An Ensemble Surrogate-Based Coevolutionary Algorithm for Solving Large-Scale Expensive Optimization Problems

Xunfeng Wu, Qiuzhen Lin, *Member, IEEE*, Jianqiang Li, *Member, IEEE*, Kay Chen Tan, *Fellow, IEEE*, and Victor C. M. Leung, *Life Fellow, IEEE*

***Abstract*—Surrogate-assisted evolutionary algorithms (SAEAs) have shown promising performance for solving expensive optimization problems (EOPs) whose true evaluations are computationally or physically expensive. However, most existing SAEAs only focus on the problems with low dimensionality and they rarely consider solving large-scale EOPs (LSEOPs). To fill this research gap, this article proposes an ensemble surrogate-based coevolutionary optimizer for tackling LSEOPs. First, some local surrogate models are trained with low-dimensional data subsets by using feature selection on the large-scale decision variables, a part of which are used to build a selective ensemble surrogate for better approximating the target LSEOP. Then, a coevolutionary optimizer guided by the ensemble surrogate is designed by running two populations to cooperatively solve the target LSEOP and the simplified auxiliary problem. The information of offspring from the two populations is shared to facilitate the coevolution process, which can exploit the searching experience from the simplified auxiliary problem to help solving the target LSEOP. Finally, an effective infill selection criterion is used to update the ensemble surrogate and enhance its approximate performance. To evaluate the performance of the proposed algorithm, a number of well-known benchmark problems are used and the experimental results validate our superior performance over nine state-of-the-art SAEAs on most cases.**

***Index Terms*—Ensemble surrogate, large-scale expensive optimization problem (LSEOP), surrogate-assisted evolutionary algorithm (SAEA).**

Manuscript received 2 March 2022; revised 14 June 2022; accepted 6 August 2022. Date of publication 16 September 2022; date of current version 18 August 2023. This work was supported in part by the National Key Research and Development Project, Ministry of Science and Technology, China, under Grant 2018AAA0101301; in part by the National Natural Science Foundation of China under Grant 61876162 and Grant 61876110; in part by the Research Grants Council of the Hong Kong, SAR, under Grant PolyU11209219; in part by the Shenzhen Scientific Research and Development Funding Program under Grant JCYJ20190808164211203; in part by the Guangdong "Pearl River Talent Recruitment Program" under Grant 2019ZT08X603; and in part by the Shenzhen Science and Technology Innovation Commission under Grant R2020A045. This article was recommended by Associate Editor H. Ishibuchi. *(Corresponding author: Qiuzhen Lin.)*

Xunfeng Wu, Qiuzhen Lin, Jianqiang Li, and Victor C. M. Leung are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: qiuzhlin@szu.edu.cn).

Kay Chen Tan is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, SAR, China (e-mail: kctan@polyu.edu.hk).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2022.3200517.

Digital Object Identifier 10.1109/TCYB.2022.3200517

I. Introduction

In many real-world applications, it is very often to optimize the single-objective optimization problems with a variety of complex properties, such as nonconvexity, multimodality, and nondifferentiability [1], [2]. Traditional mathematical methods cannot effectively solve these complex optimization problems, while some heuristic algorithms, such as evolutionary algorithm (EA) [3], [4] and particle swarm optimizer (PSO) [5], [6], are more effective and efficient [7]. In general, these heuristic algorithms implicitly assume that the evaluations of candidate solutions to get their objective values and constraints are easy and cheap. However, this assumption cannot always hold true in some real-world scenarios [8], for example, numerical simulations or physical experiments, which are usually computationally or physically expensive for evaluation. This kind of problems is often called expensive optimization problems (EOPs).

As only a limited number of true evaluations for EOPs are available, traditional heuristic algorithms cannot quickly find the satisfactory results. Thus, a new kind of EAs is proposed to solve EOPs, often called surrogate-assisted EAs (SAEAs). Using a small number of evaluated solutions as training samples, SAEAs will train a surrogate model in order to replace the computationally expensive evaluations, which can significantly reduce the computational costs for solving EOPs. In the existing SAEAs, Kriging models (also called Gaussian models) are mostly used as the surrogate models due to their promising performance [9], [10], [11]. The radial basis function (RBF) is another popular surrogate model used in SAEAs [12], [13], [14]. Other than these models, other machine learning models, such as polynomial regression (PR) [15], support vector machine (SVM) [16], random forest [17], and artificial neural network [18], also have been used as the surrogate models in SAEAs.

After determining the surrogate models, selecting solutions to be evaluated in SAEAs is also important, as truly evaluated solutions will be used to retrain the surrogate models, which will result in different approximate performances. Thus, many infill selection criteria are proposed to select the most suitable solutions for true evaluation, which can be mainly classified into three categories [19]: 1) performance-based criteria [20], [21]; 2) uncertainty-based criteria [22], [23]; and 3) their combined methods [24], [25], [26]. To be specific,the performance-based criteria depend on the approximate objective value predicted by the surrogate model [20], [21], while the uncertainty-based criteria focus on the uncertainty information returned from the surrogate model [9], [27]. Thus, the performance-based criteria can provide more useful information to capture the exact optimum position [28], while the uncertainty-based criteria are able to increase the prediction accuracy of surrogate models on some uncertain regions [29]. Their combined methods often try to keep a good balance between global exploration and local exploitation, such as the probability of improvement [25] that estimates the probability of whether the predicted value of new sample is better than the true value of the current optimal solution, lower confidence bound [10] that balances the search between promising areas and less explored areas, and expected improvement [26] that estimates the improvement of objective value by adding a new sample at a certain position.

However, most existing SAEAs only focus on tackling EOPs with no more than 300 dimensions/decision variables, despite the fact that there also exist some large-scale EOPs (LSEOPs) in real-world applications [30], [31], [32], [33] with up to 1000 dimensions. To fill this research gap, this article proposes an ensemble surrogate-based coevolutionary optimizer (ESCO) for tackling LSEOPs. The increased dimension in LSEOPs poses more difficult challenges to SAEAs on three aspects: 1) how to construct an effective surrogate model with satisfactorily good performance to approximate LSEOPs; 2) how to ensure the fast convergence of EAs for solving LSEOPs; and 3) how to effectively exploit the limited number of function evaluations to enhance the prediction accuracy of surrogate models. To solve the above challenges, a selective ensemble surrogate, a coevolutionary optimizer, and an effective infill selection criterion are accordingly proposed in this article.

To summarize, the main contributions of this article are described as follows.

1) A selective ensemble surrogate is proposed in ESCO for replacing the true expensive evaluations of LSEOPs. In this method, several local surrogate models are first built by using feature selection and roulette wheel selection on a large number of decision variables. Then, a subset of them with superior performance is selected to compose a selective ensemble surrogate, which can better approximate the fitness value of LSEOPs.
2) A coevolutionary optimizer is designed by evolving two populations to, respectively, solve the original LSEOP and the simplified auxiliary problem. This coevolutionary process is run by sharing the information of offspring generated from two populations. In this way, the solving of the original LSEOP can exploit the search experience from that of the simplified auxiliary problem.
3) An effective infill selection criterion is proposed to consider both the convergence and diversity of the solutions. Thus, solutions with better fitness values predicted by the surrogate and more diversity on decision space will be added to update our ensemble surrogate, which can further improve its prediction accuracy.

Based on the above three parts, an ESCO is assembled to better solve LSEOPs. In the experimental study, ESCO is first compared with five competitive SAEAs (DDEA-SE [14], MGP-SLPSO [34], SA-COSO [13], CA-LLSO [35], and evolutionary sampling-assisted optimizer (ESAO) [12]) that are designed for solving EOPs with a relatively small number of decision variables. Then, ESCO is further compared with four competitive SAEAs (SRK-DDEA [36], TT-DDEA [37], surrogate-assisted cooperative coevolution (SACC) [38], and SAEA-RFS [39]) that are specifically designed for solving LSEOPs. Finally, the experimental results validate the advantages of ESCO.

The remainder of this article is organized as follows. Section II provides some related works and motivations, while Section III introduces the details of the proposed ESCO. Section IV gives the experimental studies of all the compared SAEAs. Finally, Section V concludes this article and gives some future works.

II. Related Work and Motivations

A. Surrogate-Assisted Evolutionary Algorithms

Most of SAEAs are designed for solving small-scale EOPs, which are briefly introduced according to their used surrogate models.

The Kriging model is the most widely used surrogate model in SAEAs, which are validated to have promising performance for tackling EOPs. In [9], an efficient global optimizer (EGO) is proposed by first applying the Kriging model to optimize four test EOPs with 2, 2, 3, and 6 decision variables. In [10], a Gaussian process-assisted EA (GPEME) for medium-scale EOPs with 20–50 decision variables is proposed, in which a dimension reduction technique is used to map the training data to a lower dimensional space and a new surrogate model-aware search mechanism is proposed to focus on searching promising subregions. In [34], a multiobjective infill criterion-driven Gaussian process-assisted PSO (MGP-SLPSO) is presented to tackle EOPs with up to 100 decision variables and a synthesized real-world problem with 31 decision variables. In this method, a multiobjective infill criterion is designed to optimize two separate objectives, that is, the minimization of the approximated fitness and the maximization of the uncertainty in the surrogate model. Then, a nondominated sorting method [40] is applied to select the solutions for true function evaluation. In [38], an SACC algorithm is designed for solving LSEOPs, in which the Kriging model is used as the surrogate to approximate the low-dimensional subproblems decomposed from the original problem, and then the adaptive differential evolution (DE) algorithm [41] is used to optimize each subproblem.

RBF is another popularly used surrogate model in SAEAs. In [13], a surrogate-assisted cooperative swarm optimizer (SA-COSO) using RBF is proposed for solving EOPs with up to 100 decision variables. In this method, two types of surrogate-assisted PSO (a PSO with a constriction factor [42] and an RBF-assisted social learning-based PSO [43]) are designed to cooperatively search the global optimum of the target EOP. In [12], an effective ESAO is presented to solve EOPs with up to 100 dimensions and an airfoil design problem with 12 dimensions. In ESAO, a global RBF model is used to select the best solution generated by mutation and crossover, and then a nature-inspired optimizer is adopted to search the optimum of the local surrogate model that is trained with a certain number of current best samples. In [14], an offline SAEA with selective surrogate ensembles (DDEA-SE) is designed for solving offline EOPs with up to 100 dimensions, in which a large number of surrogates are trained offline using the bagging technique [44], [45] and a subset of them is adaptively selected for estimating the fitness in the evolutionary process. In [46], the RBF model is used as the surrogate and the SHADE algorithm [47] is used as evolutionary optimizer running in the cooperative coevolution framework of SACC [38], which compose the final SAEA called RBF-SHADE-SACC. In [39], a new SAEA is designed by using the random feature selection method, called SAEA-RFS. In this algorithm, the random feature selection method is used to obtain a number of subproblems and then an evolutionary optimizer assisted by the RBF model is used to optimize these subproblems sequentially. In [36], a stochastic ranking-based data-driven EA (SRK-DDEA) is designed for tackling offline LSEOPs, which proposes a stochastic ranking to manage four RBF models with different kernels. In [37], a new offline data-driven EA using tri-training (TT-DDEA) is presented. In this algorithm, a semisupervised learning method is used to select solutions with high-confidence fitness prediction, which can enrich the training data for surrogate models.

Other machine learning models are also used as the surrogate in SAEAs, such as PR, SVM, and gradient boosting classifier. In a generalizing surrogate single-objective memetic algorithm (GS-SOMA) [15], the PR model is used to successively smooth the problem landscape within the local search phase, which aims to speed up the evolutionary search. In [16], a rank-based SVM surrogate model is integrated with covariance matrix adaptation evolution strategy (CMA-ES), and the resultant SAEA is called ACM-ES. In the classifier-assisted level-based learning swarm optimizer (CA-LLSO) [35], a gradient boosting classifier is proposed in the level-based learning swarm optimizer for solving six commonly used test EMOPs with up to 300 decision variables. In this way, CA-LLSO enhances the robustness and scalability of SAEAs by using the advantages of both the classification model and swarm optimizer.

Moreover, to take advantage of diverse surrogate models, several SAEAs employ multiple surrogates or an ensemble surrogate. In [48], a surrogate-based PSO (SBPSO) is proposed to solve two EOPs and a real case study with two dimensions. SBPSO applies a hybrid surrogate model to approximate the expensive black box by combining the advantages of the traditional optimization method and PSO. In [49], a novel SAEA called SAEA-UGC is reported for handling 20 widely used test EOPs with 30 decision variables. Here, a novel uncertainty grouping-based infill criterion is proposed for global search, while an RBF model that is trained by only a certain amount of current best solutions is adopted for local search.

B. Motivations

The above SAEAs are mostly designed for tackling EOPs with no more than 300 decision variables. However, some EOPs modeled from the real-world applications [30], [31], [32], [33] may have large-scale decision variables, which should be properly addressed. Here, the dimensions of test EOPs used by most existing SAEAs (EGO [9], GPEME [10], MGP-SLPSO [34], SACC [38], SA-COSO [13], ESAO [12], DDEA-SE [14], RBF-SHADE- SACC [46], SAEA-RFS [39], GS-SOMA [15], ACM-ES [16], CA-LLSO [35], SBPSO [48], and SAEA-UGC [49]) are summarized in Table A.I of the supplementary file due to page limitations. From Table A.I of the supplementary file, EGO and SBPSO are proposed for solving EOPs with less than ten dimensions, while GPEME, MGP-SLPSO, SA-COSO, ESAO, DDEA-SE, GS-SOMA, ACM-ES, and SAEA-UGC can well solve EOPs with up to 100 dimensions. Only SACC, SAEA-RFS, SRK-DDEA, and TT-DDEA are able to solve EOPs with 1000 dimensions. Based on our experimental studies in Section IV, most of these existing SAEAs still cannot effectively solve the LSEOPs with up to 1000 dimensions.

Therefore, to fill the research gap, this article aims to design a simple and effective SAEA for solving LSEOPs. A novel selective ensemble surrogate is proposed in this article to predict the approximate objective values of LSEOPs. Some diverse local models are first trained by using different subsets with low-dimensional samples and some of them with better performance are used to compose the final selective ensemble surrogate, which can improve the accuracy for approximating LSEOPs with an acceptable cost of training time. Then, a coevolutionary optimizer is proposed to speed up the convergence when solving LSEOPs. In this procedure, a simplified auxiliary problem is artificially built, which includes only a part of decision variables in the original LSEOP. This way, the experience of searching the simplified auxiliary problem can be used to help solving the original LSEOP during the coevolutionary process by sharing their offspring. Finally, an effective infill selection criterion is proposed to consider both the solutions' convergence and diversity for true evaluations. Thus, some solutions with better approximate objective values and better diversity on decision space will be added to retrain the surrogate, which can further improve its prediction accuracy. With the above three improved components, ESCO can better solve LSEOPs as experimentally verified in Section IV.

III. Details of Our Proposed Algorithm

In this section, the details of ESCO for solving LSEOPs are presented. In Section III-A, the overall framework of ESCO is first introduced, which has four main procedures: 1) initialization; 2) selective ensemble surrogate; 3) surrogate-assisted coevolutionary optimizer (SACO); and 4) infill selection criterion. Then, the details of our selective ensemble surrogate, SACO, and infill selection criterion are, respectively, introduced in the following sections.

A. Overall Framework of ESCO

To introduce the running of ESCO, its overall framework is first given in Algorithm 1. In particular, initialization is run in line 1 to get n_s initial solutions by using the Latin hypercube sampling [50] on the large-scale decision

Algorithm 1 Overall Framework of ESCO

Input: n_s (the number of initial samples), n_{mes} (the total number of true function evaluations)

1: Initialize n_s solutions that are truly evaluated and added into the training dataset $\boldsymbol{TD}$; set $n_{tes} = n_s$
2: **while** $n_s < n_{mes}$ **do**
3:     ($\boldsymbol{LS}$, se) $\leftarrow$ **Selective-Ensemble-Surrogate** ($\boldsymbol{TD}$) **//Algorithm 2**
4:     $\boldsymbol{P} \leftarrow$ **SACO** ($\boldsymbol{TD}$, $\boldsymbol{LS}$, se) **//Algorithm 3**
5:     $s \leftarrow$ **Infill-Selection-Criterion** ($\boldsymbol{TD}$, $\boldsymbol{P}$) **//Algorithm 4**
6:     s is truly evaluated and added into $\boldsymbol{TD}$
7:     Set $n_{tes} = n_{tes} + 1$
8: **end while**

Algorithm 2 Selective-Ensemble-Surrogate ($\boldsymbol{TD}$)

Input: $\boldsymbol{TD}$ (training dataset)

Initialize t, k, and set local models set $\boldsymbol{LS} = \emptyset$
Calculate the contribution ratio r_j ($j = 1, 2, \ldots, d$) of all decision variables to the objective by (1)-(2)
for $i = 1$ to t **do**
    Obtain ind_i from the variable vector by roulette wheel selection
    for each sample s in $\boldsymbol{TD}$ **do**
        if $U(0, 1) < 3/4$ **then**
            Add $s(:, ind_i)$ to $\boldsymbol{S}_i^{train}$
        Else
            Add $s(:, ind_i)$ to $\boldsymbol{S}_i^{test}$
        end if
    end for
    Construct the local model m_i based on $\boldsymbol{S}_i^{train}$
    $\boldsymbol{LS} = \boldsymbol{LS} \cup \{m_i, ind_i\}$
    Predict the objective values Y_p for all samples in $\boldsymbol{S}_i^{test}$
    Calculate the KLD value $KLD(Y_r \mid Y_p)$ from Y_p to their true objective values Y_r by (4)
end for
Sort models in $\boldsymbol{LS}$ in ascending order based on their KLD values
Preserve only the first k models in $\boldsymbol{LS}$
Construct the selective ensemble se using $\boldsymbol{LS}$ by (5)
return $\boldsymbol{LS}$, se

Algorithm 3 SACO ($\boldsymbol{TD}$, $\boldsymbol{LS}$, se)

Input: $\boldsymbol{TD}$ (training dataset), $\boldsymbol{LS}$ (local surrogates), se (selective ensemble)

Initialize n, $g_{\max}$ and set generation counter $g = 0$
$\boldsymbol{P} \leftarrow$ Select the best n samples in $\boldsymbol{TD}$
$[m_c, ind_c] \leftarrow$ Select a local model from $\boldsymbol{LS}$ randomly
$\boldsymbol{P}_a \leftarrow$ Initialize n solutions with the same dimension as ind_c
while $g \leq g_{max}$ **do**
    Generate offspring population $\boldsymbol{O}$ from $\boldsymbol{P}$ and $\boldsymbol{O}_a$ from $\boldsymbol{P}_a$
    Generate another offspring population $\boldsymbol{Q}$, $\boldsymbol{Q}_a$ by swapping ind_c decision variables of $\boldsymbol{O}$ and $\boldsymbol{O}_a$
    $\boldsymbol{U} \leftarrow \boldsymbol{P} \cup \boldsymbol{O} \cup \boldsymbol{Q}$; $\boldsymbol{U}_a \leftarrow \boldsymbol{P}_a \cup \boldsymbol{O}_a \cup \boldsymbol{Q}_a$
    Evaluate $\boldsymbol{U}$ by se and $\boldsymbol{U}_a$ by the local model m_c
    Update $\boldsymbol{P}$ and $\boldsymbol{P}_a$ respectively using $\boldsymbol{U}$ and $\boldsymbol{U}_a$
    $g = g + 1$
end while
return $\boldsymbol{P}$

Algorithm 4 Infill-Solutions-Selection ($\boldsymbol{TD}$, $\boldsymbol{P}$)

Input: $\boldsymbol{TD}$ (training dataset), $\boldsymbol{P}$ (population obtained by Algorithm 3)

$b \leftarrow$ Obtain the best sample in $\boldsymbol{TD}$
Initialize l and set $\boldsymbol{C} = \emptyset$
for each solution s in $\boldsymbol{P}$ **do**
    if $s < b$ **then**
        $d \leftarrow$ Calculate the minimum distance in the decision space between s and all the samples of $\boldsymbol{TD}$
        if $d > l$ **then**
            $\boldsymbol{C} \leftarrow \boldsymbol{C} \cup s$
        end if
    end if
end for
if $C \neq \emptyset$ **then**
    Sort all solutions in $\boldsymbol{C}$ based on their predicted fitness values
    $s \leftarrow$ Select a solution with best predicted value in $\boldsymbol{C}$
Else
    $s \leftarrow$ Select a solution from $\boldsymbol{P}$ randomly
end if
return s

space. Then, the initial solutions are evaluated by the true expensive function and added into training dataset ($\boldsymbol{TD}$) as samples, so the number of true evaluations n_{tes} is set to n_s. After that, the optimization process of ESCO will be iteratively run in lines 3–7, which are briefly introduced as follows.

In line 3, Algorithm 2 is run with the input: $\boldsymbol{TD}$, which will output our constructed selective ensemble surrogate se with its several local surrogates $\boldsymbol{LS}$. The details of Algorithm 2 will be given in Section III-B. After that, Algorithm 3 is run in line 4 with the inputs: $\boldsymbol{TD}$, $\boldsymbol{LS}$, and se, which will coevolve two populations to, respectively, optimize the target LSEOP as guided by the above selective ensemble se and the simplified auxiliary problem as guided by some local surrogates $\boldsymbol{LS}$. Finally, Algorithm 3 will output an evolved population $\boldsymbol{P}$ for the target LSEOP, the solutions of which are not truly evaluated. The details of Algorithm 3 will be given in Section III-C. Finally, Algorithm 4 is run in line 5 with the inputs: $\boldsymbol{TD}$, $\boldsymbol{P}$, to select one promising solution s, which is used for true evaluation and added into $\boldsymbol{TD}$ in line 6 to retrain a selective ensemble se. The details of Algorithm 4 will be given in Section III-D. In line 7, the number of true evaluations n_{tes} is added by 1.

While the maximal number of true evaluations n_{mes} is not reached in line 2, the above coevolutionary process will be iteratively run to optimize the target LSEOP. Otherwise, ESCO will output the found best solution in $\boldsymbol{TD}$.

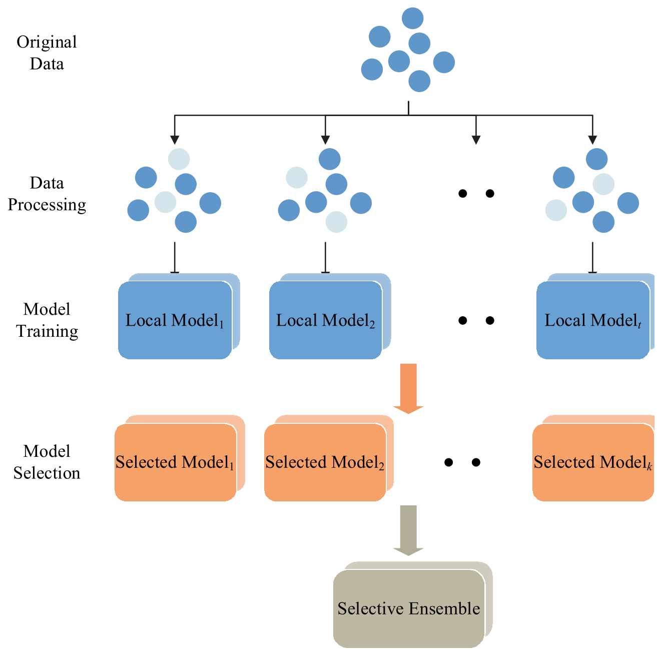


Fig. 1. Diagram of constructing selective ensemble.

B. Selective Ensemble Surrogate

The main idea of selective ensemble surrogate is to generate diverse local surrogates independently and then choose some of them to construct a selective ensemble, so as to improve the quality of the surrogate model used to predict the fitness values of solutions for LSEOPs, as shown in Fig. 1. First, this approach creates different training data subsets by using feature selection and random sampling on the original dataset, which are used to train diverse local surrogates. As the linear combination of highly nonlinear models can dramatically decrease the variance of generation error [51], RBF networks (RBFN) belonging to the nonlinear model are employed as the surrogates.

In order to do feature selection for LSEOPs, the Kendall rank correlation coefficient (KRCC) [52] is applied to measure the correlation between each decision variable and the objective. To introduce the KRCC, assume that the ith decision variable x_i is selected as an example and $\{(x_i^1, y^1), \ldots, (x_i^n, y^n)\}$ denote n samples of x_i and the corresponding objective values in the training dataset $\boldsymbol{TD}$. Then, any pair of two samples (x_i^j, y^j) and $(x_i^k, y^k)$are mutually compared. The pair of samples are said to be concordant when $x_i^j < x_i^k$ & $y^j < y^k$or $x_i^j > x_i^k$ &$y^j > y^k$, while to be discordant when $x_i^j < x_i^k$&$y^j > y^k$ or $x_i^j > x_i^k$&$y^j < y^k$ [53]. Therefore, the KRCC (τ_i) between the ith decision variable x_i and the objective y can be calculated as follows:

$$\tau_i = \frac{n_c - n_d}{n(n-1)/2} \tag{1}$$

where n_c is the number of concordant pairs and n_d is the number of discordant pairs. The value of τ_i will locate in $[-1, 1]$. Here, the value of τ_i close to -1 or 1 indicates the stronger correlation of x_i to the objective y. Especially, x_i is independent to the objective y when $\tau_i = 0$, so x_i can be ignored when training the RBFN model.

After calculating the correlation value of KRCC between each decision variable and the objective, the contribution ratio of the ith decision variables x_i to the objective y can be calculated using the KRCC (τ_i), as follows:

$$r_i = \frac{|\tau_i|}{\sum_{j=1}^{d} |\tau_j|} \tag{2}$$

where d is the dimensions of x_i. Therefore, t training sets ($\boldsymbol{S}_1^{train}, \boldsymbol{S}_2^{train}, \ldots, \boldsymbol{S}_t^{train}$) and t test sets ($\boldsymbol{S}_1^{test}, \boldsymbol{S}_2^{test}, \ldots, \boldsymbol{S}_t^{test}$) can be generated through roulette wheel selection based on the contribution ratio by (2) and random sampling. t training datasets are used to train t local RBF models ($m_1, m_2, \ldots, m_t$), while t test datasets are used to choose some of them to construct a final selective ensemble.

As experimentally studied in [14] and [29], the ensemble learning methods or the ensembles of surrogate models have shown effectiveness in enhancing the performance of SAEAs. Therefore, in this article, the ensemble method is also used to combine diverse surrogates in order to improve the prediction accuracy. In this article, the Kullback–Leibler divergence (KLD) [54] is used to select the local models, which is able to assess the similarity between two probability distributions, so as to reduce the uncertainty of prediction. In mathematical statistics, KLD can reflect how much one probability distribution diverges from another. Consider two normal distributions A and B, as follows:

$$A \sim \mathcal{N}\left(\mu_1, \sigma_1^2\right), B \sim \mathcal{N}\left(\mu_2, \sigma_2^2\right) \tag{3}$$

and the KLD from B to A is often denoted as KLD($A|B$), which can be calculated by

$$\mathrm{KLD}(A|B) = \log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2} \tag{4}$$

where μ_1 and μ_2 are the mean values, while σ_1 and σ_2 are the standard deviations for two normal distributions A and B from $\mathcal{N}(\bullet, \bullet)$, respectively.

For each local model, the predicted fitness values Y_p can be obtained for S_i^{test}. Next, the KLD value from Y_p to the true fitness values Y_r [i.e., KLD($Y_r|Y_p$)] can be calculated according to (4). It is worth noting that KLD is asymmetric and KLD($Y_r|Y_p$) is not equal to KLD($Y_p|Y_r$) [55]. After obtaining the KLD value for all t local RBFN models, a subset of the local models is selected to construct the ensemble based on their KLD values.

In this approach, top-k best weighted selection strategy (Top-K WSS) is used to choose only the first k local models with the smallest KLD values to form selective ensemble. To be specific, Top-K WSS constructs the selective ensemble as follows:

$$se = \{m_i | i \in \mathbf{I}\} \tag{5}$$

where se denotes the selective ensemble produced by Top-K WSS and $\mathbf{I}$ denotes the index set of the k best models.

Meanwhile, the predicted value of the selective ensemble is given by

$$y_{se} = \sum_{i \in \mathbf{I}} w_i y_i \tag{6}$$

$$w_i = \frac{\mathrm{KLD}_i}{\sum_{j=1}^{t} \mathrm{KLD}_j} \tag{7}$$

where y_i is the output value of the ith local models m_i, w_i is the weight assigned to m_i, and KLD_i is the KLD value of m_i.

To clarify the details of this process, Algorithm 2 is the pseudocode of constructing the selective ensemble surrogate. The inputs of Algorithm 2 include a set of sample data stored in database $\boldsymbol{TD}$, while the output is the selective ensemble surrogate se and the local models set $\boldsymbol{LS}$. At the beginning, the number of total local models t and the number of selected models k for ensemble are initialized, and local models set $\boldsymbol{LS}$ is set as an empty set in line 1. In line 2, the contribution ratio r_i of each decision variable to the objective is calculated according to (1) and (2). Then, in lines 4–12, t local models are built in turn by using diverse training samples. To be specific, the variable indexes ind_i are obtained from the original variable vector by roulette wheel selection in order to build the ith local surrogate in line 4. To generate a training data subset for the ith local surrogate, a probability-dependent sampling method is employed in lines 5–11. Each sample s_i in $\boldsymbol{TD}$ has a probability of 0.75 to be selected as training sample for the ith local model, which is added into the training set $\boldsymbol{S}_i^{train}$. After selecting t training datasets by a probability-dependent sampling method, t test datasets ($\boldsymbol{S}_1^{test}, \boldsymbol{S}_2^{test}, \ldots, \boldsymbol{S}_t^{test}$) are also generated by saving the remaining solutions that are not used as training samples. In lines 12 and 13, a local model m_i is constructed by using $\boldsymbol{S}_i^{train}$ and m_i with ind_i are added into $\boldsymbol{LS}$. The KLD value for m_i is calculated by (3) and (4) in lines 14 and 15. Finally, in lines 17–19, Top-K WSS is employed to construct a final selective ensemble se. It is worth noting that the model selection procedure will be triggered each time when the $\boldsymbol{TD}$ is updated during the evolution.

In this approach, the number of variables selected by roulette wheel selection is random and thus the variable dimension of each training subset is different. Since the probability-dependent sampling is used in our approach, the size of each training set is not fixed, which can obtain local surrogates with more diversity. In addition, in several SAEAs [56], [57], the size of the training set is usually set to be more than that of the test set. Thus, the probability to select samples as training samples is set to 0.75 as suggested in [56] and other values can also be adopted, for example, 0.8 as suggested in [57].

C. Surrogate-Assisted Coevolutionary Optimizer

After constructing the selective ensemble, SACO is further used to search more promising solutions, in which the selective ensemble surrogate se built in the above section is used to approximate the fitness values of solutions instead of true function evaluation. To clarify the running of SACO, its pseudocode is given in Algorithm 3. n best solutions from $\boldsymbol{TD}$ are selected as an initial population $\boldsymbol{P}$, while an auxiliary population $\boldsymbol{P}_a$ is initialized by the Latin hypercube sampling [50].

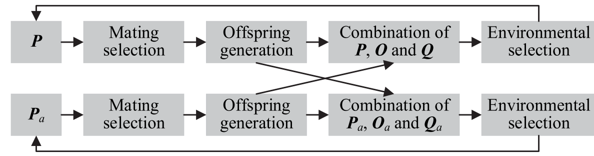


Fig. 2. Procedures of surrogate-assisted coevolutionary optimization.

It is worth noting that the dimension of $\boldsymbol{P}$ is the same as the original LSEOP, while that of $\boldsymbol{P}_a$ is consistent with the training samples of the cth local model in se (c is a random integer number in [1, k] and k is the number of selected models). In each generation, two parent sets with size $n/2$ are selected from $\boldsymbol{P}$ and $\boldsymbol{P}_a$ by the mating selection strategy, respectively. Then, the two parent populations are applied to generate their corresponding offspring sets ($\boldsymbol{O}$ and $\boldsymbol{O}_a$) with the same size by sequentially running simulated binary crossover [58] and polynomial mutation [59]. Afterward, for $\boldsymbol{P}$, another offspring population $\boldsymbol{Q}$ is produced by sequentially replacing the selected ind_i decision variables of $\boldsymbol{O}$ by the decision variables of $\boldsymbol{O}_a$ and then maintaining the remaining decision variables in $\boldsymbol{O}$, while the corresponding ind_i decision variables of $\boldsymbol{O}$ are used to form the second offspring set $\boldsymbol{Q}_a$ of $\boldsymbol{P}_a$. After that, both $\boldsymbol{P}$ and $\boldsymbol{P}_a$ are combined with their corresponding two offspring populations as the union population ($\boldsymbol{U}$ and $\boldsymbol{U}_a$). In the next step, the solutions in $\boldsymbol{U}$ are evaluated by se, and the solutions in $\boldsymbol{U}_a$ are always evaluated by the cth local model m_c in $\boldsymbol{LS}$. Once the fitness values of all solutions are predicted by our surrogates, n best solutions will be selected from $\boldsymbol{U}$ and $\boldsymbol{U}_a$, respectively, which are survived into the next generation. Finally, $\boldsymbol{P}$ is returned as the final output of SACO.

According to Algorithm 3, it is quite clear that two populations $\boldsymbol{P}$ and $\boldsymbol{P}_a$ are evolved simultaneously to solve the original problem f and the simplified auxiliary problem f_a, respectively. Since f_a only includes part of decision variables in f, the optimization of f_a is easier than that of f. Therefore, as illustrated in Fig. 2, this coevolutionary process can quickly find more promising solutions with the sharing of offspring generated by two coevolutionary populations. In other words, the convergence speed of $\boldsymbol{P}$ will be accelerated by $\boldsymbol{P}_a$, in which solutions in $\boldsymbol{P}_a$ usually have better convergence than those in $\boldsymbol{P}$.

D. Infill Solutions Selection

Since one true function evaluation has to consume a large number of computational time for LSEOPs, solutions to be truly evaluated should be carefully selected. These selected solutions are named infill solutions, which are saved into $\boldsymbol{TD}$ and used to retrain surrogate models. If too many unimportant and redundant solutions are selected as infill solutions, a very limited number of true evaluations will be wasted, resulting in making no sense for reducing the computational cost.

In this article, a simple and effective infill selection criterion is proposed and its pseudocode is given in Algorithm 4. The selection criterion is based on the predicted fitness values and the distance between candidate solutions and samples in $\boldsymbol{TD}$. The process of identifying infill solutions is conducted in the final offspring population $\boldsymbol{P}$ obtained in Section III-C. First, some unimportant solutions should be removed and the remaining ones are selected into a candidate pool $\boldsymbol{C}$. To be specific, for each solution s in $\boldsymbol{P}$, if its approximate objective value predicted by our ensemble surrogate is smaller than the true objective value of the best sample b in $\boldsymbol{TD}$, and the minimum Euclidean distance in the decision space from s to all samples is larger than the predefined threshold l, this solution will be selected into the candidate pool $\boldsymbol{C}$. After such a process is performed for each solution of $\boldsymbol{P}$, all the solutions in $\boldsymbol{C}$ are sorted in an ascending order according to their approximate fitness values, and then a solution with the best approximate fitness value is selected to retrain the surrogates. In a special case that no solution exists in the candidate pool $\boldsymbol{C}$, one solution will be randomly selected from $\boldsymbol{P}$. Note that the probability of such a special case is very small, which will not significantly impact our performance.

E. Computational Complexity

The computational complexity of ESCO is determined by the computational cost for function evaluations, training the RBFN models, running the coevolutionary optimizer, as well as selecting the infill solutions. To be specific, Algorithm 1 requires a time complexity of $O(tn_s^3)$ to obtain t RBFN models and the final selective ensemble in line 2. In line 3, it needs a time complexity of $O(g_{max}n)$ to run the SACO. In line 4, it has a time complexity of $O(n)$. Therefore, the overall worst time complexity of ESCO is $O(tn_s^3)$ in one generation, as $O(tn_s^3)$ is more complex than $O(g_{max}n)$ and $O(n)$.

IV. Numerical Experimental Study

In this section, the effectiveness of the proposed ESCO is studied by conducting extensive experiments. First, Section IV-A summarizes a set of commonly used benchmark EOPs and the parameter settings. Next, in Sections IV-B and IV-C, the experimental results between ESCO and nine competitive SAEAs are analyzed and discussed. Finally, more studies of ESCO are given in Section IV-D to analyze the effectiveness of its each component.

A. Benchmark Problems and Parameter Settings

In order to conduct experiments with some state-of-the-art SAEAs, eight benchmark EOPs (F1–F8) are widely used in the field of SAEA, that is, one unimodal EOP and seven multimodal EOPs [10], [60], [61], [62], [63]. Due to page limitations, Table A.II of the supplementary file summarizes the dimensions, global optimum, and characteristics of F1–F8. In our experimental studies, for all the compared algorithms except DDEA-SE, the number of initial samples is set to 200 for problems with $d = 100$ or 300 and to 400 for problems with $d = 500$ or 1000 (d is the dimension of decision space), as suggested in CA-LLSO [35] for a fair comparison. The total number of true function evaluations is all set to 1000. For DDEA-SE, all 1000 function evaluations are used to generate offline samples. In our algorithm, the number of local models t is set to 10 and the number of selected local models k in the Top-K WSS strategy is set to 3. The population size n and the maximum of iteration g_{max} are set to 50 and 20, respectively. The predefined threshold l is set to sqrt($10^{-6} \times d$). Due to page limitations, the details of other parameter settings are introduced in Table A.III of the supplementary file. The source code of ESCO can be downloaded at https://github.com/Xunfeng-Wu/ESCO.

B. Comparisons With the State-of-the-Art SAEAs

In order to verify the performance of our algorithm, ESCO is compared with five state-of-the-art SAEAs (DDEA-SE [14], MGP-SLPSO [34], SA-COSO [13], CA-LLSO [35], and ESAO [12]) on all the benchmark LSEOPs (F1–F8) used in this article. The average optimal function values obtained by all the compared SAEAs are collected in Table I after 20 independent runs. In addition, the Wilcoxon rank results [64] with the significance level of 0.05 are also presented in Table I. Here, symbol "$\sim$" indicates that there is no statistically significant difference between the results of ESCO and the compared algorithms, that is, their performance is statistically similar. Symbol "$+$" indicates that ESCO is significantly worse than the compared algorithms, while "$-$" denotes that ESCO is significantly better than the compared algorithms. As shown in Table I, ESCO obtains the best mean results on 22 out of 32 cases, while CA-LLSO, ESAO, and MGP-SLPSO achieve the best mean results only on 4, 4, and 2 cases, respectively. DDEA-SE and SA-COSO cannot obtain the best result on any test case in the experiment.

When compared to DDEA-SE, ESCO performs better on 27 out of 32 cases and similarly on three cases. DDEA-SE is an offline SAEA with superior performance, but its performance is significantly worse than that of ESCO, as offline SAEA only have a limited number of offline samples and no new samples can be made available to retrain the surrogate model. Thus, online SAEAs are more advantageous when new samples can be evaluated, while DDEA-SE is more suitable for the applications in which new samples are unavailable. Although MGP-SLPSO and ESCO are both online SAEAs, ESCO is also better on 24 cases and only worse on four cases when compared to MGP-SLPSO, and the advantages of ESCO are more distinct when compared to SA-COSO, as ESCO obtains the better results on 28 cases. In the comparison between ESAO and ESCO, ESCO achieves 24 better results and five similar results on a total of 32 cases, while ESAO achieves only three better results. Finally, ESCO also performs better than CA-LLSO on 26 cases and is only defeated by CA-LLSO on three cases. Thus, it is reasonable to conclude that the proposed algorithm ESCO shows distinct advantages when compared to these five state-of-the-art SAEAs.

To visually show the competitive performance of ESCO, Fig. 3 plots the convergence curves of MGP-SLPSO, SA-COSO, ESAO, CA-LLSO, and ESCO on the 100-D (100-dimension), 300-D, 500-D, 1000-D F1, and F4, in which the x-axis denotes the true fitness evaluations, while the y-axis denotes the mean function values obtained from 20 independent runs. Since DDEA-SE is an offline SAEA, it is not included for comparison in Fig. 3. On a unimodal problem F1,

TABLE I
AVERAGE OPTIMAL FUNCTION VALUES OBTAINED BY ESCO AND FIVE COMPARED SAEAs

Problem	D	DDEA-SE	MGP-SLPSO	SA-COSO	CA-LLSO	ESAO	ESCO
F1	100	4.90E+02(1.27E+02) -	**7.59E-02(1.12E-01)** +	8.86E+02(5.52E+01) -	1.00E+03(1.21E+02) -	6.07E+02(3.52E+01) -	2.37E+02(6.50E+01)
	300	3.05E+04(8.47E+03) -	4.78E+04(7.34E+03) -	7.34E+04(2.25E+04) -	2.81E+04(1.39E+03) -	3.20E+04(2.37E+03) -	**2.30E+03(1.50E+02)**
	500	1.34E+05(2.47E+04) -	3.66E+05(1.24E+04) -	3.11E+05(4.23E+04) -	1.52E+05(7.14E+03) -	1.29E+05(4.52E+03) -	**2.30E+04(6.65E+03)**
	1000	9.93E+05(1.94E+05) -	1.47E+06(2.05E+05) -	1.83E+06(1.43E+05) -	7.11E+05(4.34E+04) -	5.83E+05(2.74E+03) -	**1.77E+05(1.00E+02)**
F2	100	3.50E+02(1.46E+01) -	4.96E+02(7.00E+01) -	2.52E+03(5.26E+02) -	4.58E+02(4.76E+01) -	3.13E+02(2.16E+01) -	**2.24E+02(9.74E+00)**
	300	4.92E+03(1.16E+03) -	2.60E+04(1.50E+03) -	3.60E+04(3.22E+03) -	3.92E+03(4.45E+02) -	5.19E+03(4.67E+02) -	**1.05E+03(3.32E+02)**
	500	1.28E+04(2.79E+03) -	5.60E+04(2.32E+03) -	7.40E+04(7.75E+03) -	1.46E+04(6.56E+02) -	1.47E+04(7.23E+02) -	**3.33E+03(4.27E+02)**
	1000	7.03E+04(4.81E+03) -	9.49E+04(4.53E+03) -	1.96E+05(5.28E+02) -	4.08E+04(2.04E+03) -	3.01E+04(3.67E+03) -	**1.02E+04(3.87E+03)**
F3	100	9.38E+02(5.87E+01) -	6.63E+02(1.34E+02) -	8.68E+02(7.80E+01) -	8.50E+02(1.36E+01) -	9.57E+02(1.96E+01) -	**4.06E+02(4.88E+01)**
	300	3.23E+03(9.76E+01) -	3.63E+03(5.90E+01) -	3.40E+03(3.92E+01) -	2.90E+03(3.91E+01) -	3.04E+03(3.96E+01) -	**2.10E+03(8.00E+01)**
	500	5.48E+03(1.63E+02) -	6.18E+03(6.74E+01) -	6.27E+03(1.08E+02) -	5.20E+03(6.82E+01) -	5.23E+03(7.50E+01) -	**4.03E+03(1.76E+02)**
	1000	1.13E+04(3.29E+02) -	1.21E+04(1.45E+02) -	1.32E+04(2.94E+02) -	1.08E+04(1.35E+02) -	1.07E+04(1.67E+02) -	**8.32E+03(5.95E+02)**
F4	100	9.05E+00(3.44E-01) -	1.29E+01(6.36E-01) -	1.54E+01(4.37E-01) -	1.06E+01(3.32E-01) -	6.85E+00(1.82E-01) -	**6.11E+00(3.30E-01)**
	300	1.49E+01(3.29E-01) -	1.94E+01(1.35E-01) -	1.85E+01(2.66E-01) -	1.49E+01(2.79E-01) -	1.39E+01(1.28E-01) -	**8.52E+00(8.58E-01)**
	500	1.67E+01(1.41E-01) -	1.94E+01(4.13E-02) -	1.93E+01(2.05E-01) -	1.70E+01(1.34E-01) -	1.61E+01(1.70E-01) -	**6.77E+00(9.13E-01)**
	1000	1.89E+01(7.00E-03) -	1.92E+01(7.12E-03) -	2.00E+01(8.30E-03) -	1.76E+01(1.56E-01) -	1.63E+01(7.70E-03) -	**6.39E+00(5.80E-03)**
F5	100	1.68E+01(2.99E+00) -	**1.10E-01(7.46E-02)** +	6.26E+01(1.91E+01) -	7.88E+01(1.14E+01) -	2.71E+01(2.59E+00) -	1.16E+01(2.14E+00)
	300	5.93E+02(5.69E+01) -	1.41E+03(1.66E+02) -	1.39E+03(1.95E+02) -	7.30E+02(6.65E+01) -	7.13E+02(3.90E+01) -	**4.31E+01(3.46E+00)**
	500	2.06E+03(1.90E+02) -	5.20E+03(2.23E+02) -	4.41E+03(3.75E+02) -	2.26E+03(1.49E+02) -	1.88E+03(7.22E+01) -	**3.36E+02(5.39E+01)**
	1000	6.66E+03(1.88E+02) -	9.04E+03(3.65E+02) -	1.31E+04(3.63E+02) -	5.14E+03(3.31E+02) -	3.95E+03(5.40E-01) -	**1.42E+03(9.68E+01)**
F6	100	1.27E+03(3.64E+01) ~	8.05E+02(3.14E+01) +	1.27E+03(1.19E+02) ~	1.28E+03(3.02E+01) ~	**7.25E+02(3.49E+01)** +	1.37E+03(8.32E+01)
	300	8.63E+03(4.57E+02) -	8.49E+03(1.28E+02) -	8.97E+03(2.63E+02) -	**6.06E+03(1.77E+02)** +	8.09E+03(1.40E+02) -	7.38E+03(1.44E+02)
	500	1.54E+04(6.70E+02) -	1.53E+04(1.06E+02) -	1.61E+04(1.90E+02) -	**1.26E+04(2.07E+02)** +	1.40E+04(1.85E+02) ~	1.38E+04(9.86E+01)
	1000	2.94E+04(1.24E+02) ~	3.14E+04(3.18E+02) -	3.02E+04(4.26E+01) -	2.96E+04(2.24E+03) ~	**2.89E+04(1.67E+02)** ~	2.90E+04(3.67E+02)
F7	100	8.41E+02(2.90E+01) +	6.63E+02(2.01E+01) +	7.78E+02(6.60E+01) +	1.76E+03(4.34E+01) -	**5.09E+02(3.37E+01)** +	9.83E+02(3.85E+01)
	300	1.83E+03(5.52E+01) -	1.73E+03(3.17E+01) ~	1.76E+03(1.64E+01) -	2.30E+03(3.05E+01) -	**1.67E+03(1.78E+01)** +	1.69E+03(1.34E+01)
	500	1.88E+03(6.32E+01) -	1.77E+03(1.34E+01) ~	1.84E+03(1.31E+01) -	2.58E+03(1.23E+01) -	1.75E+03(1.59E+01) ~	**1.74E+03(4.69E+01)**
	1000	2.83E+03(1.71E+00) ~	2.88E+03(5.64E+00) ~	2.80E+03(2.54E+01) ~	**2.73E+03(1.05E+00)** +	2.82E+03(1.75E+00) ~	2.84E+03(6.71E+00)
F8	100	1.40E+03(4.60E+01) -	1.40E+03(3.52E+01) -	1.41E+03(1.94E+01) -	1.37E+03(1.92E+01) -	1.35E+03(1.32E+01) -	**1.28E+03(4.13E+01)**
	300	1.46E+03(9.91E+00) -	1.48E+03(1.26E+01) -	1.52E+03(1.64E+01) -	1.39E+03(4.00E+00) -	1.41E+03(6.80E+00) -	**1.20E+03(5.59E+01)**
	500	1.53E+03(4.26E+01) -	1.50E+03(3.71E+00) -	1.56E+03(9.26E+00) -	1.49E+03(2.94E+01) -	1.42E+03(1.29E+01) -	**1.30E+03(1.52E+02)**
	1000	1.67E+03(1.70E+00) ~	1.56E+03(2.53E+00) ~	1.55E+03(1.36E+01) ~	**1.53E+03(2.72E+00)** ~	1.54E+03(5.06E+00) ~	1.55E+03(6.76E+00)
+/-/~		1/27/4	4/24/4	1/28/3	3/26/3	3/24/5	

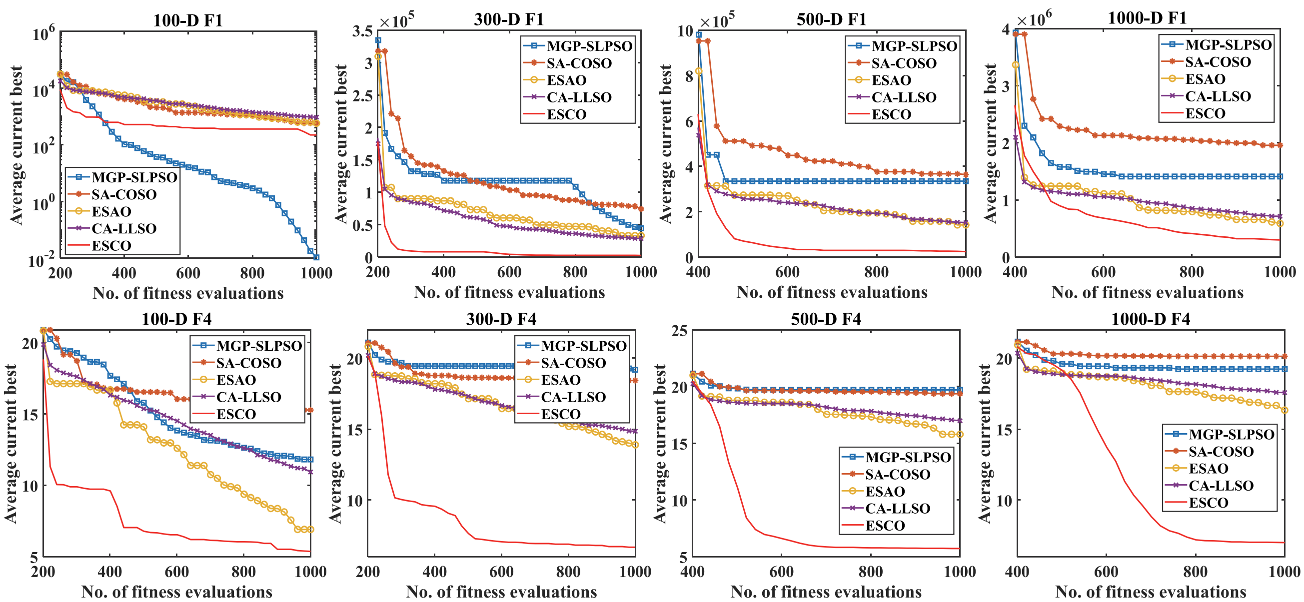


Fig. 3. Convergence curves of the proposed ESCO and four competitive SAEAs on F1 and F4.

the convergence speed of ESCO is faster than that of its four competitors on all the cases of F1. ESAO, MGP-SLPSO, and ESCO are able to find much better solutions than the other two competitors. MGP-SLPSO performs best on 100-D F1, while ESCO achieves the best results on 300-D, 500-D, and 1000-D F1. On a multimodal problem F4 with a very narrow basin around the global optimum, ESCO outperforms MGP-SLPSO, SA-COSO, ESAO, and CA-LLSO on all the cases of F4. These results indicate that although the narrow basin around the optimum of F4 makes it very difficult for the surrogate to effectively learn the fitness landscape, ESCO using our proposed selective ensemble surrogate still shows significantly better performance on F4. Due to page limitations, the convergence curves of ESCO and its four competitors on the remaining test problems (i.e., F2, F3, and F5–F8) are provided in Fig. A.1 of the supplementary file. These figures also show that ESCO has a faster convergence speed to approximate the global optima of most test cases, which confirms the efficiency of our proposed ESCO for dealing with LSEOPs.

TABLE II
SUMMARY OF COMPARISON OF ESCO WITH FOUR COMPETITIVE SAEAS

ESCO vs.		SRK-DDEA	TT-DDEA	SACC	SAEA-RFS
F1-F15 from CEC2013	Better	14	14	11	9
	Worse	0	0	1	2
	Similar	1	1	3	4
Best	9	1	0	2	3

To further evaluate the actual runtime of the above compared SAEAs, their average running times (in second: s) are plotted in Fig. A.2 of the supplementary file due to page limitations. According to Fig. A.2 of the supplementary file, DDEA-SE has a much lower computational complexity, mainly because DDEA-SE only trains a surrogate model once. Moreover, the running times of ESCO, SA-COSO, and ESAO are considerably more than that of others, which can be attributed to the use of multiple surrogate models in these algorithms. The computational cost of SAEA is mainly determined by the training time of surrogate models. More surrogate models to be trained in SAEAs will induce a higher computational cost. However, although training multiple surrogate models incurs additional computational cost (about a few minutes), it is considered negligible when compared to the computational cost (requiring hours or days) of the actual fitness evaluations in practical application.

C. Experiment on the CEC2013 Problems

In the above section, ESCO has been compared with five competitive SAEAs that are designed for solving EOPs with a relatively small number of decision variables. In this section, ESCO is further compared with four competitive SAEAs (SRK-DDEA [36], TT-DDEA [37], SACC [38], and SAEA-RFS [39]) that are designed for solving LSEOPs on the CEC2013 problems (i.e., 1000 dimensions in F1–F12 and F15, and 905 dimensions in F13 and F14) [65]. The characteristics of the CEC2013 problems can be found in [65], which can be classified into four categories: 1) fully separable problems (F1–F3); 2) partially separable problems (F4–F11); 3) overlapping problems (F12–F14); and 4) nonseparable problems (F15).

Due to page limitations, their average optimal function values are collected in Table A.IV of the supplementary file and Table II summarizes their pairwise comparison results. From Table II, ESCO still shows the best performance on most cases of CEC2013 problems when compared to four competitive SAEAs. Here, ESCO performs best on 9 out of 15 problems. When compared to SRK-DDEA, TT-DDEA, SACC, and SAEA-RFS, ESCO performs worse/better/similarly on 0/14/1, 0/14/1, 1/11/3, and 2/9/4 out of 15 cases, respectively. Therefore, it is reasonable to conclude that ESCO also shows significantly better performance than these four competitors on the CEC2013 problems.

To be specific, for three completely separable problems (F1–F3), ESCO performs best on all cases when compared to SRK-DDEA, TT-DDEA, SACC, and SAEA-RFS, and only SACC and SAEA-RFS are able to obtain similar performance on F3. In terms of eight partially separable problems (F4–F11), ESCO performs significantly better than both SRK-DDEA and TT-DDEA on all cases. Also, ESCO performs significantly better than SACC on all cases except F7 and F10, on which ESCO and SACC have similar performance. Moreover, when compared to SAEA-RFS, ESCO is able to obtain significantly better results on the three problems (i.e., F5, F9, and F10), while SAEA-RFS can surpass ESCO on two problems (F4 and F8). Especially on F8, SAEA-RFS performs much better than ESCO. As for three overlapping problems (F12–F14) and the last completely nonseparable problem (F15), the overall performance of ESCO is also the best when compared with the other four compared SAEAs, and only SACC can show better performance on F15. SAEA-RFS seems to be the most competitive one among the four compared algorithms, as SAEA-RFS achieves three best results on a total of 15 cases. However, when compared with SAEA-RFS, ESCO can still obtain nine better results and four similar results on a total of 15 cases.

Due to page limitations, in Fig. A.3 of the supplementary file, the convergence trends of ESCO and its four competitors on the CEC2013 problems are plotted. According to this figure, it can be seen that ESCO has a better convergence trend than its competitors on most problems.

D. Further Study of ESCO

1) Effectiveness of the Top-K WSS: In Section III-B, a novel model selection strategy (Top-K WSS), which selects the best k local models according to their KLD values, is proposed to construct a selective ensemble surrogate in ESCO. To study the effectiveness of our proposed model selection strategy, the Top-K WSS is compared with two other model selection strategies in ESCO, that is, the best selection strategy (BSS) [57] and the weighted selection strategy (WSS) [29].

The BSS selects the best local model with the smallest KLD value as the current surrogate, which can be modeled by

$$se = \left\{m_{\arg\min_i(\mathrm{KLD}_i)}\right\} \tag{8}$$

$$y_{se} = y_{\arg\min_i(\mathrm{KLD}_i)}. \tag{9}$$

The WSS uses a weighted sum of all the trained local models to construct our selective ensemble, as follows:

$$se = \{m_1, m_2, \ldots, m_t\} \tag{10}$$

$$y_{se} = \sum_{i=1}^{t} w_i y_i. \tag{11}$$

Similar to (5) and (6), se is the selective ensemble obtained by model selection strategies and y_{se} is the predicted values of se in (8)–(11). Therefore, ESCO is compared with its two variants in the experimental study. To facilitate the understanding, ESCO+BSS, ESCO+WSS, and ESCO indicate the proposed ESCO with the corresponding model selection strategy (BSS, WSS, and Top-K WSS), respectively.

All the three variants of ESCO are compared on the test LSEOPs with 1000 true function evaluations as introduced in Section IV-A, and their average optimal function values from 20 independent runs are collected in Table A.V of the supplementary file due to page limitations. As shown in Table III that

TABLE III
SUMMARY OF COMPARISON OF ESCO WITH FOUR VARIANTS

ESCO vs.		ESCO+BSS	ESCO +WSS	ESEA	ExI
F1-F8 (d = 100, 300, 500, 1000)	Better	17	18	17	15
	Worse	5	7	4	6
	Similar	10	7	11	11
Best	16	5	5	4	2

summarizes the pairwise comparison, ESCO is the best one, as it obtains the best performance in most cases. To be specific, ESCO achieves the best results on 16 out of 32 cases, while both ESCO+BSS and ESCO+WSS can obtain the best results on five cases. When compared to ESCO+BSS, ESCO performs better on 17 out of 32 cases and similarly on ten cases. When compared to ESCO+WSS, ESCO performs better on 18 out of 32 cases and similarly on seven cases. On the other hand, ESCO+BSS shows superior performance over ESCO on only five cases, that is, 100-D F7, 300-D F4, 500-D F3/F4, and 1000-D F7, while ESCO+WSS is better than ESCO on only seven cases, that is, 100-D F1, 300-D F4, 500-D F1/F4/F5, and 1000-D F4/F7.

Due to page limitations, Fig. A.4 of the supplementary file plots the convergence curves of ESCO with its two variants on three representative test LSEOPs, that is, F2, F5, and F6 with d = 100, 300, 500, and 1000. F2 is a multimodal problem with a narrow valley leading to the global optimum. For F2, ESCO achieves a faster convergence speed than the other two algorithms on all the dimensions of this problem. ESCO, ESCO+BSS, and ESCO+WSS perform similarly on 500-D F2 under 400–800 function evaluations. However, ESCO outperforms ESCO+ BSS and ESCO+WSS on 500-D F2 when the function is truly evaluated with 800–1000 times. Thus, ESCO shows a significant advantage in convergence speed of 1000-D F2. For the multimodal problem F5, ESCO performs better than both its two variants when d = 100, 300, and 1000, which validates the effectiveness of the proposed Top-K WSS. ESCO+WSS shows a better convergence speed than ESCO on 500-D F5. ESCO+BSS obtains the worst result on 300-D F5 due to the fact that the BSS only utilizes one best model, which indicates the importance of using multiple models. F6 is a very complicated multimodal problem and the number of its local optima is huge, which makes it very hard to optimize. From Fig. A.4 of the supplementary file, ESCO exhibits a clear advantage over the compared algorithms on the 300-D, 500-D, and 1000-D F6, while it is only slightly beaten by both ESCO+BSS and ESCO+WSS on 100-D F6. When the allowed computational budget is reduced to 400 function evaluations, the performance of ESCO remains to be clearly superior over the two competitors, confirming its competitive performance in tackling the complex problems. It also can be found that the performance of ESCO becomes more advantageous as the number of decision variables increases. Moreover, ESCO+WSS fails to obtain sufficient convergence on 300-D F6, in particular when the number of function evaluations is larger than 220. This is because a portion of good local models are able to provide promising prediction accuracy for solving complex problems, while adding more local models with poor quality may reduce the prediction performance of ensemble surrogate. The above experimental results confirm that our proposed selective ensemble surrogate using the Top-K WSS selection strategy can significantly improve the optimization performance of ESCO.

2) Contribution Analysis of the Coevolutionary Optimizer: In this section, the contribution of the coevolutionary optimizer proposed in Section III-C is studied. Here, ESCO is further compared to its variant without using the coevolutionary optimizer, which is simply denoted as ESEA in this article. In ESEA, only one population $\boldsymbol{P}$ is evolved to solve the original problem f without using the auxiliary population in Fig. 2. Due to page limitations, Table A.V of the supplementary file provides the average optimal function values found by ESCO and ESEA in 20 runs for solving test LSEOPs as summarized in Section IV-A. According to Table III that summarizes the pairwise comparison, it can be found that ESCO significantly outperforms ESEA, since ESCO performs better than or similar to ESEA on 28 out of 32 cases, while ESCO is defeated by ESEA on only four cases, that is, 100-D F1, 300-D F4, 100-D, and 1000-D F7. Thus, the coevolutionary optimizer used in ESCO contributes significantly to its improved performance, as the solving of a simple auxiliary problem can help to solve the original EOP.

According to the summary shown in Table A.V of the supplementary file, the impact of the coevolutionary optimizer in ESCO has a strong relationship with problem dimensions. When compared to ESEA, ESCO performs better/worse/similarly on 1/2/5 cases for 100-D problems, on 5/1/2 cases for 300-D problems, on 5/0/3 cases for 500-D problems, and on 6/1/1 cases for 1000-D problems. These results indicate that the coevolutionary optimizer is more important with the increase of problem dimensions, as the target LSEOP becomes more complex and the solving of the simplified auxiliary problem can help to solve the original LSEOP in such cases.

To further examine the effectiveness of our proposed coevolutionary optimizer, the convergence curves of ESCO and ESEA on a representative problem F5 with d = 100, 300, 500, and 1000 are plotted in Fig. A.5 of the supplementary file due to page limitations. Taking a closer look at this figure, an obvious observation is that the convergence speed of ESEA is slightly faster than that of ESCO on 100-D F5, while ESCO converges significantly faster than ESEA on 300-D, 500-D, and 1000-D F5. Thus, with the increase of problem dimensions, the advantages of ESCO in terms of the convergence speed and optimization performance become more and more prominent. These observations are consistent with the above conclusion that the use of the coevolutionary optimizer introduced in Section III-C guarantees a fast convergence speed of ESCO for handling LSEOPs.

3) Effectiveness of the Infill Solutions Selection: To validate the effectiveness of our proposed infill solutions selection presented in Section III-D, a common and popular infill solutions selection criterion, that is, expected improvement (ExI) [11], is embedded into ESCO by replacing the proposed infill solutions selection, giving a variant of ESCO identified as ExI in this article for the sake of convenience.

Due to page limitations, the average optimal function values obtained by ESCO and its variant using ExI are given

TABLE IV
SUMMARY OF COMPARISON OF ESCO WITH DIFFERENT k VALUES

ESCO (k = 3) vs.		k = 1	k = 5	k = 7	k = 9
F1-F8 (d = 100, 300, 500, 1000)	Better	17	17	17	16
	Worse	5	5	6	6
	Similar	10	10	9	10
Best	20	5	1	2	4

in Table A.V of the supplementary file for solving the test LSEOPs summarized in Section IV-A. Also, Table III summarizes their pairwise comparison results. Overall, when compared to ExI, our ESCO performs better on 15 cases and worse on six cases. Note that ExI performs better mainly on 100-D F7, 300-D F4, 500-D F3/F4, and 1000-D F4/F7, while ESCO performs better or similarly on all the dimensions of F1, F2, F5, F6, and F8.

4) Influence of the Selected Model Number in Top-K WSS: This part studies the impact of the number of local models (i.e., the parameter k) selected in the Top-K WSS for building the final selective ensemble surrogate. Thus, the performance of ESCO is studied with different numbers of selected local models (i.e., $k = 1$, 3, 5, 7, and 9), for solving the 100-D, 300-D, 500-D, and 1000-D F1-F8 test problems. Due to page limitations, their average optimal function values are collected in Table A.VI of the supplementary file and Table IV summarizes their pairwise comparison results. ESCO with $k = 3$ (the suggested setting) obtains the best results (marked in bold) on 20 out of 32 cases, while other ESCO variants with different k values achieve the best results on the rest 12 cases. As indicated by Wilcoxon's rank-sum tests [64], ESCO with $k = 3$ can perform significantly better than that with $k = 1$, $k = 3$, $k = 5$, and $k = 9$ on 17, 17, 17, and 16 cases, respectively. Furthermore, ESCO with $k = 3$ is outperformed by ESCO with $k = 1$, $k = 3$, $k = 5$, and $k = 9$, only on 5, 5, 6, and 6 cases, respectively. The above comparison results indicate that the number of local models in Top-K WSS to construct selective ensemble surrogate is crucial to the performance of ESCO. As verified by this experiment, the setting of selected local models ($k = 3$) to build the ensemble is effective and reasonable.

5) Effectiveness of the Selective Ensemble: Here, the effectiveness of the selective ensemble proposed in Section III-B is further studied. In our SACO, a randomly selected local model is used instead of selective ensemble to approximate the function value, which yields another variant of ESCO, called ESCO/WoSE. Due to page limitations, the results of ESCO and ESCO/WoSE on the CEC2013 test suite are presented in Table A.VII of the supplementary file.

According to Table A.VII of the supplementary file, ESCO performs significantly better than ESCO/WoSE, as ESCO can obtain the better or similar results on all the CEC2013 problems. Therefore, it is reasonable to conclude that the selective ensemble proposed in Section III-B has significant contributions for the performance improvement of ESCO.

6) Influence of the Probability for Constructing Training Set: This part considers the influence of different probabilities to select each sample as training samples in Section III-B. As the employed probability of the original ESCO is $U(0, 1) < 0.75$, its variant with $U(0, 1) < 0.5$ is employed for comparison. Due to page limitations, their comparison results are provided in Table A.VII of the supplementary file. From Table A.VII of the supplementary file, it can be seen that ESCO with $U(0, 1) < 0.75$ performs significantly better or similarly on all the CEC2013 problems when compared to that with $U(0, 1) < 0.5$. These results indicate that 0.75 is a reasonable probability to choose each sample as training sample.

7) Influence of the Threshold for Infill Solution Selection: To further study the effect of different thresholds for infill solutions selection in Section III-D, ESCO is compared with its variants with three values of l [i.e., $l = 0$, $l = \mathrm{sqrt}(10^{-3} \times d)$, and $l = \mathrm{sqrt}(10^{-9} \times d)$]. Due to page limitations, the comparison results are provided in Table A.VII of the supplementary file. The results reported in Table A.VII of the supplementary file show that ESCO with the original l value is the best among the four compared variants according to Wilcoxon′s rank-sum test. ESCO significantly outperforms the variants with $l = 0$, $l = \mathrm{sqrt}(10^{-3} \times d)$, and $l = \mathrm{sqrt}(10^{-9} \times d)$, on 12, 8, and 7 test problems, respectively. Therefore, $l = \mathrm{sqrt}(10^{-6} \times d)$ is quite suitable for ESCO.

8) Influence of the Number of Local Models: Here, in order to study the impact of the number (t) of local models built in Section III-B, Table A.VIII of the supplementary file gives the statistical results for ESCO with different t values from {3, 5, 7, 10, 15} due to page limitations. From Table A.VIII of the supplementary file, it is confirmed that $t = 10$ in our setting is reasonable and more effective, as it obtains the overall best performance when considering all the CEC2013 problems.

V. CONCLUSION AND FUTURE WORK

In this work, we have proposed an ensemble surrogate-based coevolutionary optimizer, called ESCO, for solving LSEOPs with a limited number of function evaluations. In order to construct an effective surrogate model with sufficiently high quality, ESCO adopts a selective ensemble surrogate to realize a more accurate and effective approximation for the objective values of LSEOPs. To build such a selective ensemble surrogate, several diverse local surrogate models are first trained on some data subsets selected by feature selection and roulette wheel selection. Then, ESCO selects some of the constructed local models to form the final selective ensemble model in order to further improve the prediction accuracy and efficiency. In this part, we propose a simple and effective strategy (Top-K WSS), which chooses top-k local models with the smallest KLD values and assigns them with different weights based on the KLD values to compose an ensemble surrogate. After obtaining the selective ensemble surrogate, a coevolutionary optimizer is designed to optimize the original LSEOP with the help of a simplified auxiliary problem, which can speed up the convergence of EAs. Finally, a simple yet effective infill selection criterion is proposed to identify promising infill samples with both good convergence and diversity, which helps to further enhance the prediction accuracy of the surrogate model. When compared to nine competitive SAEAs, the experiments on a number of test LSEOPs with up to 1000 dimensions have confirmed the advantages of our algorithm in most cases.

As observed from the experimental results in this article, although the proposed ESCO is able to outperform the existing competitive SAEAs for solving most test LSEOPs, this work is preliminary and we will further improve the performance for solving LSEOPs with far more than 1000 dimensions in our future work. Moreover, the application of the proposed algorithm to some real-world LSEOPs will be considered in our future work.

REFERENCES

[1] K. Sindhya, V. Ojalehto, J. Savolainen, H. Niemistö, J. Hakanen, and K. Miettinen, "Coupling dynamic simulation and interactive multiobjective optimization for complex problems: An APROS-NIMBUS case study," *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2546–2558, 2014.

[2] T. Chugh, K. Sindhya, K. Miettinen, Y. Jin, T. Kratky, and P. Makkonen, "Surrogate-assisted evolutionary multiobjective shape optimization of an air intake ventilation system," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2017, pp. 1541–1548.

[3] D. Dasgupta and Z. Michalewicz, *Evolutionary Algorithms in Engineering Applications*. Heidelberg, Germany: Springer, 2013.

[4] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[5] Q. Yang *et al.*, "A distributed swarm optimizer with adaptive communication for large scale optimization," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3393–3408, Jul. 2020.

[6] A. Song, W. Chen, X. Luo, Z. Zhan, and J. Zhang, "Scheduling workflows with composite tasks: A nested particle swarm optimization approach," *IEEE Trans. Services Comput.*, vol. 15, no. 2, pp. 1074–1088, Mar./Apr. 2022.

[7] W. Chen, Y. Jia, F. Zhao, X. Luo, X. Jia, and J. Zhang, "A cooperative co-evolutionary approach to large-scale multisource water distribution network optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 842–857, Oct. 2019.

[8] H. Wang, Y. Jin, and J. O. Janson, "Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 939–952, Dec. 2016.

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

[10] B. Liu, Q. Zhang, and G. G. E. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 180–192, Apr. 2014.

[11] N. Namura, K. Shimoyama, and S. Obayashi, "Expected improvement of penalty-based boundary intersection for expensive multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 898–913, Dec. 2017.

[12] X. Wang, G. G. Wang, B. Song, P. Wang, and Y. Wang, "A novel evolutionary sampling assisted optimization method for high dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 815–827, Oct. 2019.

[13] 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.

[14] H. Wang, Y. Jin, C. Sun, and J. Doherty, "Offline data-driven evolutionary optimization using selective surrogate ensembles," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 203–216, Apr. 2019.

[15] 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. 2009.

[16] I. Loshchilov, M. Schoenauer, and M. Sebag, "Comparison-based optimizers need comparison-based surrogates," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 2010, pp. 364–373.

[17] H. Wang and Y. Jin, "A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 536–549, Feb. 2020.

[18] A. Nesrine, B. Slim, and B. S. Lamjed, "Steady state IBEA assisted by MLP neural networks for expensive multi-objective optimization problems," in *Proc. Annu. Conf. Genet. Evol. Comput.*, 2014, pp. 581–588.

[19] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 442–458, Jun. 2019.

[20] R. G. Regis, "Particle swarm with radial basis function surrogates for expensive black-box optimization," *J. Comput. Sci.*, vol. 5, no. 1, pp. 12–23, 2014.

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

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

[23] M. Emmerich, A. Giotis, M. Özdemir, T. Bäck, and K. Giannakoglou, "Metamodel—Assisted evolution strategies," in *Proc. Int. Conf. Parallel Problem Solving Nature*, 2002, pp. 361–370.

[24] Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum, "Combining global and local surrogate models to accelerate evolutionary optimization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 1, pp. 66–76, Jan. 2007.

[25] M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single- and multi-objective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 421–439, Aug. 2006.

[26] W. Ponweiser, T. Wagner, and M. Vincze, "Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2008, pp. 3515–3522.

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

[28] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Trans. Evol. Comput.*, vol. 6, no. 5, pp. 481–494, Oct. 2002.

[29] 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.

[30] A. Sanchez, A. B. Moreno, D. Velez, and J. F. Vélez, "Analyzing the influence of contrast in large-scale recognition of natural images," *Integr. Comput.-Aided Eng.*, vol. 23, no. 3, pp. 221–235, 2016.

[31] L. Zhou, L. Zhen, R. Baldacci M. Boschetti, Y. Dai, and A. Lim, "A heuristic algorithm for solving a large-scale real-world territory design problem," *Omega*, vol. 103, Sep. 2021, Art. no. 102442.

[32] G. Calabrò, V. Torrisi, G. Inturri, and M. Ignaccolo, "Improving inbound logistic planning for large-scale real-world routing problems: A novel ant-colony simulation-based optimization," *Eur. Transp. Res. Rev.*, vol. 12, no. 1, pp. 1–11, 2020.

[33] D. Molina, A. R. Nesterenko, and A. LaTorre, "Comparing large-scale global optimization competition winners in a real-world problem," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2019, pp. 359–365.

[34] 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.

[35] F. F. Wei *et al.*, "A classifier-assisted level-based learning swarm optimizer for expensive optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 2, pp. 219–233, Apr. 2021.

[36] P. Huang, H. Wang, and W. Ma, "Stochastic ranking for offline data-driven evolutionary optimization using radial basis function networks with multiple kernels," in *Proc. IEEE Symp. Comput. Intell. (SSCI)*, 2019, pp. 2050–2057.

[37] P. Huang, H. Wang, and Y. Jin, "Offline data-driven evolutionary optimization based on tri-training," *Swarm Evol. Comput.*, vol. 60, Feb. 2021, Art. no. 100800.

[38] I. De Falco, A. D. Cioppa, and G. A. Trunfio, "Investigating surrogate-assisted cooperative coevolution for large-scale global optimization," *Inf. Sci.*, vol. 482, pp. 1–26, May 2019.

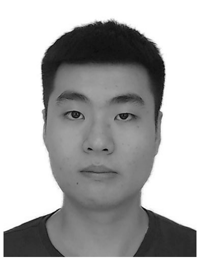
[39] G. Fu, C. Sun, Y. Tan, G. Zhang, and Y. Jin, "A surrogate-assisted evolutionary algorithm with random feature selection for large-scale expensive problems," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 2020, pp. 125–139.

[40] 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.

[41] J. Zhang, and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

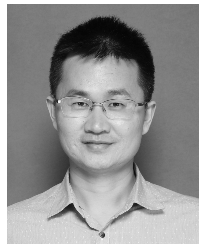
[42] M. Clerc, "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization," in *Proc. Congr. Evol. Comput.*, vol. 3, 1999, pp. 1951–1957.

[43] R. Cheng and Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Inf. Sci.*, vol. 291, pp. 43–60, Jan. 2015.
[44] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[45] J. H. Friedman and P. Hall, "On bagging and nonlinear estimation," *J. Stat. Plan. Inference*, vol. 137, no. 3, pp. 669–683, 2007.
[46] Z. Ren *et al.*, "Surrogate model assisted cooperative coevolution for large scale optimization," *Appl. Intell.*, vol. 49, no. 2, pp. 513–531, 2019.
[47] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, 2013, pp. 71–78.
[48] Y. Tang, J. Chen, and J. Wei, "A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions," *Eng. Optim.*, vol. 45, no. 5, pp. 557–576, 2013.
[49] Q. Liu, X. Wu, Q. Lin, J. Ji, and K. C. Wong, "A novel surrogate-assisted evolutionary algorithm with an uncertainty grouping based infill criterion," *Swarm Evol. Comput.*, vol. 60, Feb. 2021, Art. no. 100787.
[50] M. Stein, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.
[51] Z. H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Hoboken, NJ, USA: Chapman Hall/CRC, 2019.
[52] L. Zhuang, K. Tang, and Y. Jin, "Metamodel assisted mixed-integer evolution strategies based on Kendall rank correlation coefficient," in *Proc. Int. Conf. Intell. Data Eng. Autom. Learn.*, 2013, pp. 366–375.
[53] M. Kendall and J. D. Gibbons, *Rank Correlation Methods*. London, U.K.: Edward Arnold, 1990.
[54] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, 1951.
[55] S. I. Amari and A. Cichock, "Information geometry of divergence functions," *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 58, no. 1, pp. 183–195, 2010.
[56] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 74–88, Feb. 2019.
[57] A. Habib, H. K. Singh, T. Chugh, T. Ray, and K. Miettinen, "A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 1000–1014, Dec. 2019.
[58] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 4, pp. 115–148, 1995.
[59] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Comput. Sci. Inform.*, vol. 26, no. 4, pp. 30–45, 1996.
[60] P. N. Suganthan *et al.*, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Kanpur Genet. Algorithms Lab., IIT Kanpur, Kanpur, India, KanGAL Rep. 2005005, May 2005.
[61] H. Yu, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Surrogate-assisted hierarchical particle swarm optimization," *Inf. Sci.*, vols. 454–455, pp. 59–72, Jul. 2018.
[62] H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2664–2677, Sep. 2017.
[63] H. Yu, Y. Tan, C. Sun, and J. Zeng, "A generation-based optimal restart strategy for surrogate-assisted social learning particle swarm optimization," *Knowl. Based Syst.*, vol. 163, pp. 14–25, Jan. 2019.
[64] M. Hollander, D. Wolfe, and E. Chicken, *Nonparametric Statistical Methods*, vol. 751. Hoboken, NJ, USA: Wiley, 2013.
[65] X. Li, K. Tang, M. N. Omidvar, Z. Yang, K. Qin, and H. China, "Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization," *Gene*, vol. 7, no. 33, p. 8, 2013.



Xunfeng Wu received the B.S. degree from Shaoguan University, Shaoguan, China, in 2018, and the M.S. degree from Shenzhen University, Shenzhen, China, in 2021, where he is currently pursuing the Ph.D. degree with the College of Computer Science and Software Engineering.

His current research interests are in evolutionary computation, multiobjective optimization, and data-driven optimization.

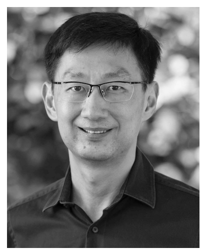


Qiuzhen Lin (Member, IEEE) received the B.S. degree from Zhaoqing University, Zhaoqing, China, in 2007, the M.S. degree from Shenzhen University, Shenzhen, China, in 2010, and the Ph.D. degree from the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, in 2014.

He is currently an Associate Professor with the College of Computer Science and Software Engineering, Shenzhen University. He has published over 60 research papers since 2008. His current research interests include artificial immune system, multiobjective optimization, and dynamic system.

Jianqiang Li (Member, IEEE) received the B.S. and Ph.D. degrees in automation from the South China University of Technology, Guangzhou, China, in 2003 and 2008, respectively.

He is a Professor with the College of Computer and Software Engineering, Shenzhen University, Shenzhen, China. He led a project of the National Natural Science Foundation, and a project of the Natural Science Foundation of Guangdong Province, China. His major research interests include embedded systems and Internet of Things.



Kay Chen Tan (Fellow, IEEE) received the B.Eng. degree (First-Class Hons.) and the Ph.D. degree from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively.

He is currently a Chair Professor (Computational Intelligence) with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. He has published over 300 refereed articles and seven books.

Prof. Tan is currently the Vice-President (Publications) of IEEE Computational Intelligence Society, USA. He served as the Editor-in-Chief for the *IEEE Computational Intelligence Magazine* from 2010 to 2013 and the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION from 2015 to 2020. He currently serves as an editorial board member for more than ten journals. He is an IEEE Distinguished Lecturer Program Speaker and the Chief Co-Editor of Springer Book Series on *Machine Learning: Foundations, Methodologies, and Applications*.

Victor C. M. Leung (Life Fellow, IEEE) received the B.A.Sc. degree (Hons.) in electrical engineering and the Ph.D. degree in electrical engineering (through the Canadian Natural Sciences and Engineering Research Council Postgraduate Scholarship) from the University of British Columbia, Vancouver, BC, Canada, in 1977 and 1982, respectively.

He is currently a Distinguished Professor of Computer Science and Software Engineering with Shenzhen University, Shenzhen, China. He is also an Emeritus Professor of Electrical and Computer Engineering and the Director of the Laboratory for Wireless Networks and Mobile Systems, The University of British Columbia (UBC), Vancouver, BC, Canada. He is named in the current Clarivate Analytics list of "Highly Cited Researchers." His research interests include wireless networks and mobile systems. He has published widely in these areas.

Dr. Leung received the 1977 APEBC Gold Medal, the 1977–1981 NSERC Postgraduate Scholarships, the IEEE Vancouver Section Centennial Award, the 2011 UBC Killam Research Prize, the 2017 Canadian Award for Telecommunications Research, the 2018 IEEE TCGCC Distinguished Technical Achievement Recognition Award, and the 2018 ACM MSWiM Reginald Fessenden Award. His coauthored articles won the 2017 IEEE ComSoc Fred W. Ellersick Prize, the 2017 IEEE SYSTEMS JOURNAL Best Paper Award, the 2018 IEEE CSIM Best Journal Paper Award, and the 2019 IEEE TCGCC Best Journal Paper Award. He is serving on the editorial boards for IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, IEEE ACCESS, IEEE NETWORK, and several other journals. He is a Fellow of the Royal Society of Canada (Academy of Science), the Canadian Academy of Engineering, and the Engineering Institute of Canada.