore, it is necessary to design new acquisition

functions based on the characteristics of the posterior distribution of sparse Gaussian processes. Additionally, it is worthwhile to investigate suitable subspace selection methods based on the properties of the problem. Moreover, the knee points of some MOPs are challenging to calculate or locate [74]. Further work exploring more effective methods for identifying knee points and designing more efficient solution completion methods is highly desirable. we will solve more complex optimization problems in CODU, including more complex linkage between variables and the objective functions [40]. Finally, improving the training efficiency of the sparse Gaussian process model itself, which consumes most of the computational resources in the proposed algorithm, will be a focus of interest.

REFERENCES

[1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Hoboken, NJ, USA: Wiley, 2001.

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

[3] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "Handling computationally expensive multiobjective optimization problems with evolutionary algorithms-a survey," *Soft Comput.*, vol. 23, pp. 3137–3166, 2019.

[4] Y. Liu, J. Liu, Y. Jin, F. Li, and T. Zheng, "A surrogate-assisted two-stage differential evolution for expensive constrained optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 3, pp. 715–730, Jun. 2023.

[5] W. Luo, R. Yi, B. Yang, and P. Xu, "Surrogate-assisted evolutionary framework for data-driven dynamic optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 2, pp. 137–150, Apr. 2019.

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

[7] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016.

[8] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, "Recent advances in Bayesian optimization," *ACM Comput. Surv.*, vol. 55, 2023, Art. no. 287. [Online]. Available: https://dl.acm.org/doi/10.1145/3582078

[9] G. Sun, L. Li, J. Fang, and Q. Li, "On lower confidence bound improvement matrix-based approaches for multiobjective Bayesian optimization and its applications to thin-walled structures," *Thin-Walled Struct.*, vol. 161, 2021, Art. no. 107248.

[10] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 1015–1022.

[11] C. Chevalier and D. Ginsbourger, "Fast computation of the multi-points expected improvement with applications in batch selection," in *Proc. Int. Conf. Learn. Intell. Optim.*, 2013, pp. 59–69.

[12] I. Couckuyt, D. Deschrijver, and T. Dhaene, "Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization," *J. Glob. Optim.*, vol. 60, no. 3, pp. 575–594, 2014.

[13] P. Hennig and C. J. Schuler, "Entropy search for information-efficient global optimization," *J. Mach. Learn. Res.*, vol. 13, no. 6, pp. 1809–1837, 2012.

[14] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani, "Predictive entropy search for efficient global optimization of black-box functions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 918–926.

[15] Z. Wang and S. Jegelka, "Max-value entropy search for efficient Bayesian optimization," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3627–3635.

[16] S. Qin, C. Sun, Y. Jin, and G. Zhang, "Bayesian approaches to surrogate-assisted evolutionary multi-objective optimization: A comparative study," in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2019, pp. 2074–2080.

[17] K. Kandasamy, J. Schneider, and B. Póczos, "High dimensional Bayesian optimisation and bandits via additive models," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 295–304.

[18] D. Eriksson, M. Pearce, J. R. Gardner, R. Turner, and M. Poloczek, "Scalable global optimization via local Bayesian optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5497–5508.

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

[20] D. Guo, X. Wang, K. Gao, Y. Jin, J. Ding, and T. Chai, "Evolutionary optimization of high-dimensional multiobjective and many-objective expensive problems assisted by a dropout neural network," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 4, pp. 2084–2097, Apr. 2022.

[21] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When Gaussian process meets Big Data: A review of scalable GPs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4405–4423, Nov. 2020.

[22] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1257–1264.

[23] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 210–224, Apr. 2012.

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

[25] D. Zhan and H. Xing, "A fast kriging-assisted evolutionary algorithm based on incremental learning," *IEEE Trans. Evol. Comput.*, vol. 25, no. 5, pp. 941–955, Oct. 2021.

[26] D. Guo, Z. Ren, Y. Liang, and A. Chen, "Scaling up radial basis function for high-dimensional expensive optimization using random projection," in *Proc. IEEE Congr. Evol. Comput.*, 2020, pp. 1–8.

[27] C. Li, S. Gupta, S. Rana, V. Nguyen, S. Venkatesh, and A. Shilton, "High dimensional Bayesian optimization using dropout," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 2096–2102.

[28] G. Fu, C. Sun, Y. Tan, G. Zhang, and Y. Jin, "A surrogate-assisted evolutionary algorithm with random feature selection for large-scale expensive problems," in *Proc. Int. Conf. Parallel Problem Solving from Nature*, 2020, pp. 125–139.

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

[30] Y. Liu, J. Liu, and Y. Jin, "Surrogate-assisted multipopulation particle swarm optimizer for high-dimensional expensive optimization," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 7, pp. 4671–4684, Jul. 2022.

[31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[32] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. 5th Conf. Evol. Methods Des. Optim. Control Appl. Ind. Problems*, 2001, pp. 95–100.

[33] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[34] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.

[35] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. Int. Conf. Parallel Problem Solving from Nature*, 2004, pp. 832–842.

[36] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[37] V. A. Shim, K. C. Tan, and K. K. Tan, "A hybrid adaptive evolutionary algorithm in the domination-based and decomposition-based frameworks of multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, 2012, pp. 1–8.

[38] Y. Xu, O. Ding, R. Qu, and K. Li, "Hybrid multi-objective evolutionary algorithms based on decomposition for wireless sensor network coverage optimization," *Appl. Soft Comput.*, vol. 68, pp. 268–282, 2018.

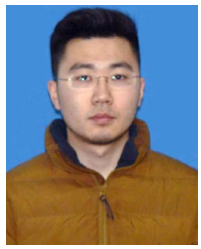
[39] Y. Tian et al., "Evolutionary large-scale multi-objective optimization: A survey," *J. ACM*, vol. 1, no. 1, pp. 1–34, 2021.

[40] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "Test problems for large-scale multiobjective and many-objective optimization," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4108–4121, Dec. 2017.

[41] Y. Tian, S. Yang, L. Zhang, F. Duan, and X. Zhang, "A surrogate-assisted multiobjective evolutionary algorithm for large-scale task-oriented pattern mining," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 2, pp. 106–116, Apr. 2019.

[42] J. Knowles, "ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 50–66, Feb. 2006.

[43] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, Jun. 2010.
[44] 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 Trans. Evol. Comput.*, vol. 22, no. 1, pp. 129–142, Feb. 2018.
[45] Q. Lin, X. Wu, L. Ma, J. Li, M. Gong, and C. A. C. Coello, "An ensemble surrogate-based framework for expensive multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 4, pp. 631–645, Aug. 2022.
[46] Z. Song, H. Wang, C. He, and Y. Jin, "A kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 6, pp. 1013–1027, Dec. 2021.
[47] H. Hao, A. Zhou, H. Qian, and H. Zhang, "Expensive multiobjective optimization by relation learning and prediction," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 1157–1170, Oct. 2022.
[48] J. Lin, C. He, and R. Cheng, "Adaptive dropout for high-dimensional expensive multiobjective optimization," *Complex Intell. Syst.*, vol. 8, pp. 1–15, 2021.
[49] M. Seeger, "Gaussian processes for machine learning," *Int. J. Neural Syst.*, vol. 14, no. 2, pp. 69–106, 2004.
[50] M. Ali and L. Liu, "Light-weight pointcloud representation with sparse Gaussian process," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 4931–4937.
[51] R. Cheng et al., "A benchmark test suite for evolutionary many-objective optimization," *Complex Intell. Syst.*, vol. 3, no. 1, pp. 67–81, 2017.
[52] B. He, S. Zhang, F. Yang, C. Yan, D. Zhou, and X. Zeng, "An efficient Bayesian optimization approach for analog circuit synthesis via sparse Gaussian process modeling," in *Proc. IEEE Des., Automat. Test Eur. Conf. Exhib.*, 2020, pp. 67–72.
[53] M. Mlakar, D. Petelin, T. Tušar, and B. Filipič, "GP-DEMO: Differential evolution for multiobjective optimization based on Gaussian process models," *Eur. J. Oper. Res.*, vol. 243, no. 2, pp. 347–361, 2015.
[54] M. McIntire, D. Ratner, and S. Ermon, "Sparse Gaussian processes for Bayesian optimization," in *Proc. Conf. Uncertainty Artif. Intell.*, 2016, pp. 517–526.
[55] A. Yang, C. Li, S. Rana, S. Gupta, and S. Venkatesh, "Sparse spectrum gaussian process for Bayesian optimization," in *Proc. Adv. Knowl. Discov. Data Mining - 25th Pacific-Asia Conf.*, Part II, in Lecture Notes in Computer Science, vol. 12713, May 11–14, 2021, pp. 257–268.
[56] Y. Jin, M. Hüsken, M. Olhofer, and B. Sendhoff, "Neural networks for fitness approximation in evolutionary optimization," in *Knowledge Incorporation in Evolutionary Computation*. New York, NY, USA: Springer, 2005, pp. 281–306.
[57] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 329–355, Jun. 2010.
[58] R. Sheikholeslami and S. Razavi, "Progressive latin hypercube sampling: An efficient approach for robust sampling-based analysis of environmental models," *Environ. Modelling Softw.*, vol. 93, pp. 109–126, 2017.
[59] S. Belakaria, A. Deshwal, N. K. Jayakodi, and J. R. Doppa, "Uncertainty-aware search framework for multi-objective bayesian optimization," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 10044–10052.
[60] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 29–38, Feb. 2006.
[61] X. Lin, H.-L. Zhen, Z. Li, Q. Zhang, and S. Kwong, "A batched scalable multi-objective Bayesian optimization algorithm," 2018, *arXiv:1811.01323*.
[62] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 761–776, Dec. 2015.
[63] Q. Zhang et al., "Multiobjective optimization test instances for the CEC 2009 special session and competition," Univ., Essex, Colchester, U.K., and Nanyang Technological Univ., Singapore, Tech. Rep. CES-487, 2008.
[64] C. Yang, J. Ding, Y. Jin, and T. Chai, "Offline data-driven multiobjective optimization: Knowledge transfer between surrogates and generation of final solutions," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 409–423, Jun. 2020.
[65] C. A. C. Coello et al., *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5. New York, NY, USA: Springer, 2007.
[66] M. T. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 421–439, Aug. 2006.
[67] H. Qian and Y. Yu, "Solving high-dimensional multi-objective optimization problems with low effective dimensions," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 875–881.
[68] H. Wang, H. Xu, Y. Yuan, and Z. Zhang, "An adaptive batch Bayesian optimization approach for expensive multi-objective problems," *Inf. Sci.*, vol. 611, pp. 446–463, 2022.
[69] T. Sonoda and M. Nakata, "Multiple classifiers-assisted evolutionary algorithm based on decomposition for high-dimensional multi-objective problems," *IEEE Trans. Evol. Comput.*, vol. 26, no. 6, pp. 1581–1595, Dec. 2022.
[70] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017.
[71] J. Haydary, *Chemical Process Design and Simulation: Aspen Plus and Aspen Hysys Applications*. Hoboken, NJ, USA: Wiley, 2019.
[72] N. Jaquier and L. Rozo, "High-dimensional Bayesian optimization via nested Riemannian manifolds," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 20939–20951.
[73] D. Brockhoff, A. Auger, N. Hansen, and T. Tušar, "Using well-understood single-objective functions in multiobjective black-box optimization test suites," *Evol. Comput.*, vol. 30, no. 2, pp. 165–193, 2022.
[74] G. Yu, L. Ma, Y. Jin, W. Du, Q. Liu, and H. Zhang, "A survey on knee-oriented multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 6, pp. 1452–1472, Dec. 2022.



Haofeng Wu received the M.S. degree in control engineering from the Shenyang University of Technology, Shenyang, China, in 2017. He is currently working toward the Ph.D. degree with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang. His research interests include surrogate-assisted evolutionary optimization and its applications in industrial processes.

Yaochu Jin (Fellow, IEEE) 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. In 2023, he joined the Westlake University, Hangzhou, as a Chair Professor of AI, leading the Trustworthy and General AI Lab. He was an Alexander von Humboldt Professor of AI with the Faculty of Technology, Bielefeld University, Bielefeld, Germany, from 2021 to 2023, and a Distinguished Chair, Professor of Computational Intelligence with the Department of Computer Science, University of Surrey, Guildford, U.K. He was a Finland Distinguished Professor with the University of Jyväskylä, Jyväskylä, Finland, Changjiang Distinguished Visiting Professor with Northeastern University, Shenyang, China, and Distinguished Visiting Scholar with the University of Technology Sydney, Sydney, NSW, Australia. His main research interests include multi-objective and data-driven evolutionary optimization, evolutionary multi-objective learning, trustworthy AI, and evolutionary developmental AI. Dr. Jin is currently the President-Elect of the IEEE Computational Intelligence Society and an Editor-in-Chief of *Complex & Intelligent Systems*. He was the recipient of the 2018, 2021 and 2023 IEEE Transactions on Evolutionary Computation Outstanding Paper Award, and 2015, 2017, and 2020 IEEE Computational Intelligence Magazine Outstanding Paper Award. He was named by the Clarivate as "a Highly Cited Researcher" from 2019 to 2022 consecutively. He is a Member of Academia Europaea.

Kailai Gao received the M.S. degree in control engineering in 2019 from Northeastern University, Shenyang, China, where she is currently working toward the Ph.D. degree in control theory and control engineering with the State Key Laboratory of Synthetical Automation for Process Industries. Her research interests include evolutionary transfer optimization, bayesian optimization, and their applications in the industrial process.

Jinliang Ding (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in control theory and control engineering from Northeastern University, Shenyang, China, in 2001, 2004, and 2012, respectively. He is currently a Professor with the State Key Laboratory of Synthetical Automation for Process Industry, Northeastern University. He has authored or coauthored more than 200 refereed journal papers and refereed papers at international conferences. He has also invented or co-invented more than 50 patents. His research interests include modeling, plant-wide control and optimization for the complex industrial systems, machine learning, industrial artificial intelligence, and computational intelligence and application. Dr. Ding was the recipient of the Young Scholars Science and Technology Award of China in 2016, National Science Fund for Distinguished Young Scholars in 2015, National Technological Invention Award in 2013, Natural Science Award of Liaoning Province in 2022, and three First-Prize of Science and Technology Awards of the Ministry of Education in 2006, 2012, and 2018, respectively. One of his articles published on Control Engineering Practice was selected for the Best Paper Award of 2011–2013. He was an Associate Editor for IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON EMERGMING TOPICS IN COMPUTATIONAL INTELLIGENCE, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: EXPRESS BRIEFS.

Ran Cheng (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Surrey, Guildford, U.K., in 2016. He is currently an Associate Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. Dr. Cheng was the recipient of the 2018 and 2021 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award, 2019 IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation Award "Nature Inspired Optimization of Large Problems," and 2020 *IEEE Computational Intelligence Magazine* Outstanding Paper Award. He is the Founding Chair of IEEE Symposium on Model-Based Evolutionary Algorithms. He is an Associate Editor for IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, and IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE.