Offline Data-Driven Multiobjective Optimization: Knowledge Transfer Between Surrogates and Generation of Final Solutions

Cuie Yang, Jinliang Ding, *Senior Member, IEEE*, Yaochu Jin, *Fellow, IEEE*, and Tianyou Chai, *Fellow, IEEE*

***Abstract*—In offline data-driven optimization, only historical data is available for optimization, making it impossible to validate the obtained solutions during the optimization. To address these difficulties, this paper proposes an evolutionary algorithm assisted by two surrogates, one coarse model and one fine model. The coarse surrogate (CS) aims to guide the algorithm to quickly find a promising subregion in the search space, whereas the fine one focuses on leveraging good solutions according to the knowledge transferred from the CS. Since the obtained Pareto optimal solutions have not been validated using the real fitness function, a technique for generating the final optimal solutions is suggested. All achieved solutions during the whole optimization process are grouped into a number of clusters according to a set of reference vectors. Then, the solutions in each cluster are averaged and outputted as the final solution of that cluster. The proposed algorithm is compared with its three variants and two state-of-the-art offline data-driven multiobjective algorithms on eight benchmark problems to demonstrate its effectiveness. Finally, the proposed algorithm is successfully applied to an operational indices optimization problem in beneficiation processes.**

***Index Terms*—Knowledge transfer, multiobjective evolutionary algorithms (MOEAs), multisurrogate, offline data-driven optimization.**

Manuscript received August 24, 2018; revised December 17, 2018 and February 28, 2019; accepted June 18, 2019. Date of publication July 1, 2019; date of current version May 29, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61525302 and Grant 61590922, in part by the Project of Ministry of Industry and Information Technology of China under Grant 20171122-6, in part by the Projects of Shenyang under Grant Y17-0-004, in part by the Fundamental Research Funds for the Central Universities under Grant N160801001 and Grant N161608001, in part by the National Key Research and Development Program of China under Grant 2018YFB1701104, in part by the Royal Society under Grant IEC\NSFC\170279, and in part by the Outstanding Student Research Innovation Project of Northeastern University under Grant N170806003. *(Corresponding authors: Jinliang Ding; Yaochu Jin.)*

C. Yang, J. Ding, and T. Chai are with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China (e-mail: cuieyang@outlook.com; jlding@mail.neu.edu.cn; tychai@mail.neu.edu.cn).

Y. Jin is with the Department of Computer Science, University of Surrey, Guildford GU27 XH, U.K., and also with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China (e-mail: yaochu.jin@surrey.ac.uk).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2019.2925959

I. Introduction

IN SOLVING real-world optimization problems, computationally intensive numerical simulations or physical experiments often need to be conducted to evaluate the objective functions, e.g., in integrated circuit design [1], antenna design [2], hybrid vehicle control [3], or aerodynamic design optimization [4], [5]. In many other situations, only historical data are available for optimization [6]–[10], where the optimization problem is solved on the basis of collected data without resorting to any physical models. These optimization problems are often referred to data-driven optimization problems [6], [11], which can be divided into online and offline data-driven optimization problems [6]. In the online data-driven optimization, a small number of candidates can be validated during the optimization with the real objective functions (obtained by either performing numerical simulations or experiments), and the newly generated data can also be used to update the surrogates. By contrast, in the offline data-driven optimization, no new data can be generated and only the historical data can be exploited to manage the surrogates [7], [9].

Most data-driven optimization relies on surrogate models to assist the optimizer to guide the search [11]–[13]. Many machine learning methods can be employed to construct surrogates, including artificial neural networks (ANNs) [14], [15], polynomial regression (PR) [16], support vector machines (SVMs) [17], radial basis function (RBF) networks [18]–[20], and Gaussian processes (GPs). GPs are also known as Kriging or design and analysis of computer experiment models [21]–[23].

Multisurrogate methods have been investigated in data-driven optimization due to their promising performance. One intuitive idea of multiple models is to use ensembles, which have been shown to be able to improve the accuracy and reliability of the estimated fitness [13], [24]. Ensembles may consist of homogeneous [25]–[27] or heterogeneous base models [28], [29]. Different from ensembles, multiple surrogates have also been used to exploit the balance between "curse of uncertainty" and "bless of uncertainty," typically with the help of a global model and a local model. The global surrogate model aims to capture the global profile of the fitness landscape by smoothing out the local optima, thereby helping the optimizer explore the search space. By contrast, the local surrogate model is constructed around the promising

region found by the current population to exploit the local details of the fitness landscape [28], [30], [31]. The global and local surrogates can also be used in parallel during the optimization. For instance, in [20], each of the surrogates is trained separately using different training sets. Then a new candidate is evaluated by a selected surrogate that has the least prediction error. A global RBF surrogate model was combined with a local fitness estimation method [32] in a hierarchical particle swarm optimization (PSO) algorithm [33]. Most recently, a surrogate-assisted cooperative PSO was suggested to solve high-dimensional data-driven optimization problems [34], where a social learning PSO assisted by a global RBF works with a PSO assisted by the local fitness estimation method.

Despite the great success achieved in online data-driven optimization, little work has been dedicated to more challenging, offline data-driven optimization problems with few exceptions. In [7], a low-order PR model and a GP model are constructed, where the low-order PR is used as the real fitness function to produce new data during search process for model management, while the GP model is employed as the surrogate to assist the evolutionary search. In [9], a large number of base learners are built offline and then a subset of these base learners are adaptively selected to make sure that the surrogate is able to best approximate the local fitness landscape. Note that the algorithm reported in [6] is developed for optimization driven by a large amount of data, which is meant for reducing the computation time by using less data by means of adaptive clustering.

This paper aims to address the following two main challenges in offline data-driven multiobjective optimization where very limited historical data are available.

1) How to design surrogate models using the limited historical data only that are able to correctly guide the search? In offline data-driven optimization, building reliable surrogates plays an essential role in exploiting promising solutions because the optimization is guided by the surrogates only and no new data can be generated during the search for updating or validating the surrogates.
2) How to select solutions to be implemented when the optimization is completed. Recall that the surrogates built offline have not been updated during the search and the "optimal" solutions achieved at the end of the optimization may not be really optimal due to the approximation errors in evaluating the objectives. Compared to conventional multiobjective optimization problems (MOPs), it is even trickier for the user to select solutions to be implemented at the end of offline data-driven multiobjective optimization.

To tackle the above challenges, this paper proposes a multisurrogate approach with knowledge being transferred between the surrogates to improve the reliability of the surrogates. In addition, a method for selecting final solutions with the help of a set of reference vectors (RVs) is designed to improve the reliability of the solutions. The main contributions of this paper can be summarized as follows.

1) A multisurrogate model consisting of a coarse surrogate (CS) and a fine surrogate (FS) is proposed to guide the evolutionary optimization. The CS, which is constructed in a low-dimensional subspace using a simple structure, is meant to capture the global profile of the fitness landscapes. The FS, by contrast, aims to exploit promising solutions by reusing the knowledge transferred from the course model. It should be noted that the proposed multisurrogate model is different from most existing multisurrogate models [25], [28] in that the coarse model in this paper is constructed in a subspace of the original search space. The potential benefits of constructing a low-dimensional surrogate are twofold. First, the low-dimensional surrogate is expected to improve the model accuracy given limited training data, especially for high-dimensional problems. Second, the low-dimensional surrogate is able to smooth out some local minimums so that it can more easily capture the main profile of the fitness landscape.
2) A knowledge transfer technique is introduced to transfer the knowledge acquired by the CS to the FS. After the coarse model locates the promising areas, the fine model will exploit this knowledge to more accurately identify the optimums and accelerate the convergence, thereby enhancing the search efficiency.
3) An RV-based method is proposed to generate the final solution set. At first, all candidate solutions, which are achieved in each generation of the fine search, are grouped into different clusters according to a set of RVs. Then, one solution is generated by averaging those in each cluster and all these averaged solutions will be the final solution set. Our empirical results indicate that the quality of these solutions is better than that of the nondominated solutions obtained in the last generation.

The rest of this paper is organized as follows. Section II briefly introduces offline data-driven optimization problems and knowledge transfer techniques in evolutionary optimization. In Section III, the proposed offline data-driven multiobjective optimization algorithm is described in detail. The comparative experimental results on the benchmark problems are presented in Section IV, followed by an application of the proposed algorithm to operational indices optimization of a beneficiation process in Section V. Section VI concludes this paper with a summary and a discussion for future work.

II. Related Work

A. Offline Data-Driven Optimization Problems

Offline data-driven optimization problems widely exist in the real world [11], [35], such as trauma systems design [6], performance optimization of fused magnesium furnaces [7], and operational indices optimization of beneficiation processes [8], in which the objective functions cannot be directly calculated using mathematical equations and only data are available for fitness evaluations [11], [35]. Offline data-driven optimization starts with a certain amount of collected data, which are used to construct surrogates for searching optimal solutions [9]. During the optimization, surrogate models can be updated to improve the search efficiency either by using generated synthetic data from other surrogates, reusing

knowledge collected from the optimization, or newly collected data that are not under the control of the optimizer. An example of model management techniques using information during the optimization can be found in [9], which adaptively selects a subset of the base learners of an ensemble at each iteration according to the location of the best individual. Once the surrogate-assisted optimization is completed, the best solution (set) will be implemented to solve the real-world problem.

B. Multiobjective Optimization Problems

Many real-world problems involve multiple conflicting objectives to be optimized simultaneously, which are commonly referred to as MOPs. This paper considers the following box-constrained MOP:

$$\begin{aligned} \min \quad & F(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_M(\mathbf{x})) \\ \text{s.t.} \quad & \mathbf{x} \in [\mathbf{l}, \mathbf{u}] \end{aligned} \tag{1}$$

where $\mathbf{l}$ and $\mathbf{u}$ are the lower and upper bounds of the search space, $(f_1(\mathbf{x}), \ldots, f_M(\mathbf{x}))$ is the objective vector containing M objectives, and $\mathbf{x} = (x_1, \ldots, x_n)$ are the decision variables.

Due to the conflicting nature of the objectives, no single solution is able to optimize all objectives at the same time. Instead, a set of tradeoff optimal solutions can be found, which is called the Pareto set (PS) and its image in the objective space is known as the Pareto front (PF). Evolutionary algorithms (EAs) are well suited for MOPs because maintaining a population of candidates enables EAs to explore a set of diversified optimal solutions to approximate the PF. A large number of multiobjective EAs (MOEAs) have been proposed in the last decades, which can generally be classified into Pareto domination-based approaches [36], indicator-based approaches [37], and decomposition-based approaches [38].

C. Knowledge Transfer in Optimization

Knowledge transfer refers to sharing knowledge and providing inputs to improve problem solving [39]. Knowledge acquisition from the evolutionary optimization process and reuse of the acquired knowledge in EAs have been shown to be effective in speeding up convergence as well as enhancing the quality of the obtained optimal solutions [40], [41]. One class of knowledge transfer techniques transfers the knowledge of the tasks similar to the current task to prevent a cold start of EAs. For example, one early work on knowledge transfer uses the optimal solutions of the past similar problems as the shared knowledge, and then incorporate them into the initial population in solving the current task [42]–[44]. Furthermore, the techniques of reusing knowledge from heterogeneous problems are investigated [45]–[48]. The main idea is to transfer the structured knowledge learned from the previous tasks to the current task. In addition, in model-based optimization algorithms, such as estimation of distribution algorithms (EDAs), the distance distribution information about similar previous tasks is considered as knowledge and combined with the current task to improve the efficiency of model construction [49], [50]. In genetic algorithms, the building blocks of related problems are reused in the current optimization task [51]–[53].

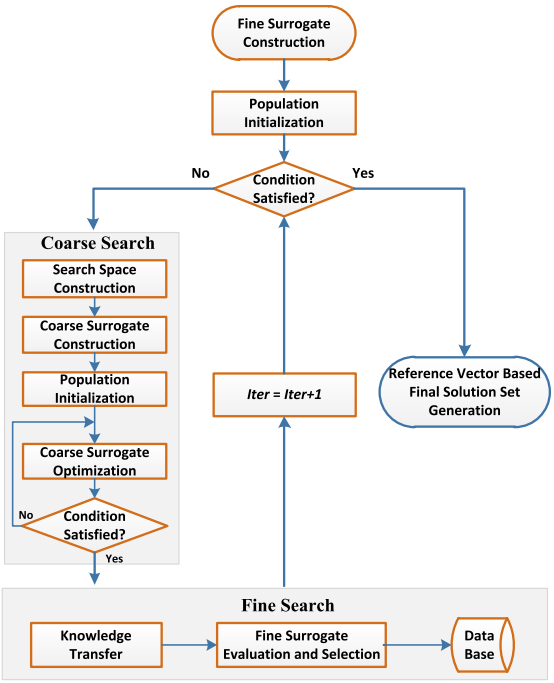

Fig. 1. Framework of the proposed offline data-driven multiobjective optimization.

Most recently, multifactorial EAs (MFEAs) [54]–[56] have been proposed to transfer knowledge among tasks to be solved in parallel. MFEAs allow knowledge sharing via two techniques, namely, assortative mating and vertical cultural transmission, to ensure that the tasks can not only maintain their own specific knowledge but also achieve knowledge from other tasks [54].

Inspired by the success of knowledge transfer in evolutionary optimization, this paper employs the knowledge transfer technique to transfer knowledge between two surrogates to enhance the capability of exploiting promising solutions. Both two surrogates are trained using the historical data and have shared knowledge in nature.

III. Proposed Algorithm

A. Overall Framework

A generic diagram of the proposed algorithm is presented in Fig. 1. Before the optimization starts, a data set of the problem to be solved, (X, Y), is collected to train the surrogates, where X represents the decision variables, and Y the corresponding objective values. The algorithm starts with construction of the FS, which remains unchanged in the following optimization process. The widely used RBF model [18]–[20] is employed to build the FS in this paper. Afterwards, a population consisting of N individuals is randomly initialized, denoted as PF_0, which will be evaluated using the FS.

At each iteration (*Iter*) of main optimization loop, a CS, which is a PR model, is constructed in a subspace of the search space and a search assisted by the CS will be carried out to find the promising regions of the search space. Then a fine search exploiting the knowledge transferred from the coarse

search is conducted. The nondominated solutions found during the fine search at each iteration are stored in a DB.

The coarse-fine search process continues until some termination condition is met. Then, an RV-based solution generation strategy is performed to produce a set of final solutions for the user to implement using the solutions stored in *DB*. In the following sections, we will detail the three main components of the proposed algorithm, i.e., the coarse search, the fine search and the RV-based final solution set generation.

B. Coarse Search

In the coarse search, the CS is built in a subspace of the original search space. As shown in Fig. 1, the coarse search is composed of four steps, namely, determination of the subspace of the CS, construction of CS, population initialization and a number of iterations of search assisted by the CS. Note that the search space of the coarse search is the input space of the CS, denoted by *DL*, which is controlled by a coefficient *rv* as follows:

$$DL = \lceil rv \times D \rceil \tag{2}$$

where D is the dimension of the original optimization problem, i.e., $DL \leq D$.

For the sake of simplicity, *DL* decision variables are randomly chosen from X and denoted as XL. Then a new data set (XL, Y) is used to train the CS. In this paper, a second-order PR is adopted for the CS, which aims to smooth out local optima. It is worth noting that the search space of the CS changes at each iteration of the main loop so that the coarse search is able to find promising regions in different subspaces during the optimization.

Any MOEA assisted by the CS can be employed for the coarse search. The maximum number generations *TC* is taken as the termination condition of the coarse search. When the optimization terminates, individuals of the last generation will be stored in *DB*. Algorithm 1 details the coarse search procedure.

Algorithm 1 Coarse Search

1: **Input:** $rv \leq 1$: A coefficient controls the dimension of coarse model; (X, Y): The historical data set; *TC*: The maximum number of generations for coarse search;
2: **Output:** P_{TL}: The population at the last generation of the coarse search; *Dind*: Index of the selected decision variables for the coarse model;
3: Calculate the dimension of coarse model $DL = \lceil rv \times D \rceil$;
4: Randomly select *DL* decision variables from the variable vector and record their index as *Dind*;
5: *XL*=*X*(:,*Dind*) //take out the selected variables and denoted as *XL*;
6: Training the PR model using (*XL*,*Y*) as the CS;
7: Initialize the population for the coarse search: create the initial population P_0 with N individuals; randomized individuals, each individual has *DL* dimension;
8: /* Coarse surrogate assisted optimization */
9: **while** $t < TC$ **do**
10: Generate the parent population from P_t using tournament selection;
11: Create the offspring population Q_t by applying crossover and mutation on the parent population;
12: Combine the parent population P_t and the offspring population Q_t as P_t;
13: Perform environmental selection on P_t to select the parent population for the next generation P_{t+1};
14: $t = t + 1$;
15: **end while**
16: $P_{TL} = P_{t+1}$;

C. Fine Search

In this paper, the FS is trained to accurately approximate the real fitness functions. However, an accurate surrogate model is not necessarily always be desirable in optimization since the real fitness functions typically have multiple local optima, making the search more challenging. Thus, the knowledge, specifically the solutions, acquired during the coarse search is expected to be able to enhance the search performance of FS by reducing the likelihood of getting stuck in a local optimum, since we hypothesize that the CS and FS are correlated yet the CS is smoother. For this reason, a knowledge transfer is employed to transfer the knowledge of CS to FS during the search. This paper borrows the transfer strategy proposed in [54], which realizes knowledge transfer among optimization tasks during offspring production. Details of the fine search are described in Algorithm 2.

Steps 3–9 of Algorithm 2 describe steps for knowledge transfer between the CS and FS. As mentioned above, the dimension of the FS is the same as the original optimization problems, while the dimension of CS is a subspace of the original problem, which changes from generation to generation of the coarse search. For this reason, we first convert the population P_{TL} of coarse search to an intermediate population *PL*, whose dimension is same as that of the fine search, as shown in steps 3–6 in Algorithm 2. Specifically, population *PL* maintains the decision variables of P_{TL} of the coarse search, and copies the remaining variables from the current population of

Algorithm 2 Fine Search

1: **Input:** PF_{iter}: Current population of the fine search; P_{TL}: the population of the coarse search; *Dind*: Index of the selected decision variables for coarse search; N: Population size;
2: **Output:** PF_{iter+1}: Population of fine search in the next iteration;
3: Set $PL = PF_{iter}$;
4: **for** i=1:N **do**
5: Set $PL(i,Dind)=P_{TL}(i,:)$;
6: **end for**
7: $P = PL \cup PF_{iter}$;
8: P = randomly shuffled P;
9: Q=crossover + mutation(P);
10: $P = Q \cup PF_{iter}$;
11: PF_{iter+1} = environmental selection(P);
12: Add PF_{iter+1} to *DB*;

the fine search, PF_{iter}. After that, PL and PF_{iter} are combined as the parent population P (step 7), which serves as the mating pool after being randomly shuffled (step 8). In the next step, new offspring Q is generated by applying crossover and mutation on the parent individuals. Once the fitness of all offspring individuals in Q is calculated using FS, the popular environment selection method in NSGA-II [36] is employed to select N individuals from the combination of Q and the current population PF_{iter}. The selected solutions are passed to the next generation and also stored in DB.

As described above, the combined population contains individuals from two different populations, PL and PF_{iter}. Thus in reproduction, one of the following three cases may occur for the two parent individuals for crossover, i.e., both are from population PL, both from population PF_{iter}, or one from PL and the other from PF_{iter}. In the last case, the solutions of CS are implicitly transferred to the offspring of FS, thus FS is able to automatically gain useful knowledge from CS during environmental selection. In this way, the fine search is able to exploit beneficial knowledge from the coarse search to accelerate the search process.

D. Reference Vector-Based Final Solution Set Generation

In online data-driven optimization, the final solution (set) can be relatively easily determined since all solutions are evaluated using the real objective functions. In offline optimization, however, the observed objective values of candidate solutions are all predicted by the surrogates, which may not be correct due to the approximation errors introduced by the surrogates. Our empirical results indicate that selecting the nondominated solutions only for final implementation may not be the best choice in offline data-driven optimization. Instead, we propose an RV-based final solution set generation strategy to obtain a solution set for final implementation. The main idea is to group the solutions achieved at each generation of the fine search and then generate one solution by averaging the solutions in each group. Recall that all solutions obtained at each generation of the fine search are stored in a database DB. The underlying motivation of clustering is to reduce the errors introduced by surrogate models by averaging a number of similar solutions within a cluster.

The question now is how to group the solutions stored in DB. For diversity of the solutions, this paper adopts a set of RVs for final solution generation, which have been widely adopted to guide the search process in many-objective optimization [57], [58]. The final solution generation method starts with categorizing the solutions in DB into a number of groups. We assume that for each group, ρ solutions will be used for averaging, where ρ is a parameter to be specified. Grouping solutions in DB consists of two steps. In the first step, a set of RV is generated and each solution in DB is associated to one RV, as proposed in RVEA [58]. The resulting cluster is denoted as C_j, $j = 1, 2, \ldots, K$, where K is the number of the RVs predefined by the user to indicate a set of preferred solutions. Note that similar to RVEA, all objective values are normalized to $[0, 1]$ before being clustered.

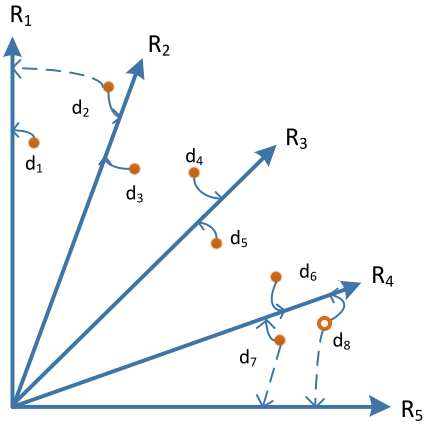

Fig. 2. Illustration example of RV-based clustering, where eight solutions are grouped into five clusters. A solid arrow line indicates that a solution is associated with an RV in the first stage, while a dashed arrow line means that a solution is added to a cluster of an RV in the second step. Finally, a circle means that a solution is removed from a cluster in the second step.

The second step aims to ensure that each cluster contains ρ solutions.

The second step is carried out due to the fact that there are might be too few or too many solutions in one cluster created in the first step. In case there are too few solutions in a cluster, i.e., smaller than the predefined number ρ, solutions in the neighboring RVs will be added to the cluster until ρ are found. In case there are too many solutions in a cluster, it is desirable to remove those inferior candidates from the cluster so that they are not used in averaging. Here, for a minimization problem, solutions that are far from the origin in the objective space will be removed.

Fig. 2 presents an illustrative example of the clustering process, where eight solutions, d_1–d_8, need to be divided into five clusters represented by five RVs R_1–R_5. In the figure, a solid arrow shows the association relationship of a solution to an RV determined in the first step. For example, solution d_1 is associated with R_1, d_2 and d_3 are associated to R_2, and d_6–d_8 are associated to R_4. However, no solution is associated to R_5. Suppose the required number of solutions in each cluster is defined to be $\rho = 2$, then the cluster for R_1 has one solution less and the cluster for R_5 lacks two solutions. Consequently, solution d_2, which is the closest to R_1 in terms of its angle with the RV, will be added to the cluster for R_1. Similarly, solutions d_7 and d_8 are added to the cluster for R_5. By contrast, the cluster for R_4 has too many solutions and one of them should be deleted. Since the Euclidean distance of solution d_8 to the origin is the largest among the three solutions in the cluster for R_4, d_8 will be removed.

Once ρ solutions are associated to each cluster, the average of the ρ solutions in each cluster will be considered to be a final solution. The whole process of final solution generation is presented in Algorithm 3.

In the proposed final solution set generation method, all solutions achieved during the optimization will participate in the generation of the final solutions, although some of them may be eventually discarded and others may contribute more than once. In addition to the RV-based method suggested in

Algorithm 3 RV-Based Solution Set Generation

Input: K: The number of reference vectors; AF: Objective values of the solutions stored in DB; AS: Decision variables of the solutions in DB; C_j, $j = 1, 2, ..., K$: The initial clusters created by reference vector association; ρ: Predefined number of solutions in each cluster
Output: The generated solution set TS for final implementation;
/* Clusters construction */
for j=1:$|K|$ **do**
  /* The first case */
  if $|C_j|$ is smaller than ρ **then**
    $I = \underset{i\in\{1,2,...,|AF|\}}{\text{descendsort}} \cos\theta_{i,j}$;
    Construct a cluster around j-th reference vector: $C_j = \{I_1, I_2, ..., I_\rho\}$;
    /* The second case */
  else
    for i=1:$|C_j|$ **do**
      Calculate the Euclidean distance to the origin: $d_i = \|C_{j,i}\|$;
    end for
    $I = \underset{i\in\{1,2,...,|C_i|\}}{ascendsort} \; d_i$;
    Find individuals with a smaller Euclidean distance: $C_j = \{I_1, I_2, ..., I_\rho\}$;
  end if
end for
/* Generate the final solution */
for j=1:K **do**
  Average the decision variables: $TS = TS \cup mean(AS\{C_j\})$;
end for

this paper, other ideas, e.g., using other clustering methods rather than the reference-based method, or using a better subset of the solutions for clustering, can also be used. Therefore, in the empirical studies, we compare the RV-based clustering method with the k-means clustering method, and also test a strategy that only uses the nondominated solutions to generate the final solution set. The results, as listed in Tables S.B and S.C in Appendix A of the supplementary material, demonstrate that the RV-based clustering method using all solutions in clustering achieves the best overall performance.

IV. Simulation Results and Discussion

In this section, empirical studies are conducted to verify the performance of the proposed offline data-driven multiobjective optimization algorithm, MS-RV for short. To this end, eight test problems are adopted for comparison, as listed in Table S.C in Appendix B of the supplementary material, of which four are taken from the DTLZ test suite [59] and the rest four (F1–F4) from [60]. Similar to [22], the value 20π within cosine in the original DTLZ1 and DTLZ3 is changed to 2π and denoted as DTLZ1a and DTLZ3a, meanwhile, the parameter α in DTLZ4 is also changed from 100 to 10, which is named as DTLZ4a, to reduce the complexity of the problem.

TABLE I
Definition of Three Variants of MS-RV

Algorithm	Definition
FS-RV	Fine surrogate only with the final solution set being generated by reference vector based clustering
MS-ND	Coarse and fine surrogates with the final solution set being generated by non-dominate sort
FS-ND	Fine surrogate with the final solution set being generated by non-dominate sort

Among the test problems, the decision variables of the DTLZ problems are independent of each other, while the variables in the F test suite are linearly correlated. Regarding the number of decision variables, D =10, 30, and 50 are considered for all the instances. Regarding the number of objectives of the DTLZ test problems, $M = 2$ and $M = 3$ are considered.

In the experiments, we first examine the efficiency of the coarse-fine search using the RV-based final solution set generation strategy (MS-RV). Then, we compare MS-RV with two offline surrogate-assisted multiobjective optimization algorithms using NSGAII_GP [7] and K-RVEA [10] as the basic search method, respectively, which is proposed in [61]. To investigate the importance of the two strategies in MS-RV, MS-RV is also compared with its three variants, FS-RV, MS-ND, and FS-ND, as listed in Table I.

In all algorithms, RBF is constructed using the toolbox in [62] and GP is built using the toolbox in [63].

A. Experimental Settings

The general and the specific parameters for each algorithm used in the experiments are summarized as follows.

1) All the compared algorithms adopt NSGA-II [36] as the basic search method. The distribution indexes of both crossover and mutation in NSGA-II are set to 20. The crossover probability and mutation probability are set to pc = 1.0 and pm = $1/D$, respectively, where D is the number of decision variables of the original optimization problems.
2) The initial training data set (historical data set) for each experiment are sampled using the Latin hypercube sampling method [64] and the size is set to $11D$, where D is the number of decision variables. The population size N is set to 50. For each algorithm, 100 final solutions are generated for bi-objective and 105 for tri-objective instances. The number of candidates in each cluster in the RV-based final solution set generation is set to $\rho = 20$. Twenty independent runs are performed for each algorithm on each test instance.
3) The termination condition of each run is the maximum number of generation for fine search $TF = 40$.
4) The maximum number of generations for coarse in each generation of fine search is set to $TC = 15$. The coefficient of rv, which determines the dimension of CS, is set to 0.3.
5) In NSGAII_GP, the parameters are the same as the original algorithm except for the maximum number of

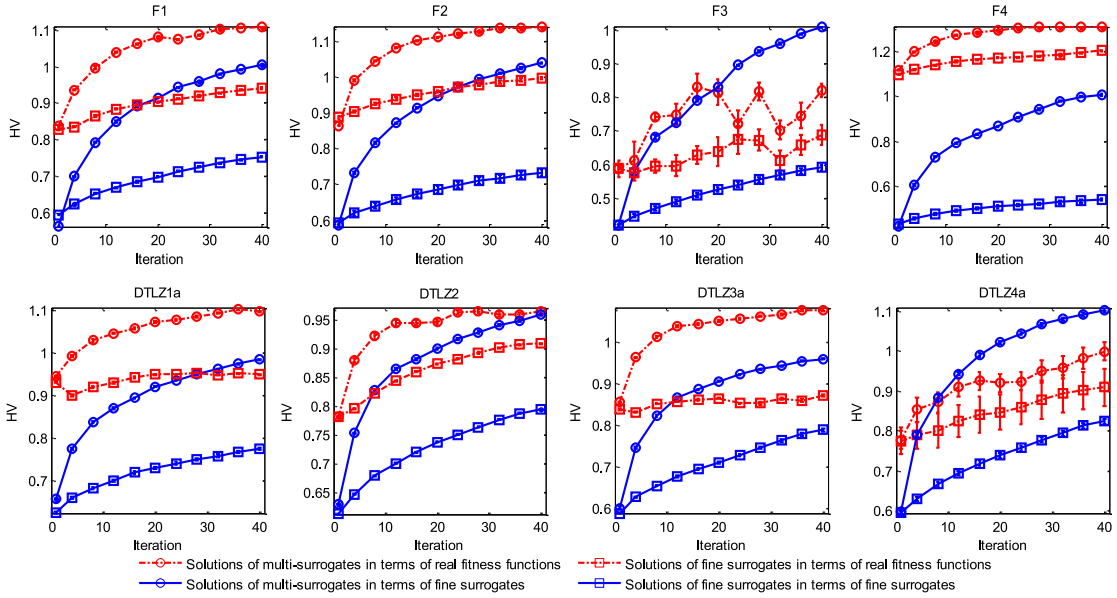

Fig. 3. Comparison of the HV values of the solution set over the generations, which are calculated based on the real objective functions and the FS, respectively.

generations, which is set to 20 for bi-objective and 21 for tri-objective problems in order to obtain the same number of final solutions.

6) The parameters of K-RVEA are set the same as in [10].

Inverted generational distance (IGD) [60] and hypervolume (HV) [65], [66] are adopted to measure quality of the solution sets in terms of both convergence and diversity. In calculating IGD, 500 uniformly distributed reference solutions are sampled from the PF. To calculate HV, all solution sets are combined and their objective values are normalized to $[0, 1]$. Then $y^* = (1.1, 1.1)$ and $y^* = (1.1, 1.1, 1.1)$ are set to the reference point for bi-objective and tri-objective test instances, respectively.

B. Experimental Studies

1) Effectiveness of the Proposed Coarse-Fine Search: In this section, we examine the efficiency of the coarse-fine search component in MS-RV in solving offline data-driven problems via in comparison with single fine search (the single FS assisted optimization) using eight test instances DTLZ1a ($M = 2, D = 30$), DTLZ2 ($M = 2, D = 30$), DTLZ3a ($M = 2, D = 30$), DTLZ4a ($M = 2, D = 30$), F1 ($D = 30$), F2 ($D = 30$), F3 ($D = 30$), and F4 ($D = 30$). For the solutions of each generation, we calculate the HV of solutions according to the objective values evaluated using the surrogates and the real objective functions, respectively. The mean and standard of HV values over 30 runs over the generations on each test problem are plotted in Fig. 3. Note that we do not compare the exact HV between the surrogate and its corresponding real objective functions as they are normalized into different region in calculating HV. Instead, we just want to note that the HV values calculated according to the real and estimated fitness values during the optimization are strongly correlated.

As can be observed from Fig. 3, the HV of the solution set in the last generation are better (validated by real objective functions) compared with the initial population on all the test problems. These results indicate that the search assisted by an FS or a multisurrogate constructed on the historical data is able to find solutions better than those in the historical data. In addition, we can find that, the HV values of solution sets according to the surrogates and the real objective functions show a consistent trend on all test instances except for F3, which exists a slight fluctuation at the later generations. The above finding implies that the better solutions evaluated by the surrogates will usually have better objective values if evaluated using the real objective functions. It can also be found in the figure that the HV values of solution sets achieved by the coarse-fine search are significantly better than those found by the single fine model according to both FSs and real fitness functions, indicating a fast convergence achieved by the coarse-fine search compared to the single FS. The encouraging convergence speed of the coarse-fine search might be attributed to the knowledge transferred from the CS to the fine model.

2) Effectiveness of Reference Vector-Based Final Solution Set Generation: As mentioned above, the nondominated solutions in the *DB* may not be the best solutions when evaluated using the real objective functions because of the approximation errors introduced by the surrogates. For this reason, we should not directly use those nondominated solutions in *DB*. Instead, we propose an RV-based strategy to generate a set of final solutions as described in Section III-D. This section investigates the effectiveness of the RV-based final solution generation method by comparing the quality of the solutions generated using the RV-based method with that of the nondominated solutions in *DB* on the eight test instances.

In this experiment, the solutions generated during the optimization during the 60 iterations are stored in the *DB*. Then two sets of solutions, each consisting of 100 solutions,

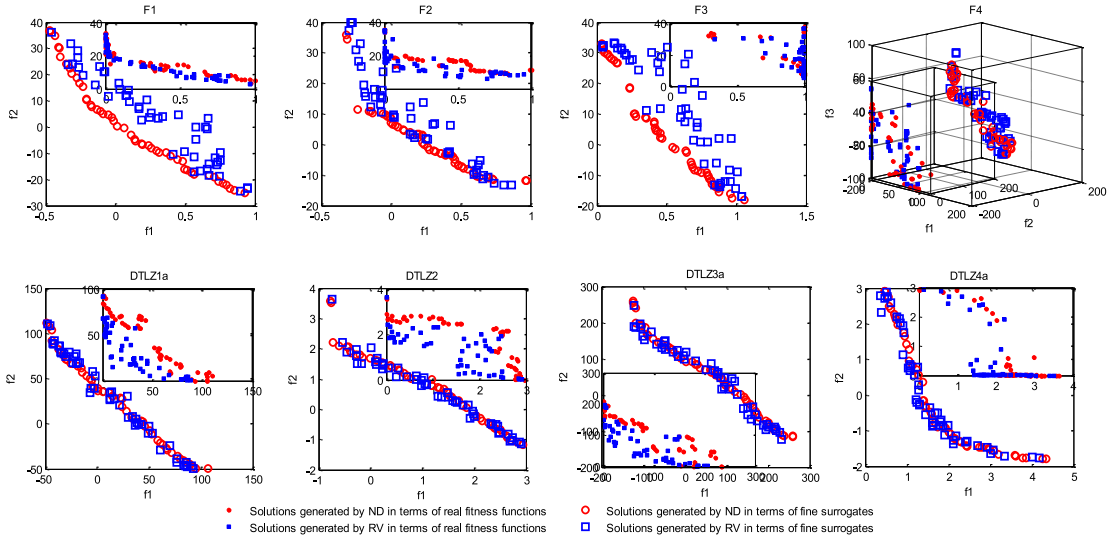

Fig. 4. Visualization of the solutions generated by ND and RV-based generation strategy, where the solutions generated by ND are denoted by circles and solutions generated by RV-based strategy are denoted by squares.

TABLE II
STATISTICAL RESULTS OF IGD OF THE COMPARED ALGORITHMS ON F TEST INSTANCES OVER 30 INDEPENDENT RUNS, WHERE THE BEST RESULT ON EACH TEST INSTANCE IS HIGHLIGHTED

Prob	Dec.	K-RVEA	NSGAII_GP	FS-ND	MS-ND	FS-RV	MS-RV
F1	D=10	**1.162e+00 (1.161e−01)** +	4.280e+00 (3.096e−01) ≈	8.844e+00 (1.424e+00) −	5.966e+00 (1.024e+00) ≈	6.427e+00 (5.236e−01) ≈	3.893e+00 (5.010e−01)
	D=30	7.427e+00 (3.324e−01) −	9.258e+00 (3.569e−01) −	7.786e+00 (5.012e−01) −	3.284e+00 (1.728e−01) ≈	7.602e+00 (4.353e−01) −	2.703e+00 (1.456e−01)
	D=50	9.482e+00 (3.139e−01) −	9.972e+00 (3.835e−01) −	9.794e+00 (3.215e−01) −	4.735e+00 (2.255e−01) −	8.421e+00 (2.430e−01) −	**3.666e+00 (1.777e−01)**
F2	D=10	4.513e+00 (3.817e−01) ≈	6.872e+00 (3.554e−01) ≈	7.167e+00 (6.565e−01) ≈	5.860e+00 (6.282e−01) ≈	7.516e+00 (4.935e−01) ≈	4.522e+00 (4.643e−01)
	D=30	2.199e+01 (6.046e−01) −	1.186e+01 (5.073e−01) −	9.072e+00 (4.262e−01) −	4.667e+00 (3.175e−01) ≈	8.909e+00 (3.858e−01) −	4.252e+00 (3.604e−01)
	D=50	2.728e+01 (7.625e−01) −	1.185e+01 (5.245e−01) −	1.277e+01 (4.422e−01) −	7.148e+00 (2.216e−01) −	1.111e+01 (4.364e−01) −	**5.184e+00 (2.082e−01)**
F3	D=10	**2.502e+00 (1.876e−01)** +	5.964e+00 (6.698e−01) ≈	9.264e+00 (6.825e−01) −	5.623e+00 (3.815e−01) ≈	7.082e+00 (3.486e−01) −	4.633e+00 (2.608e−01)
	D=30	1.067e+01 (3.683e−01) −	1.049e+01 (4.354e−01) −	1.191e+01 (3.297e−01) −	7.059e+00 (4.004e−01) −	1.005e+01 (4.703e−01) −	**4.634e+00 (2.088e−01)**
	D=50	1.317e+01 (4.273e−01) −	1.377e+01 (3.958e−01) −	1.420e+01 (5.786e−01) −	7.166e+00 (2.786e−01) ≈	1.176e+01 (4.713e−01) −	5.782e+00 (3.026e−01)
F4	D=10	2.088e+00 (1.283e−01) ≈	4.901e+00 (3.786e−01) −	6.754e+00 (8.021e−01) −	4.107e+00 (4.113e−01) ≈	6.215e+00 (5.803e−01) −	2.577e+00 (2.478e−01)
	D=30	3.249e+01 (9.182e−01) −	3.443e+01 (1.399e+00) −	3.724e+01 (2.162e+00) −	1.527e+01 (8.209e−01) ≈	3.318e+01 (2.179e+00) −	1.321e+01 (6.326e−01)
	D=50	6.457e+01 (1.365e+00) −	3.873e+01 (1.537e+00) −	7.978e+01 (2.567e+00) −	4.287e+01 (1.764e+00) −	7.185e+01 (2.281e+00) −	**2.734e+01 (1.328e+00)**
−/+/≈		8/2/2	9/0/3	11/0/1	4/0/8	10/0/2	

are generated by nondominated sort (ND) and RV-based generation method, respectively. The objective values of these solutions validated by the real fitness functions and the FSs, respectively, are plotted in Fig. 4. From the figure, we can make the following observations. First, the quality of these two sets of solutions are very different according to the surrogates and the real fitness functions. More specially, the solutions achieved by RV is no worse than those obtained by ND on all test problems when validated by the real fitness functions, although they appear to be worse according to the surrogates. This observation suggests that it might not be a good idea to directly present the nondominated solutions found (according to the surrogates) in the last generation of the coarse-fine search to the user for possible implementation. Second, the solutions obtained RV on the F problems and DTLZ problems are either comparable or much better than those obtained ND. The promising performance of solutions achieved by RV may be attributed to the fact that averaging solutions in each cluster may reduce the approximation error introduced by the surrogates, thereby enhancing the quality of the solutions.

From the result in Fig. 4, we can also see that the performance of the solutions obtained by RV on the DTLZ test functions is consistently much better than that on F1–F4

TABLE III
STATISTICAL IGD OF THE COMPARED ALGORITHMS ON DTLZ TEST INSTANCES OVER 30 INDEPENDENT RUNS, WHERE THE BEST RESULT ON EACH TEST INSTANCE IS HIGHLIGHTED

Prob	Obj.	Dec.	K-RVEA	NSGAII_GP	FS-ND	MS-ND	FS-RV	MS-RV
DTLZ1a	M=2	D=10	7.303e+01 (2.092e+00) $-$	1.420e+02 (8.000e+00) $-$	1.400e+02 (1.347e+01) $-$	9.277e+01 (8.223e+00) $-$	1.098e+02 (8.882e+00) $-$	**3.409e+01 (2.294e+00)**
		D=30	7.195e+02 (2.720e+01) $-$	8.060e+02 (2.157e+01) $-$	7.005e+02 (2.185e+01) $-$	4.946e+02 (2.287e+01) $-$	4.737e+02 (1.927e+01) $-$	**2.531e+02 (8.460e+00)**
		D=50	1.260e+03 (8.061e+01) $-$	1.546e+03 (3.935e+01) $-$	1.358e+03 (4.051e+01) $-$	1.203e+03 (3.222e+01) $-$	8.543e+02 (3.193e+01) $-$	**5.306e+02 (1.085e+01)**
	M=3	D=10	4.739e+01 (2.188e+01) $\approx$	1.167e+02 (5.983e+00) $-$	1.358e+02 (6.113e+00) $-$	4.027e+01 (5.362e+00) $\approx$	8.014e+01 (5.087e+00) $-$	3.805e+01 (3.254e+00)
		D=30	5.956e+02 (5.304e+01) $-$	6.985e+02 (2.036e+01) $-$	7.244e+02 (1.999e+01) $-$	5.893e+02 (2.279e+01) $-$	2.708e+02 (1.308e+01) $\approx$	2.666e+02 (9.830e+00)
		D=50	1.172e+03 (1.362e+02) $-$	1.303e+02 (3.440e+00) $-$	1.313e+03 (3.327e+01) $-$	1.260e+03 (3.218e+01) $-$	4.910e+02 (2.134e+01) $\approx$	5.312e+02 (1.759e+01)
DTLZ2	M=2	D=10	1.714e−01 (6.159e−03) $\approx$	4.450e−01 (1.604e−02) $-$	4.140e−01 (2.171e−02) $-$	3.230e−01 (3.141e−02) $-$	2.130e−01 (1.016e−02) $-$	1.445e−01 (1.056e−02)
		D=30	1.464e+00 (3.612e−02) $-$	1.651e+00 (4.512e−02) $-$	1.256e+00 (3.630e−02) $-$	1.137e+00 (5.664e−02) $-$	6.854e−01 (2.905e−02) $-$	**5.291e−01 (1.749e−02)**
		D=50	2.749e+00 (5.577e−02) $-$	3.042e+00 (8.488e−02) $-$	2.697e+00 (8.008e−02) $-$	2.282e+00 (8.857e−02) $-$	1.283e+00 (4.391e−02) $-$	**8.448e−01 (2.548e−02)**
	M=3	D=10	2.480e−01 (7.944e−03) $\approx$	4.774e−01 (1.281e−02) $-$	4.525e−01 (1.501e−02) $-$	3.307e−01 (1.758e−02) $-$	2.815e−01 (9.995e−03) $\approx$	2.523e−01 (8.172e−03)
		D=30	1.477e+00 (3.908e−02) $-$	1.618e+00 (3.871e−02) $-$	1.628e+00 (4.168e−02) $-$	1.208e+00 (4.496e−02) $-$	7.524e−01 (1.953e−02) $-$	**6.289e−01 (1.658e−02)**
		D=50	2.836e+00 (5.895e−02) $-$	3.040e+00 (8.018e−02) $-$	2.943e+00 (7.285e−02) $-$	2.743e+00 (6.607e−02) $-$	1.262e+00 (3.058e−02) $\approx$	1.137e+00 (3.158e−02)
DTLZ3a	M=2	D=10	1.475e+02 (1.313e+01) $-$	3.184e+02 (1.095e+01) $-$	3.872e+02 (1.172e+01) $-$	1.002e+02 (5.522e+00) $-$	2.777e+02 (1.027e+01) $-$	**7.689e+01 (4.571e+00)**
		D=30	2.057e+03 (2.574e+02) $-$	2.118e+03 (5.393e+01) $-$	1.826e+03 (4.582e+01) $-$	1.363e+03 (5.816e+01) $-$	1.358e+03 (5.370e+01) $-$	**6.137e+02 (2.676e+01)**
		D=50	3.445e+03 (1.401e+03) $-$	4.087e+03 (9.775e+01) $-$	3.666e+03 (1.093e+02) $-$	3.105e+03 (1.045e+02) $-$	2.336e+03 (9.209e+01) $-$	**1.349e+03 (5.877e+01)**
	M=3	D=10	1.642e+02 (3.127e+01) $-$	2.691e+02 (1.317e+01) $-$	3.566e+02 (2.582e+01) $-$	6.032e+01 (4.096e+00) $\approx$	2.086e+02 (1.609e+01) $-$	4.661e+01 (2.799e+00)
		D=30	1.906e+03 (2.699e+02) $-$	2.196e+03 (5.114e+01) $-$	2.059e+03 (5.495e+01) $-$	1.443e+03 (7.121e+01) $-$	9.130e+02 (3.509e+01) $\approx$	7.515e+02 (4.384e+01)
		D=50	3.682e+03 (1.451e+02) $-$	4.098e+02 (9.425e+00) $-$	4.131e+03 (1.094e+02) $-$	3.460e+03 (9.808e+01) $-$	1.634e+03 (5.466e+01) $-$	**1.366e+03 (5.600e+01)**
DTLZ4a	M=2	D=10	2.667e−01 (1.116e−02) $\approx$	4.620e−01 (1.669e−02) $\approx$	8.821e−01 (2.190e−02) $-$	4.277e−01 (4.243e−02) $\approx$	6.844e−01 (3.515e−02) $-$	3.687e−01 (4.168e−02)
		D=30	1.590e+00 (4.005e−02) $-$	1.863e+00 (4.842e−02) $-$	1.512e+00 (4.102e−02) $-$	1.083e+00 (3.088e−02) $-$	1.062e+00 (3.242e−02) $-$	**8.124e−01 (2.682e−02)**
		D=50	2.887e+00 (9.379e−02) $-$	3.211e+00 (6.891e−02) $-$	2.746e+00 (7.183e−02) $-$	2.345e+00 (6.374e−02) $-$	1.372e+00 (4.468e−02) $\approx$	1.199e+00 (3.895e−02)
	M=3	D=10	2.083e+02 (2.794e+01) $-$	2.579e+02 (1.361e+01) $-$	3.645e+02 (2.578e+01) $-$	5.885e+01 (4.126e+00) $\approx$	2.128e+02 (1.631e+01) $-$	4.589e+01 (2.779e+00)
		D=30	1.568e+00 (4.348e−02) $-$	2.059e+00 (5.754e−02) $-$	1.702e+00 (4.995e−02) $-$	1.669e+00 (5.304e−02) $-$	1.205e+00 (5.063e−02) $\approx$	1.108e+00 (2.803e−02)
		D=50	2.789e+00 (6.937e−02) $-$	3.180e+00 (7.616e−02) $-$	2.924e+00 (7.584e−02) $-$	2.961e+00 (8.172e−02) $-$	1.531e+00 (6.575e−02) $\approx$	1.588e+00 (4.401e−02)
$-/+/\approx$			18/1/5	24/0/0	24/0/0	20/0/4	19/0/5	

in comparison with the quality of the ND. The reason might be that the decision variables of F1–F4 are linearly correlated while those of the DTLZ test instances are independent of each other.

3) Comparison With Offline Data-Driven Evolutionary Algorithms: Tables II and III present the statistical results of the IGD values obtained by the six compared algorithms on F1–F4 and the DTLZ test suite over 30 independent runs, where the best result of each test instance is highlighted. The Wilcoxon rank sum test is also adopted at a significance level of 0.05, where symbols "+," "−," and "≈" indicates that the result obtained by other algorithms is significantly better, significantly worse and no difference to that obtained by the proposed algorithm MS-RV, respectively. Note that solutions of each algorithm are evaluated using the real objective functions before calculating of the IGD value. From these two tables, we can draw the following conclusions.

First, the proposed algorithm MS-RV achieves the best overall performance against K-RVEA, NSGAII_GP, and its three variants. In particular, MS-RV significantly outperforms K-RVEA and NSGAII_GP on higher dimensional problems, i.e., $D = 30$ and $D = 50$ of both DTLZ and F test instances. These results demonstrate that MS-RV is competitive in solving high-dimensional problems in comparison with K-RVEA and NSGAII_GP.

Second, regarding the performance of the surrogates, the comparative results in terms of the IGD values between the two ND algorithms FS-ND (FS) and MS-ND (multisurrogates)

TABLE IV
STATISTICAL IGD RESULTS OF THE COMPARED ALGORITHMS WITH DIFFERENT FS MODELS ON EIGHT TEST INSTANCES OVER 30 INDEPENDENT RUNS, WHERE THE BEST RESULT ON EACH TEST INSTANCE IS HIGHLIGHTED

	SVR				GP			
Prob	FS-ND	MS-ND	FS-RV	MS-RV	FS-ND	MS-ND	FS-RV	MS-RV
F1	6.985e+00 (4.783e−01) −	4.181e+00 (5.174e−01) ≈	5.639e+00 (5.144e−01) −	3.402e+00 (4.502e−01)	1.023e+01 (1.398e+00) ≈	6.026e+00 (5.353e−01) ≈	8.980e+00 (1.316e+00) ≈	5.668e+00 (4.845e−01)
F2	6.020e+00 (5.848e−01) −	3.538e+00 (3.630e−01) ≈	6.125e+00 (5.750e−01) −	3.273e+00 (3.106e−01)	1.408e+01 (2.106e+00) −	6.337e+00 (8.119e−01) ≈	8.993e+00 (7.613e−01) −	5.345e+00 (4.912e−01)
F3	5.613e+00 (3.720e−01) −	4.637e+00 (4.226e−01) ≈	5.513e+00 (3.842e−01) −	3.311e+00 (2.655e−01)	1.007e+01 (7.468e−01) −	6.435e+00 (4.261e−01) ≈	9.286e+00 (4.625e−01) ≈	6.671e+00 (4.484e−01)
F4	5.169e+00 (4.190e−01) −	3.270e+00 (1.487e−01) −	4.782e+00 (3.767e−01) −	**2.683e+00 (1.826e−01)**	7.167e+00 (5.548e−01) −	2.703e+00 (2.531e−01) ≈	4.612e+00 (2.767e−01) −	2.020e+00 (1.408e−01)
DTLZ1a	1.135e+02 (4.207e+00) −	6.661e+01 (4.445e+00) −	1.050e+02 (4.749e+00) −	**4.085e+01 (3.380e+00)**	2.068e+02 (1.084e+01) −	1.474e+02 (1.401e+01) −	8.275e+01 (6.416e+00) −	**4.947e+01 (3.457e+00)**
DTLZ2	5.954e−01 (4.114e−02) −	5.473e−01 (3.522e−02) −	4.979e−01 (3.079e−02) −	**3.154e−01 (1.788e−02)**	5.117e−01 (2.297e−02) −	4.199e−01 (1.599e−02) −	2.557e−01 (1.305e−02) −	**1.769e−01 (5.355e−03)**
DTLZ3a	3.773e+02 (2.170e+01) −	2.552e+02 (1.573e+01) −	2.783e+02 (2.260e+01) −	**1.553e+02 (1.232e+01)**	4.832e+02 (2.170e+01) −	3.308e+02 (1.573e+01) −	1.933e+02 (2.260e+01) −	**1.145e+02 (1.232e+01)**
DTLZ4a	6.146e−01 (4.094e−02) −	5.351e−01 (3.471e−02) −	5.063e−01 (3.056e−02) −	**3.158e−01 (1.769e−02)**	5.680e−01 (4.331e−02) ≈	5.968e−01 (4.935e−02) ≈	4.057e−01 (2.380e−02) ≈	4.228e−01 (3.891e−02)

demonstrate that MS-ND performs better than FS-ND on 25 out of 36 test instances and has never outperformed by the single FS assisted algorithm.

4) Scalability of the Proposed Algorithm on Different Models: In this section, we compare FS-ND, MS-ND, FS-RV, and MS-RV on eight test problems when using SVR and GP, respectively, as the FS. For each test problem, the number of decision variables D is set to 10 due to the prohibitive computational cost of GP on high-dimensional problems.

The IGD values of the solution sets obtained by the algorithms under comparison averaged over 30 independent runs when using different FS models are presented in Table IV. We can find from these results that the four algorithms assisted by SVR and GP, respectively, perform similarly when they are assisted by the RBF model. Furthermore, MS-RV achieves the best overall performance among the four compared algorithms on the eight test problems. These findings indicate that the knowledge transfer strategy used in MS-RV can also work well when other models than the RBF are used as the surrogate.

5) Sensitivity of the Proposed Algorithm to Data Size: In this section, we evaluate the sensitivity of the performance of FS-ND, MS-ND, FS-RV, and MS-RV on the data size ($5D$, $10D$, $15D$, $20D$, and $25D$, D is the number of the decision variables) on four test problems with $M = 2$ and $D = 30$.

The mean and standard deviation of the IGD values of the solution sets obtained by each algorithm over 20 independent runs for the different data sizes are plotted in Fig. 5. On F1 and F3, the IGD values of all algorithms vary little when the size of the offline data changes. On DTLZ2 and DTLZ4a, the IGD values of MS-RV and MS-ND also improves as the number of data increases, although the performance of FS-RV and FS-ND generally remains unchanged. To better understand the above observations, we calculate the root mean square error (RMSE) of each fine model trained using different offline data sizes and the results are plotted in Fig. S.A in Appendix C of the supplementary material. The results

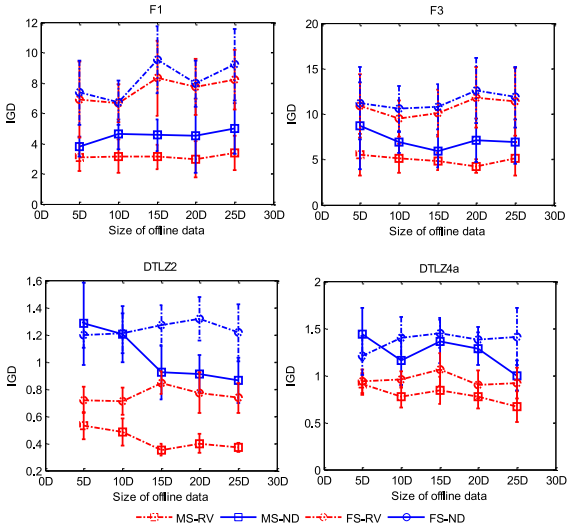

Fig. 5. Mean and standard deviation of IGD values obtained by FS-ND, MS-ND, FS-RV, and MS-RV using different sizes of historical data.

in the figure show that the approximation error of the fine models on F1, DTLZ1, and DTLZ4a gradually decreases as the size of the data increases, while the error on F3 remains nearly unchanged. These results indicate that the quality of fine models is generally becoming better as the number of data increases. However, only the coarse-fine search is able to make use of the improved accuracy of the surrogates on the DTLZ problems.

C. Comparison With Online Algorithms

In general, the offline data-driven optimization algorithm can be seen as one step in the online algorithms once the final solution set is generated and validated. In this section, the proposed MS-RV is modified to be an online algorithm. To this end, six generated solutions from each run of MS-RV are evaluated by the real fitness functions and are used for updating the surrogate model. To evaluate the

TABLE V
STATISTICAL IGD RESULTS OF THE ONLINE VERSION OF MS-RV AND THREE ONLINE ALGORITHMS ON EIGHT TEST INSTANCES OVER 30 INDEPENDENT RUNS, WHERE THE BEST RESULT ON EACH TEST INSTANCE IS HIGHLIGHTED

Prob.	Dec.	K-RVEA	ParEGO	MOEA/D-EGO	MS-RV
F1	D = 10	**3.816e−01(1.627e−02)** +	4.408e+00(3.639e−01) −	7.154e−01(3.465e−02) −	4.569e−01(1.258e−02)
	D = 30	2.260e+00(1.532e−01) ≈	9.069e+00(3.105e−01) −	5.060e+00(2.666e−01) −	8.494e−01(1.994e−02)
	D = 50	6.724e+00(5.180e−01) −	1.017e+01(1.913e−01) −	8.952e+00(3.423e−01) −	**1.465e+00(5.688e−02)**
F2	D = 10	5.965e−01(8.323e−03) −	6.611e+00(6.912e−01) −	7.744e−01(4.534e−02) −	**5.660e−01(1.127e−02)**
	D = 30	5.454e+00(1.033e+00) ≈	1.262e+01(2.371e−01) −	7.779e+00(3.904e−01) −	1.305e+00(4.859e−02)
	D = 50	1.129e+01(2.129e−01) −	1.213e+01(2.521e−01) −	1.114e+01(3.593e−01) −	**2.541e+00(5.934e−02)**
F3	D = 10	**1.076e−01(7.496e−03)** +	6.165e+00(5.154e−01) −	1.180e+00(1.629e−01) ≈	1.372e+00(1.655e−01)
	D = 30	3.913e+00(8.112e−01) ≈	1.196e+01(3.183e−01) −	6.538e+00(4.715e−01) −	2.868e+00(8.924e−02)
	D = 50	6.041e+00(5.802e−01) ≈	1.274e+01(3.638e−01) −	1.092e+01(5.382e−01) −	3.939e+00(1.025e−01)
F4	D = 10	1.032e+00(6.919e−02) −	6.214e+00(6.025e−01) −	1.717e+00(1.311e−01) −	**5.473e−01(2.923e−02)**
	D = 30	3.932e+01(1.374e+00) −	5.850e+01(1.029e+00) −	3.569e+01(2.555e+00) −	5.333e+00(1.415e−01)
	D = 50	5.054e+01(5.098e+00) −	6.651e+01(1.711e+00) −	5.960e+01(1.008e+00) −	**1.064e+01(4.319e−01)**
DTLZ1a	D = 10	7.829e+00(3.916e−01) −	2.113e+01(5.408e+00) −	8.656e+00(6.330e−01) −	**3.244e−01(9.714e−03)**
	D = 30	3.940e+01(1.371e+00) −	5.827e+01(1.005e+00) −	3.638e+01(2.527e+00) −	**5.311e+00(1.389e−01)**
	D = 50	1.153e+03(1.674e+01) −	3.309e+02(7.754e+01) −	8.397e+01(2.524e+00) −	**1.582e+01(4.361e−01)**
DTLZ2	D = 10	1.408e−01(4.109e−03) −	3.743e−01(3.548e−03) −	3.724e−01(2.612e−03) −	**7.895e−02(4.067e−04)**
	D = 30	1.016e+00(1.701e−02) −	1.453e+00(1.430e−02) −	1.106e+00(2.215e−02) −	**2.618e−01(1.646e−03)**
	D = 50	2.202e+00(3.380e−02) −	2.742e+00(2.586e−02) −	2.255e+00(5.857e−02) −	**5.171e−01(6.838e−03)**
DTLZ3a	D = 10	3.394e+00(1.898e−01) −	2.912e+01(1.370e+00) −	2.227e+01(1.496e+00) −	**8.332e−01(2.614e−02)**
	D = 30	9.588e+01(2.628e+00) −	1.797e+02(1.577e+00) −	1.064e+02(4.872e+00) −	**1.952e+01(6.503e−01)**
	D = 50	3.338e+03(3.091e+01) −	3.711e+03(2.313e+01) −	2.980e+02(8.317e+00) −	**6.174e+01(1.757e+00)**
DTLZ4a	D = 10	2.509e−01(5.972e−03) −	5.064e−01(9.452e−03) −	4.679e−01(5.400e−03) −	**1.958e−01(2.960e−03)**
	D = 30	1.179e+00(2.204e−02) −	1.605e+00(1.732e−02) −	1.269e+00(2.412e−02) −	**4.563e−01(4.297e−03)**
	D = 50	3.073e+00(3.243e−02) −	3.063e+00(2.451e−02) −	2.313e+00(3.649e−02) −	**7.726e−01(9.772e−03)**

performance of the online version of MS-RV, we compare it with three online multiobjective data-driven optimization algorithms, K-RVEA [61], MOEA/D-EGO [23], and ParEGO [22] on the eight test problems ($M = 3$ in DTLZ problems) adopted in this paper. In the experiments, the initial training data size is set to $11D - 1$ and the maximum number of evaluations for each compared algorithm is set to 150. The three online algorithms are implemented in PlatEMO [67].

The IGD results of the four compared algorithms are presented in Table V. We can see from the table that the online version of MS-RV significantly outperforms the three online data-driven optimization algorithms, especially on high-dimensional problems. The results imply that MS-RV is potentially also promising for solving high-dimensional online data-driven optimization problems.

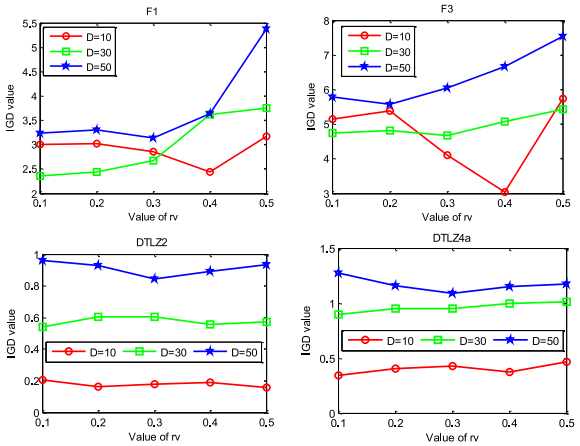

Fig. 6. Averaged IGD values obtained by MS-RV using different coefficients *rv*.

D. Parameter Sensitivity Analysis

In the proposed MS-RV algorithm, there are three parameters that may influence the performance of the algorithm, i.e., the coefficient *rv* that controls the dimension of coarse model, the maximum number of generations *TC* of the search assisted by the CS in each generation, and the number of solutions ρ for averaging in each cluster. In this section, we will analyze the sensitivity of the performance of MS-RV to the three parameters on four test functions, namely F1 ($D = 30$), F3 ($D = 30$), DTLZ2 ($M = 2$ and $D = 30$), and DTLZ4a ($M = 2$ and $D = 30$).

1) Coefficient (rv): The *rv* determines the dimension of CSs, which influences search assisted by the CSs as well as the knowledge to be transferred to the FSs, which eventually influences the performance of MS-RV. Fig. 6 presents the obtained results within 40 generations on the above four instances with 10, 30, and 50 dimensions, in which the average IGD values with different *rv* are plotted. In this figure, we can find that F1 and F3 are more sensitive to *rv* than DTLZ2 and DTLZ4a, probably because the decision variables of F1 and F3 are correlated. It can also be seen that MS-RV achieves the best overall IGD values when *rv* is around 0.3. Therefore, we use $rv = 0.3$ in the experiments for comparisons.

2) Maximum Number of Generations of the Coarse Search (TC): As discussed in Section III, the proposed multisurrogate method reuses knowledge of acquired by the CSs to enhance the convergence of the fine search. *TC* can directly influences the knowledge acquired from the coarse search, thereby influencing the performance of MS-RV. Fig. 7 plots the average IGD values with different *TC* on the above four test problems. As can be seen in the figure, MS-RV is slightly more sensitive to *TC* on F1 and F3 than on the two DTLZ test problems.

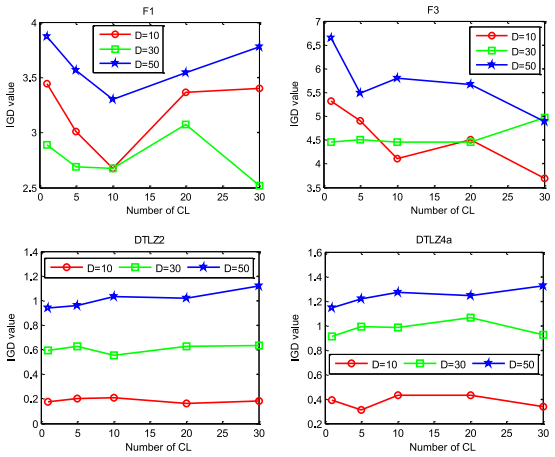

Fig. 7. Averaged IGD values obtained by MS-RV using different number of generations *TC* in the coarse search.

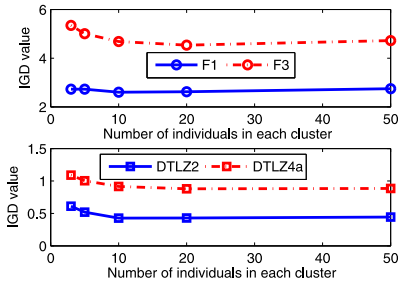

Fig. 8. Averaged IGD values obtained by MS-RV with different numbers of solutions in each cluster ρ.

Nevertheless, the performance is satisfactory in general when *TC* is set between 10 and 20. Accordingly, we use $TC = 15$.

3) Number of Solutions in Each Cluster (ρ): The purpose of averaging over ρ solutions in each cluster is to reduce the influence of the errors introduced by the surrogates. If ρ is too small, averaging will not work properly. However, a too large ρ may lead to poor performance as many very different will participate in the averaging. For this reason, we investigate a proper ρ for generating high quality solutions on 30-D F1, F3, DTLZ2, and DTLZ4 instances. The comparative results of $\rho = 3, 5, 10, 20,$ and 50 are shown in Fig. 8. From these results, we can see that the algorithm has shown the best performance when ρ is around 20. Therefore, we set $\rho = 20$.

V. Case Study

In this section, we consider the application of the proposed MS-RV algorithm to a real-world operational indices optimization of the beneficiation process. This is a typical offline data-driven problem in that mathematical equations of the objective functions cannot be obtained due to the complex physical and chemical reactions in the process and only a small amount of historical data can be used. A brief introduce of the operational indices optimization problem is presented in Appendix D of the supplementary material.

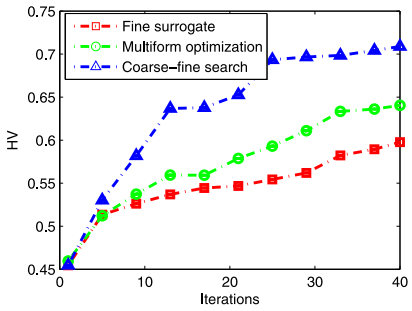

Fig. 9. HV of solution sets obtained by the proposed algorithm over the generations.

A. Problem Description

The purpose of operational indices optimization in beneficiation processes aims to improve the concentrate grade (G), concentrate yield (Y), as well as to decrease the energy consumption (E) including the costs of roasting unit, grinding units by properly coordinating the operating state of each unit. In this paper, we consider the following 15 operate indices, namely, particle sizes of the raw ore entered in LMPL and HMPL (pl, ph) and grade of the raw ore entered in LMPL and HMPL (gl, gh), capacity and run time of the shaft furnace roasting (sc, st), grade of waste ore (gw), grade of feed ore of grindings in LMPL and HMPL (gfl, gfh), capacity of grindings in LMPL and HMPL (gcl, gch) and running time of grindings in LMPL and HMPL (gtl, gth), grade of tailings from LMPL and HMPL (tl, th). These 15 operational indices are taken as the decision variables and denoted as $X = (pl, ph, gl, gh, sc, st, gw, gfl, gfh, gcl, gch, gtl, gth, tl, th)$. The optimization problem can be formulated as follows:

$$\begin{aligned} \min \quad & -G, -Y, E \\ \text{s.t.} \quad & G = \Phi_1(X), Y = \Phi_2(X) \\ & E = sc + 0.3st + gcl + gch + gtl + gth \end{aligned} \tag{3}$$

where Φ_1 and Φ_2 represent the unknown correlation between the objectives and the decision variables in the operational indices optimization problem.

B. Optimization Results

In this real-world application, we are not able to validate the obtained solutions as no "real objective functions" are available for validation. For this reason, we evaluate the search ability of coarse-fine search strategy in the proposed MF-RV by comparing with the FS and the multiform optimization approach [68], which is the most recent proposed algorithm in solving the operational indices optimization problem. Note that the target accurate model in multiform approach is replaced with RBF for fair comparison. Each of the compared algorithm is employed to solve the operational indices optimization problem, which contains 150 pairs of collected historical data. We first conduct 30 independent runs of each algorithm, each run performing 40 generations of fine search.

The mean and standard deviation of the HV values of the solution sets obtained by three compared algorithms over the

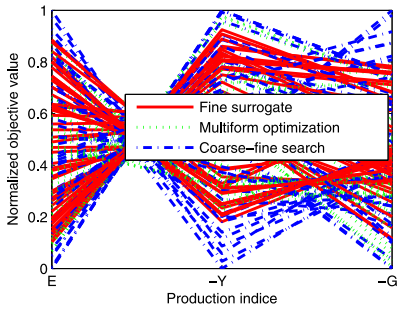

Fig. 10. Parallel coordinate plot of a set of 50 nondominated solutions obtained of each algorithm.

generations are plotted in Fig. 9. We can see from the figure that the HV values gradually increase over the generations, indicating that all compared algorithms converge on the operational indices optimization problem in general as discussed in Section IV-B. We can also see from Fig. 9 that the coarse-fine search strategy maintains the best HV values among the compared algorithms, demonstrating its best search ability compared with the other two algorithms. To further examine the quality of the solutions found by each algorithm, a set of 50 nondominated solutions are plotted using parallel coordinates in Fig. 10. The figure shows that the solutions of each production index achieved by the coarse-fine strategy are distributed in a larger region. This result confirms that the coarse-fine search strategy is able to achieve a more diverse solution set.

VI. Conclusion

In this paper, an offline data-driven optimization algorithm, called MS-RV, has been proposed for solving offline multiobjective data-driven optimization problem. The developed algorithm builds a CS and an FS using the historical data set. The CS, which is dynamically constructed in the subspace of the original search space, is used for quick exploration of a promising region, whereas the FS aims to help the optimizer exploit the promising solutions. Meanwhile, a knowledge transfer method is employed to transfer the knowledge from the CS to the FS to enhance the convergence of the optimization process assisted by the FS. After that, an RV-based final solution set generation strategy is proposed to reduce the influence of the approximation errors introduced by the surrogates, thereby generating high-quality solutions. We compare the MS-RV algorithm with two state-of-the-art algorithms K-RVEA and NSGAII_GP and three variants of MS-RV on eight MOPs of 10, 30, and 50 dimensions. Empirical results demonstrate that MS-RV achieves the best overall performance than the compared algorithms. Finally, MS-RV is applied to solve a real-world operational indices optimization problem, which is a typical offline data-driven optimization problem.

Despite the encouraging performance of the proposed algorithm on the test problems, we find from the experiments that the algorithm is not able to well approximate the true PF for hard optimization problems, such as F3 and DTLZ4a. One reason may be that the offline data of these problems are ill-distributed in the objective space. In the future, we are interested in improving the performance of the proposed algorithm for addressing ill-distributed historical data by developing new surrogates, surrogate-management strategies, and final solution generation methods.

References

[1] D. J. Allstot, K. Choi, and J. Park, *Parasitic-Aware Optimization of CMOS RF Circuits*. New York, NY, USA: Springer, 2003.

[2] T. A. Milligan, *Modern Antenna Design*. Hoboken, NJ, USA: Wiley, 2005.

[3] R. Cheng, T. Rodemann, M. Fischer, M. Olhofer, and Y. Jin, "Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 1, no. 2, pp. 97–111, Apr. 2017.

[4] 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.

[5] H. Wang, J. Doherty, and Y. Jin, "Hierarchical surrogate-assisted evolutionary multi-scenario airfoil shape optimization," in *Proc. Congr. Evol. Comput.*, 2018, pp. 8–13.

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

[7] D. Guo, T. Chai, J. Ding, and Y. Jin, "Small data driven evolutionary multi-objective optimization of fused magnesium furnaces," in *Proc. IEEE Symp. Series Comput. Intell. (SSCI)*, 2016, pp. 1–8.

[8] J. Ding, T. Chai, H. Wang, and X. Chen, "Knowledge-based global operation of mineral processing under uncertainty," *IEEE Trans. Ind. Informat.*, vol. 8, no. 4, pp. 849–859, Nov. 2012.

[9] 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. doi: 10.1109/TEVC.2018.2834881.

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

[11] Y. Jin, "Data driven evolutionary optimization of complex systems: Big data versus small data," in *Proc. ACM Genet. Evol. Comput. Conf. Companion*, 2016, pp. 1281–1282.

[12] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments—A survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.

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

[14] 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.

[15] A. Gaspar-Cunha and A. Vieira, "A hybrid multi-objective evolutionary algorithm using an inverse neural network," in *Hybrid Metaheuristics*. Valencia, Spain: Springer, 2004, pp. 25–30.

[16] Y. Lian and M.-S. Liou, "Multiobjective optimization using coupled response surface model and evolutionary algorithm," *AIAA J.*, vol. 43, no. 6, pp. 1316–1325, 2005.

[17] M. Herrera, A. Guglielmetti, M. Xiao, and R. F. Coelho, "Metamodel-assisted optimization based on multiple kernel regression for mixed variables," *Struct. Multidiscipl. Optim.*, vol. 49, no. 6, pp. 979–991, 2014.

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

[19] S. Z. Martínez and C. A. Coello Coello, "MOEA/D assisted by RBF networks for expensive multi-objective optimization problems," in *Proc. ACM 15th Annu. Conf. Genet. Evol. Comput.*, 2013, pp. 1405–1412.

[20] A. Isaacs, T. Ray, and W. Smith, "An evolutionary algorithm with spatially distributed surrogates for multiobjective optimization," in *Proc. Aust. Conf. Artif. Life*, 2007, pp. 257–268.

[21] D. Buche, N. N. Schraudolph, and P. Koumoutsakos, "Accelerating evolutionary algorithms with Gaussian process fitness function models," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 2, pp. 183–194, May 2005.

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

[23] 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.

[24] D. Lim, Y.-S. Ong, Y. Jin, and B. Sendhoff, "A study on metamodeling techniques, ensembles, and multi-surrogates in evolutionary computation," in *Proc. ACM 9th Annu. Conf. Genet. Evol. Comput.*, 2007, pp. 1288–1295.

[25] Y. Jin and B. Sendhoff, "Reducing fitness evaluations using clustering techniques and neural network ensembles," in *Proc. Genet. Evol. Comput. Conf.*, 2004, pp. 688–699.

[26] A. Rosales-Pérez, C. A. Coello Coello, J. A. Gonzalez, C. A. Reyes-Garcia, and H. J. Escalante, "A hybrid surrogate-based approach for evolutionary multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2013, pp. 2548–2555.

[27] T. Goel, R. T. Haftka, W. Shyy, and N. V. Queipo, "Ensemble of surrogates," *Struct. Multidiscipl. Optim.*, vol. 33, no. 3, pp. 199–216, 2007.

[28] 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.

[29] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble based infill criterion for evolutionary multi-objective optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1012–1025, Mar. 2019. doi: 10.1109/TCYB.2018.2794503.

[30] Z. Zhou, Y. S. Ong, M. H. Nguyen, and D. Lim, "A study on polynomial regression and Gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm," in *Proc. IEEE Congr. Evol. Comput.*, vol. 3, 2005, pp. 2832–2839.

[31] 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.

[32] C. Sun, J. Zeng, J. Pan, S. Xue, and Y. Jin, "A new fitness estimation strategy for particle swarm optimization," *Inf. Sci.*, vol. 221, pp. 355–370, Feb. 2013.

[33] 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, 2015.

[34] 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.

[35] 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. doi: 10.1109/TEVC.2018.2869001.

[36] 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.

[37] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.

[38] 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.

[39] I. Koprić, *Citizens as Partners: Information, Consultation and Public Participation in Policy-Making*. Org. Econ. Co-Oper. Develop., Paris, France, 2001.

[40] Y. Jin, Ed., *Knowledge Incorporation in Evolutionary Computation*. Berlin, Germany: Springer, 2006.

[41] A. Gupta, Y.-S. Ong, and L. Feng, "Insights on transfer optimization: Because experience is the best teacher," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 51–64, Feb. 2018.

[42] S. J. Louis and J. McDonnell, "Learning with case-injected genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 4, pp. 316–328, Aug. 2004.

[43] P. Cunningham and B. Smyth, "Case-based reasoning in scheduling: Reusing solution components," *Int. J. Prod. Res.*, vol. 35, no. 11, pp. 2947–2962, 1997.

[44] A. Moshaiov and A. Tal, "Family bootstrapping: A genetic transfer learning approach for onsetting the evolution for a set of related robotic tasks," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2014, pp. 2801–2808.

[45] L. Feng, Y.-S. Ong, A.-H. Tan, and I. W. Tsang, "Memes as building blocks: A case study on evolutionary optimization + transfer learning for routing problems," *Memetic Comput.*, vol. 7, no. 3, pp. 159–180, 2015.

[46] L. Feng, Y.-S. Ong, M.-H. Lim, and I. W. Tsang, "Memetic search with interdomain learning: A realization between CVRP and CARP," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 644–658, Oct. 2015.

[47] L. Feng, Y.-S. Ong, I. W.-H. Tsang, and A.-H. Tan, "An evolutionary search paradigm that learns with past experiences," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2012, pp. 1–8.

[48] L. Feng, Y.-S. Ong, S. Jiang, and A. Gupta, "Autoencoding evolutionary search with learning across heterogeneous problems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 760–772, Oct. 2017.

[49] M. Pelikan and M. W. Hauschild, "Learn from the past: Improving model-directed optimization by transfer learning based on distance-based bias," Missouri Estim. Distrib. Algorithms Lab., Univ. Missouri at St. Louis, St. Louis, MO, USA, Rep. 2012007, 2012.

[50] R. Santana, A. Mendiburu, and J. A. Lozano, "Structural transfer using EDAS: An application to multi-marker tagging SNP selection," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2012, pp. 1–8.

[51] E. Haslam, B. Xue, and M. Zhang, "Further investigation on genetic programming with transfer learning for symbolic regression," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2016, pp. 3598–3605.

[52] M. Iqbal, W. N. Browne, and M. Zhang, "Reusing building blocks of extracted knowledge to solve complex, large-scale Boolean problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 465–480, Aug. 2014.

[53] D. O'Neill, H. Al-Sahaf, B. Xue, and M. Zhang, "Common subtrees in related problems: A novel transfer learning approach for genetic programming," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2017, pp. 1287–1294.

[54] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: Toward evolutionary multitasking," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 343–357, Jun. 2016.

[55] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, "Multi-objective multifactorial optimization in evolutionary multitasking," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1652–1665, Jul. 2017.

[56] J. Ding, C. Yang, Y. Jin, and T. Chai, "Generalized multitasking for evolutionary optimization of expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 44–58, Feb. 2019. doi: 10.1109/TEVC.2017.2785351.

[57] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[58] 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.

[59] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, vol. 1, 2002, pp. 825–830.

[60] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, Feb. 2008.

[61] 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.

[62] G. Jekabsons, *RBF: Radial Basis Function Interpolation for MATLAB/OCTAVE*, vol. 1, Riga Tech. Univ., Riga, Latvia, 2009.

[63] S. Lophaven, H. Nielsen, and J. Søndergaard, *DACE—A MATLAB Kriging Toolbox, Version 2.0*, Tech. Univ. Denmark, Lyngby, Denmark, 2002.

[64] R. L. Iman, "Latin hypercube sampling," in *Encyclopedia of Quantitative Risk Analysis and Assessment*. Chichester, U.K.: Wiley, 2008.

[65] 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.

[66] L. M. S. Russo and A. P. Francisco, "Quick hypervolume," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 481–502, Aug. 2014.

[67] 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.

[68] C. Yang, J. Ding, Y. Jin, C. Wang, and T. Chai, "Multitasking multiobjective evolutionary operational indices optimization of beneficiation processes," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1046–1057, Jul. 2019. doi: 10.1109/TASE.2018.2865593.

[69] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Comput.*, vol. 3, no. 2, pp. 246–257, Jun. 1991.

Cuie Yang received the B.Sc. degree from Henan Polytechnic University, Jiaozuo, China, in 2014, and the M.Sc. degree from Northeastern University, Shenyang, China, in 2016, where she is currently pursuing the Ph.D. degree in control theory and control engineering with the State Key Laboratory of Synthetical Automation for Process Industry.

Her current research interests include multitasking evolutionary optimization, data-driven evolutionary optimization, and their application.

Jinliang Ding (M'09–SM'14) received the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2012.

He is a Professor with the State Key Laboratory of Synthetical Automation for Process Industry, Northeastern University. He has authored or coauthored over 100 refereed journal papers and refereed papers at international conferences. He is also the inventor or co-inventor of 17 patents. His current research interests include modeling, plant-wide control and optimization for the complex industrial systems, stochastic distribution control, and multiobjective evolutionary algorithms and its application.

Dr. Ding was a recipient of the Young Scholars Science and Technology Award of China in 2016, the National Science Fund for Distinguished Young Scholars in 2015, the National Technological Invention Award in 2013, two First-Prize of Science and Technology Award of the Ministry of Education in 2006 and 2012, and the IFAC Control Engineering Practice 2011–2013 Paper Prize.

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

He is currently a Distinguished Chair Professor of Computational Intelligence with the Department of Computer Science, and the Head of the Nature Inspired Computing and Engineering Group, University of Surrey, Guildford, U.K., where he heads the Nature Inspired Computing and Engineering Group. He was a Finland Distinguished Professor and a Changjiang Distinguished Visiting Professor. He has (co)authored over 300 peer-reviewed journal and conference papers and holds eight patents on evolutionary optimization.

Dr. Jin was a recipient of the 2014 and 2016 IEEE Computational Intelligence Magazine Outstanding Paper Award, the 2018 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award, and the Best Paper Award of the 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and *Complex and Intelligent Systems*. He is also an Associate Editor or an Editorial Board Member of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON NANOBIOSCIENCE, *Evolutionary Computation*, *BioSystems*, *Soft Computing*, and *Natural Computing*. He is an IEEE Distinguished Lecturer for the period from 2017 to 2019.

Tianyou Chai (M'90–SM'97–F'08) received the Ph.D. degree in control theory and engineering from Northeastern University, Shenyang, China, in 1985.

He has been with the Research Center of Automation, Northeastern University, since 1985, where he became a Professor in 1988 and a Chair Professor in 2004. He is also the Founder and the Director of the Center of Automation, which became a National Engineering and Technology Research Center in 1997. He has made a number of important contributions in control technologies and applications. He has authored and coauthored two monographs, 84 peer reviewed international journal papers, and around 219 international conference papers. He has been invited to deliver over 20 plenary speeches in international conferences of IFAC and the IEEE. His current research interests include adaptive control, intelligent decoupling control, integrated plant control and systems, and the development of control technologies with applications to various industrial processes.

Prof. Chai was a recipient of three prestigious awards of National Science and Technology Progress, the 2002 Technological Science Progress Award from the Ho Leung Ho Lee Foundation, the 2007 Industry Award for Excellence in Transitional Control Research from the IEEE Control Systems Society, and the 2010 Yang Jia-Chi Science and Technology Award from the Chinese Association of Automation. He is a member of the Chinese Academy of Engineering, an Academician of the International Eurasian Academy of Sciences, and an IFAC Fellow. He is a Distinguished Visiting Fellow of the Royal Academy of Engineering (U.K.), and an Invitation Fellow of the Japan Society for the Promotion of Science.