A Kriging-Assisted Evolutionary Algorithm Using Feature Selection for Expensive Sparse Multi-Objective Optimization

Zheng Tan
Xidian University
School of Artificial Intelligence
Xi'an, China
zhengtan@stu.xidian.edu.cn

Handing Wang
Xidian University
School of Artificial Intelligence
Xi'an, China
hdwang@xidian.edu.cn

***Abstract*—The Pareto sets of many real-world multi-objective optimization problems in engineering and computer fields are high-dimensional but sparse. Such multi-objective optimization problems are called large-scale sparse multi-objective optimization problems. A sparse evolutionary algorithm has also been raised and verified effective on benchmark problems. However, in practical applications, it needs a large number of expensive function evaluations. Although surrogate-assisted evolutionary algorithms are common solutions to deal with expensive optimization problems within limited computation resources and especially Kriging-assisted evolutionary algorithms are widely used, they cannot cope with large-scale expensive sparse multi-objective optimization problems due to the inaccurate surrogate models on high-dimensional problems. Therefore, we first propose a feature selection operator based on non-dominated sorting to choose the non-zero decision variables in the Pareto set. Then, the dimension of the original problem is reduced and a Kriging-assisted multi-objective evolutionary algorithm is employed to solve the reformulated problem. Finally, the selected zero decision variables are added to the obtained optimal solutions as the result of the original problem. The experimental results on benchmark problems show that our proposed algorithm outperforms the existing algorithms.**

***Index Terms*—Sparse multi-objective optimization problems, surrogate-assisted multi-objective evolutionary algorithms, feature selection, Kriging model.**

I. Introduction

Since the objectives of multi-objective optimization problems (MOPs) are conflicted [1], it is impossible to find a single optimal solution for all the objectives, instead a set of Pareto optimal solutions is obtained which makes a trade-off between different objectives. These solutions constitute the Pareto front and corresponding decision set is called Pareto set. In many real-world problems, their Pareto sets may be sparse, which means many of the decision variables in Pareto set are zero. Such problems are known as sparse MOPs [2]. Sparse Pareto sets exist widely in real-world MOPs. Taking the weight matrix for a neural network [3] as an example, many neurons are not connected and in the weight matrix their corresponding elements are zero.

This work was supported in part by the National Natural Science Foundation of China (No. 61976165).

Evolutionary algorithms (EAs) have been verified effective on MOPs [4] [5], because they naturally provide a set of candidate solutions. Many existing multi-objective evolutionary algorithms (MOEAs), like MOEA/D [6] and NSGA-III [7], have a good performance on benchmark problems. However, the execution of MOEAs on real-world MOPs needs a large number of function evaluations, which are often expensive. For instance, a CFD simulation [8] requires at least several minutes. One common possible solution is to adopt cheap surrogate models to approximate the expensive function evaluations in EAs. Surrogate-assisted evolutionary algorithms (SAEAs) perform well with limited number of expensive function evaluations [9], [10]. For example, MOEA/D-EGO [11] and K-RVEA [12] are two effective surrogate-assisted MOEAs, where the Kriging models with the uncertainty information are popular choices for SAEAs. In addition, surrogate models are adaptively updated by selecting points to be re-evaluated during the optimization process of SAEAs, which is known as the model management strategy [13]. The choices of surrogate model and model management are two key factors to effectiveness of SAEAs.

A main challenge to Kriging-assisted EAs is that their models dramatically consume training time as the problem dimension increases. For large-scale sparse expensive MOPs, high-dimensional decision variables degenerate the efficiency of Kriging-assisted EAs. Therefore, dimension reduction is necessary before using Kriging-assisted EAs on large-scale problems. In this work, we adopt K-RVEA [12] as the main optimizer for expensive sparse MOPs. To overcome the dimension curse, a feature selection operator is proposed to choose the non-zero decision variables in Pareto sets of sparse MOPs. Therefore, We term our proposed algorithm K-RVEA(FS).

The reminder of the paper is organized as follows. Section II briefly introduces existing dimension reduction techniques and K-RVEA. The proposed algorithm K-RVEA(FS) is described in Section III. We compare the proposed algorithm with three existing MOEAs in Section IV. Finally, the conclusions and future work are presented in Section V.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

TABLE I
FEATURE SELECTION TECHNIQUES

	Advantage	Disadvantage	Example
Filter methods	Generality less computation complexity without training model	Less acurate than wrapper	Relief Variance thresholder
Wrapper methods	Better accuracy simple to use interacts with classifier	More computation source risky for over-fitting	MSE-based method
Embedding methods	Mixed with model selection less complexity than wrapper	Classifier dependent selection	L1-normalize Decision tree

II. PRELIMINARIES

A. Dimension Reduction Algorithms

With the increasing dimensions, the construction of the surrogate models becomes complex and time-consuming. In other words, the source of time and hardware needed may be unaffordable and the methods are suffering from the curse of dimensionality [14]. Principal component analysis (PCA) [15] and latent dirichlet allocation (LDA) [16] are two widely used dimension reduction methods based on variable analysis. However, both techniques are hard to be used for addressing the high dimensions of sparse MOPs due to the lost of original dimensions. Feature selection, as a kind of dimension reduction algorithms, can be adopted to select a subset of decision variables from the whole high-dimensional decision space. Existing methods of feature selection can be divided into three categories roughly: filter, wrapper and embedding methods [17].

- Filter methods select a subset of features without considering subsequent algorithms. There are some famous filter methods including Relief [18] and FOCUS [19], which process the whole data and assess the correlations between features and labels.
- Wrapper methods [20] wrap the selection process via a learning algorithm. Wrapper methods constantly add or delete the features in the selected subset and assess the performance of the subset based on a specific algorithm. Then, wrapper methods select an outperforming subset based on a performance metric. Due to the constant updating of the subset, wrapper methods consume more computation resources than filter and embedding methods but they usually have a good accuracy.
- Embedding methods [21] aim to find a trade-off between computation budget and accuracy. Embedding methods select features by the model selection, which rank features by the weights. In the construction of the model, the relevant features are selected at the same time.

The main advantages and disadvantages of mentioned methods have been represented in Table I. In addition to those feature selection methods, a number of methods are combining some knowledge in their specific areas, like calculating the similarity between different features [22] and clustering the features to reduce the size of set based on a specific metric [23].

However, most mentioned feature selection methods need a large dataset and times of training on models. For expensive MOPs, the above two conditions mean a large number of expensive function evaluations and massive computation sources in model training, which is often unacceptable for real-world problems. During updating the surrogate model in SAEAs, only a small number of data can be added, which cannot significantly assist the dimension reduction in the decision space. Therefore, to deal with expensive large-scale MOPs, effective feature selection methods are needed.

B. Kriging Model

The Kriging model, also known as Gaussian process model, is the surrogate model used in K-RVEA [12]. The reason why the Kriging model is popular is that it could both provides predicted values and their uncertainty information. For an input x , the Kriging model predicts its output value with two parts:

$$y(x) = \mu(x) + z(x), \tag{1}$$

where $\mu(x)$ is a polynomial approximation model and $z(x)$ is assumed as a Gaussian process whose mean value is zero and variance is σ^2. To build the model, we need to sample some points. For two inputs x^i and x^j, the covariance between two random processes $\epsilon\left(x^i\right)$ and $\epsilon\left(x^j\right)$ are defined by

$$\operatorname{cov}\left[\epsilon\left(x^i\right), \epsilon\left(x^j\right)\right] = \sigma^2 \mathbf{R}\left(\left[R\left(x^i, x^j\right)\right]\right). \tag{2}$$

Thus, the correlation matrix of N_I data points can be written as follows:

$$\mathbf{R} = \begin{bmatrix} R\left(x^1, x^2\right) & \cdots & R\left(x^1, x^{N_I}\right) \\ \vdots & \ddots & \vdots \\ R\left(x^{N_I}, x^1\right) & \cdots & R\left(x^{N_I}, x^{N_I}\right). \end{bmatrix} \tag{3}$$

The correlation function adopted in K-RVEA is

$$R\left(x^i, x^j\right) = \exp\left(-\sum_{k=1}^{n} \theta_k \left|x_k^i - x_k^j\right|^2\right), \tag{4}$$

where θ_k is a correlation parameter. Once we construct our model using the sample points, the prediction for a new input $\bar{x}$ from the Kriging model can be written as:

$$\bar{y} = \beta + r^T(\bar{x})\mathbf{R}^{-1}(y - F\beta), \tag{5}$$

where F is a column vector of N_I components and $r^T(\bar{x})$ is the correlation vector between $\bar{x}$ and sample points. To fit the

model, θ_k is obtained by maximizing the following likelihood function:

$$\arg\max_{\theta_k} \left(-\left(n_s \ln\left(\hat{\sigma}^2\right) + \ln|\boldsymbol{R}|\right)/2\right). \tag{6}$$

The uncertainty information has made the Kriging model outperform other surrogate models in some aspects. However, the computational cost of Kriging model is high due to its high computation complexity [12]. For a large-scale optimization problem, this weakness may lead to poor performance.

C. Kriging-Assisted Reference Vector Guided Evolutionary Algorithm

Reference vector guided evolutionary algorithm (RVEA) [24] adopts a number of reference vectors to divide the objective space to several small parts and these reference vectors guide the evolutionary search in the population. With guidance of reference vectors, the population evolves to the Pareto front. K-RVEA [12] is an RVEA variant using the Kriging model to replace a number of expensive function evaluations. As shown in Fig. 1, K-RVEA borrows the evolutionary search of RVEA and samples new data points to update the Kriging model during the optimization process.

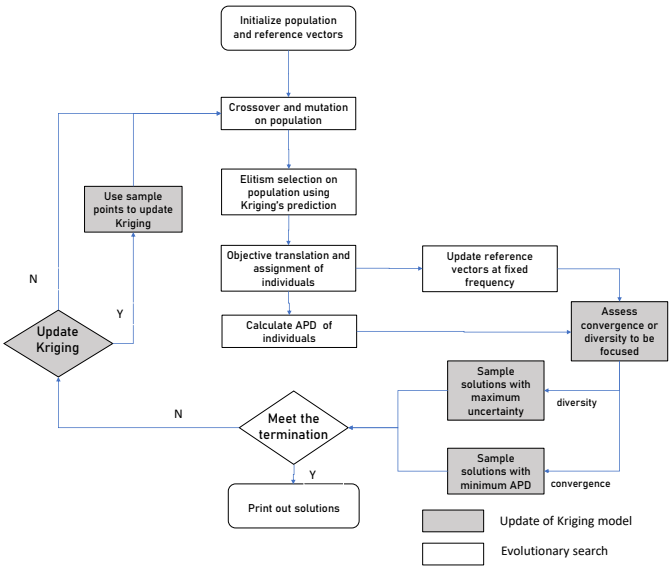

Fig. 1. Flowchart of K-RVEA

1) Generation of Reference Vectors: At the beginning of K-RVEA, uniformly distributed reference vectors are generated based on reference points, which are sampled on a hyperplane using the simplex-lattice design method [25].

$$\begin{cases} p_i = \left(p_i^1, p_i^2, \ldots, p_i^M\right) \\ s.t. \sum_{j=1}^{M} p_i^j = 1, \end{cases} \tag{7}$$

where $i = 1, 2, ..., N$, N is the number of uniformly distributed points and M is the number of objectives. p_i^j can be $\left\{\frac{0}{H}, \frac{1}{H}, \ldots \frac{H}{H}\right\}$ and H is an integer we adopted in the method. By calculating the permutation above, we can get some uniformly distributed vectors. Then, we change the original reference factors into

$$v_i = \frac{p_i}{\|p_i\|}. \tag{8}$$

Thus, reference vectors will be mapped from a hyperplane to a hypersphere and all those reference vectors are unit vectors. The objective space is divided into some subspace by those unit reference vectors.

2) Evolutionary Search: Before reference vectors are applied into process of evolution, every objective value is changed into

$$\hat{f_i^j} = f_i^j - f_i^*, \tag{9}$$

where f_i^j is the value of j-th individual on i-th objective and f_i^* is the minimum objective value of the i-th objective. After the translation, all the objective values are positive. The cosine value of the angle between individual and every reference vector is calculated. We denote the minimum angle and the corresponding cosine value between individual i and reference vectors by θ_i and a_i. By the assignment, the whole population is also divided into subpopulations. In each subpopulation, only one individual is selected based on both convergence and diversity. In both K-RVEA and RVEA, convergence is measured by the distance between the translated objective vector and the origin point and diversity is measured by a_i. Hence, when K-RVEA selects an individual in the subpopulation, the goal is to choose the more representative individual balancing both convergence and diversity. Angle penalized distance (APD) metric is adopted as follows:

$$d^i = (1 + P(\theta_i)) \cdot \|\overline{f^i}\|, \tag{10}$$

where $\|\overline{f^i}\|$ is the distance between the translated objective vector of individual i and the origin point. Penalty function $P(\theta_i)$ is defined as:

$$P(\theta_i) = k \cdot \left(\frac{t}{t_{max}}\right)^{\alpha} \cdot \frac{\theta_i}{\gamma_n}, \tag{11}$$

where γ_n is the smallest angle between any two adjacent reference vectors. Parameter α is prefixed and other important parameters are the rate of the number of current generation t and the maximum number of generation t_{max}. The performance of MOEAs on the diversity and convergence varies in different stages. The APD metric adaptively balances convergence and diversity by a penalty function in different stages of the algorithm. In the early stage, convergence is considered more to push the population to the Pareto front and while in the late stage the main goal is to distribute the population uniformly in the Pareto front.

3) Adaption of Reference Vectors: With the searching process, the population is close to the Pareto front. To further improve the diversity, both K-RVEA and RVEA update their reference vectors based on the shape of the obtained Pareto front by the following adaption,

$$v_{t+1,i} = \frac{v_{0,i} \circ \left(z_t^{\max} - z_t^{\min}\right)}{\left\|v_{0,i} \circ \left(z_t^{\max} - z_t^{\min}\right)\right\|}, \tag{12}$$

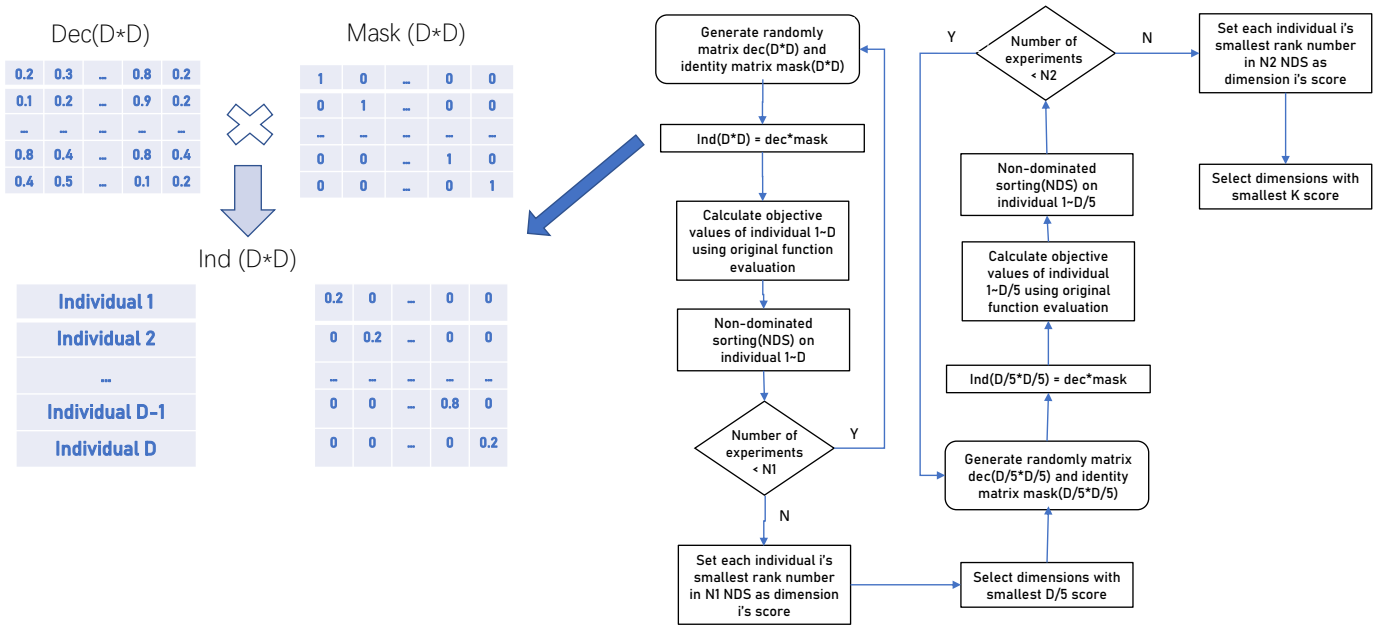

Fig. 2. Feature selection using non-dominated sorting.

where $z_t^{\max}$ and $z_t^{\min}$ are maximum and minimum objective values at the current generation. By updating the reference vectors with the population, a set of objective vectors with a uniform distribution could be obtained when the ranges of different objective functions are different.

4) Updating Kriging Model: During the evolutionary search in K-RVEA, all objective values of individuals are predicted by the Kriging model. The Kriging model in K-RVEA is set at a fixed frequency of updating one time every 20 generations. A key part of model management is to select solutions to be re-evaluated using original expensive function evaluation. Due to the limited computation source, the number of individuals to be re-evaluated is a fixed number (often less than 1000). In K-RVEA, in order to sample the proper individuals and assist the search for the Pareto set, a strategy of dynamically sampling points is used for balancing convergence and diversity. Two kinds of reference vectors are adopted in the strategy: fixed reference vectors V_f which are evenly distributed in objective space and adaptive vectors V_a.

To assess the performance of population on the convergence and diversity, we compare the number of empty vectors in V_f. A fixed reference vector is called empty if no individuals are assigned to it. $\left|V_f^{ia}\right|_{t_u-1}$ stands for the number of empty vectors in fixed reference vectors at the last model updating and $\left|V_f^{ia}\right|_{t_u}$ stands for the number of empty vectors in fixed reference vectors at the current model updating. Then, we decide which indicator to be prioritized by calculating $\left|V_f^{ia}\right|_{t_u} - \left|V_f^{ia}\right|_{t_u-1} - \delta$, if it is positive, diversity should be focused on and we will select individuals based on the uncertainty information provided by the Kriging model, otherwise, the convergence will be focused on. As we can see from RVEA [24], APD is the metric to assess convergence and diversity at the same time, which provides the information for the selection of new sampled points. In short, whether the re-evaluated solutions is chosen using the uncertainty or the APD value depends on the stage of the optimization process.

III. Proposed Algorithm

To solve an expensive sparse MOP, we first apply a feature selection operator to the decision space in the original MOP and select the non-zero dimensions to reformulate the problem. Then, we perform K-RVEA [12] to search the optimal solutions of the re-formulated problem. Finally, the zero dimensions will be added to these obtained optimal solutions to build the solutions to the original MOP as shown in Fig. II-C3.

Our feature selection operation is inspired by SparseEA [2] to select the non-zero dimensions in the Pareto set. Firstly, we randomly generate the real number matrix dec in the size of $D * D$ and a unit matrix $mask$ in the size of $D * D$, where D denotes the number of decision variables. The goal of adopting the $mask$ is to assess the importance of each decision variable. One non-zero dimension in each individual ensures the independency. Then, the data matrix ind is obtained by:

$$(ind_1, ind_2, \ldots) = (\mathrm{dec}_1 \times \mathrm{mask}_1, \mathrm{dec}_2 \times \mathrm{mask}_2, \ldots)\,. \tag{13}$$

From Equation (13), ind is set to be zero except the i-th element in the i-th individual. The population is sorted by non-dominated sorting. Non-dominated sorting assigns ranks to individuals based on the Pareto dominating relation [26]. Using one existing non-dominated sort algorithm [27], we sort the whole population and assign ranks to each individual ind_i. The individual with small rank dominates the individual

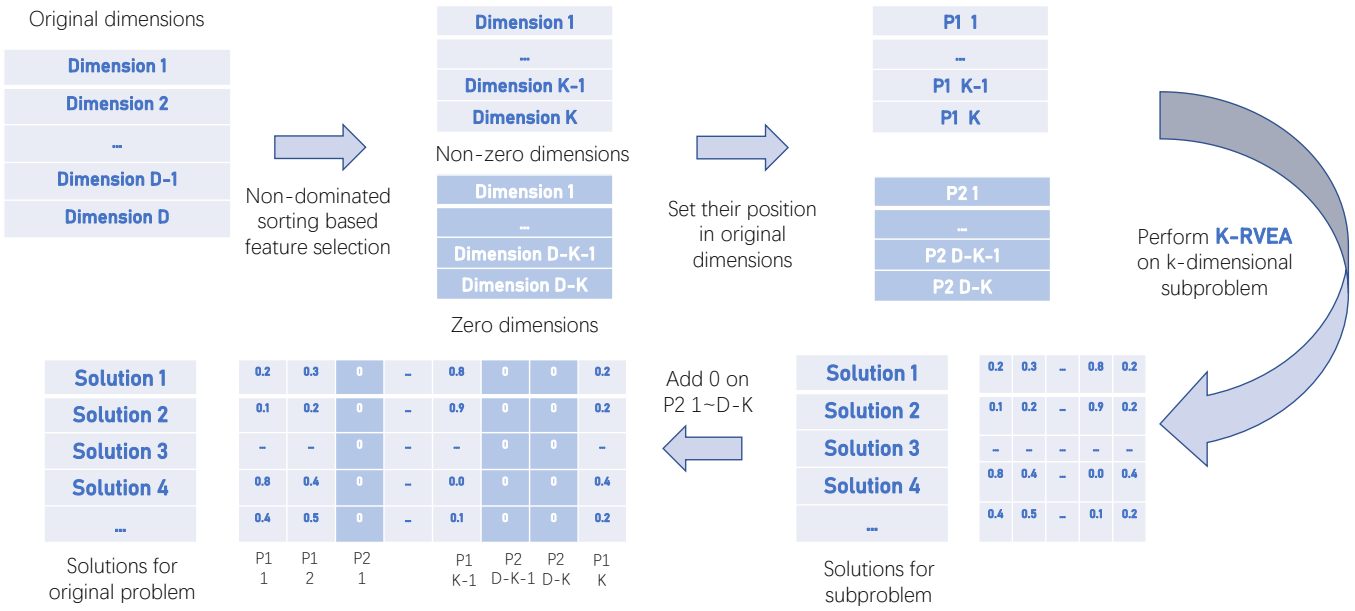

Fig. 3. General process of the proposed algorithm for large-scale expensive sparse MOPs.

with large rank. The rank of each individual measures its performance and the contribution of corresponding decision variable to the objective value. Non-zero decision variables occur in individuals with small ranks, which means they are more likely to be in the non-zero decision variables in the Pareto set. Each individual's rank is regarded as the score for the decision variable and a small score means the high possibility of being the non-zero decision variables.

To improve the robustness of the feature selection method, we repeat this possibility assessment with different ind for several times and use the best score of every dimension in different sortings as its final score.

The process of feature selection will be accurate if we make enough function evaluations and non-dominated sortings. However, to get a trade-off between efficiency and budget, we divide the selection of non-zero dimensions into two steps as shown in Fig. II-C2. In the first step, we generate randomly $N1$ different ind and calculate their objective values using expensive function evaluation, then we make non-dominated sortings on them respectively and set the smallest rank number of each ind_i obtained in $N1$ sorts as the score of dimension i. Considering the robustness of the operator, we select dimensions with smallest $D/5$ score for subsequent selection. Then based on the $D/5$ dimensions, we perform $N2$ times of non-dominated sorts on $N2$ different ind. The score of dimension i is set as the smallest rank number ind_i obtained in $N2$ sorts. At last, we select dimensions with smallest K score as the non-zero dimensions. The whole process of non-dominated sorting based feature selection needs $(N1 * D + N2 * D/5)$ functions evaluations.

As shown in Fig. II-C3, once non-zero dimensions are selected out, the positions of non-zero dimensions and zero dimensions in original dimensions are recorded as $P1$ and $P2$. We modify dimensions of problem to K and then perform K-RVEA on the re-formulated problem. After the execution of K-RVEA, optimal solutions with K dimensions could be obtained. To form the solutions for original problems, individuals with D dimensions are generated and we copy the results of solutions with K dimensions in position of $P1$ and add 0 in positions of $P2$ respectively.

IV. Experimental Results

A. Experiment Settings and Benchmarks

To test the performance of the proposed algorithm K-RVEA(FS), we choose the SMOP test suite with 100 decision variables [2] whose objective functions are shown below in our experiment.

$$\begin{aligned} \text{Minimize} \quad & f_1(\mathrm{x}) = h_1(\mathrm{x})\,(1 + g_{ns}(\mathrm{x}) + g_s(\mathrm{x})) \\ & f_2(\mathrm{x}) = h_2(\mathrm{x})\,(1 + g_{ns}(\mathrm{x}) + g_s(\mathrm{x})) \\ & \cdots. \\ & f_M(\mathrm{x}) = h_M(\mathrm{x})\,(1 + g_{ns}(\mathrm{x}) + g_s(\mathrm{x})) \end{aligned} \tag{14}$$

where h is the shape function which influence the shape of Pareto front, the function g controls fitness landscape, θ is the sparsity parameter. Small θ means the high sparsity of problem. Also, the known number of non-zero decision variables K is defined as follows:

$$K = \lceil \theta(D - M + 1) \rceil, \tag{15}$$

where D is the number of decision variables and M is the number of objectives.

To conduct the proposed algorithm in our experiment on SMOP, $N1$ and $N2$, times of non-dominated sortings on ind in two steps, are set as 4 and 5. The whole process of

TABLE II
ACCURACY OF FEATURE SELECTION USING NCA, RRELIEFF, AND THE PROPOSED NON-DOMINATED SORTING-BASED FEATURE SELECTION.

	NCA(dec)	**NCA**(ind)	**RRelieF**(dec)	**RRelieF**(ind)	**Non-dominated Sorting FS**(ind)
SMOP1	11.23%	13.23%	16.36%	10.86%	**98.18**%
SMOP2	24.52%	34.53%	18.83%	13.63%	**97.57**%
SMOP3	22.72%	50.24%	21.74%	55.06%	**99.69**%
SMOP4	19.23%	18.44%	15.21%	23.63%	**99.39**%
SMOP5	19.56%	22.81%	16.36%	18.29%	**29.09**%
SMOP6	20.39%	19.54%	19.39%	19.45%	**30.92**%
SMOP7	18.53%	43.92%	18.43%	17.09%	**99.39**%
SMOP8	19.42%	20.16%	17.76%	14.54%	**99.09**%

TABLE III
STATISTICAL RESULTS OF THE IGD (AVERAGE AND STANDARD DEVIATION IN BRACKETS) OBTAINED BY COMPARED ALGORITHMS WITH 800 FUNCTION EVALUATIONS ON SMOP1-8. THE BEST RESULTS ARE HIGHLIGHTED

	NSGAII	**SparseEA**	**K-RVEA**	**K-RVEA(FS)**
SMOP1	1.0266(0.0426)+	0.1195(0.0133)+	1.2867(0.2142)+	**0.0293(0.0043)**
SMOP2	1.8761(0.0412)+	0.1974(0.0142)+	2.0899(0.0812)+	**0.0663(0.0075)**
SMOP3	2.2815(0.0384)+	0.1348(0.0303)+	2.5004(0.0574)+	**0.0285(0.0017)**
SMOP4	0.9339(0.0193)+	0.0582(0.0132)+	0.8242(0.0231)+	**0.0033(0.0004)**
SMOP5	0.7264(0.0221)+	0.0854(0.0172)+	0.6946(0.0421)+	**0.0250(0.0026)**
SMOP6	0.3212(0.0225)+	0.0717(0.0082)+	0.2591(0.0317)+	**0.0232(0.0028)**
SMOP7	1.8552(0.0751)+	0.2610(0.0317)+	2.4262(0.2362)+	**0.0928(0.0079)**
SMOP8	3.3539(0.0286)+	0.5247(0.0148)+	3.6724(0.0662)+	**0.1902(0.0472)**

$+$ means K-RVEA(FS) shows a statistically better performence.
$-$ means K-RVEA(FS) shows a statistically worse performence.
$\approx$ means K-RVEA(FS) doesn't show a statistically diffrent performence.

non-dominated sorting based feature selection needs $5D(500)$ functions evaluations. For Kriging-assisted algorithms, K-RVEA and K-RVEA(FS), the initial training dataset is set as 100 individual with their real function evaluations. In evolution, 5 re-evaluated individuals are selected in each generation, and the Kriging models update in every 20 generations. All the compared algorithms stop by 800 real function evaluations.

B. Effects of Feature Selection Techniques

In this subsection, we compare our proposed feature selection operation with classical feature selection methods, NCA and RReliefF, on the 100-dimensional SMOP test suite ($\theta = 0.1$), where the non-zero dimensions are known in [2] and can be the ground truth for the performance assessment. In the comparative experiment, we use two kinds of $5D$ sampled datasets from the SMOP test suite: one is obtained by dec in Equation (13) and the other is obtained by ind in Equation (13). NCA and RReliefF can be applied to both datasets, but the proposed non-dominated sorting based feature selection can be only applied to the ind dataset. We calculate the the accuracy of the selected non-zero decision variables to evaluate the performance of the compared methods.

The accuracy of three algorithms on two datasets has been presented in Table IV. We can see that our proposed feature selection operation finds the non-zero dimensions in the Pareto set accurately, while compared algorithms cannot find the target dimension mostly. In most problems, non-dominated feature selection could find all the non-zero dimensions.

As mentioned above, a large number of feature selection algorithms require a large amount of data. If it is a small set of data with only $5D$ sampled points, it is difficult to perform well. The accuracy of most existing feature selection methods is guaranteed by the sufficient data. Also, the proposed feature selection method cannot guarantee that the exactly correct result and this is the reason why we applied the feature selection operation to the $5D$ dataset rather than the D dataset. Therefore, the proposed feature selection operation is effective for the dimension reduction of expensive sparse MOPs.

C. Comparative Experiments

In this subsection, we discuss the behavior of K-RVEA(FS) by comparing with three different algorithms: NSGA-II [28], SparseEA [2], and K-RVEA [12] on SMOP1-8 with 100 decision variables.

- NSGA-II is a popular Pareto-based MOEA using non-dominated sort and crowding distance. We choose NSGA-II as a compared algorithm as a baseline.
- SparseEA is a new and effective MOEA for sparse MOPs [2], because it reduces the search space by finding important decision variables. We choose SparseEA as a compared algorithm as a representative MOEA for sparse MOPs.
- K-RVEA is a representative surrogate-assisted MOEA for expensive MOPs.

All the compared algorithms run 20 independent times and stop by 800 expensive function evaluations. We use inverse generation distance (IGD) [29] that assesses both convergence

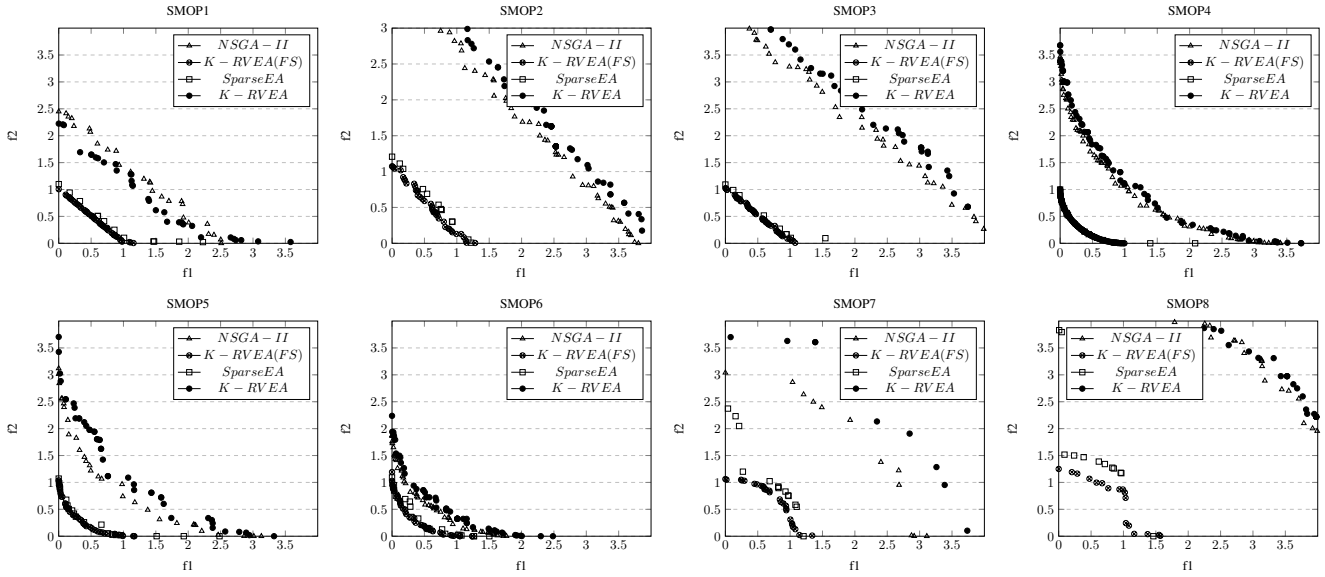

Fig. 4. Obtained Pareto fronts of NSGA-II, SparseEA, K-RVEA, and K-RVEA(FS) with the median IGD value on SMOP1-8.

TABLE IV
STATISTICAL RESULTS OF TIME(S) (AVERAGE AND STANDARD DEVIATION IN BRACKETS) SPENT BY K-RVEA AND K-RVEA(FS) ON SMOP1-8. THE BEST RESULTS ARE HIGHLIGHTED.

	K-RVEA	K-RVEA(FS)
SMOP1	5.8074e+2(2.16e+1)+	**1.6257e+1**(3.66e-1)
SMOP2	5.7166e+2(1.62e+1)+	**1.5892e+1**(2.91e-1)
SMOP3	5.6747e+2(1.63e+1)+	**1.5334e+1**(2.85e-1)
SMOP4	5.7236e+2(7.64e+0)+	**1.6021e+1**(1.94e-1)
SMOP5	5.6843e+2(1.01e+1)+	**1.5837e+1**(2.18e-1)
SMOP6	5.7109e+2(1.38e+1)+	**1.6445e+1**(1.43e-1)
SMOP7	5.6881e+2(1.26e+1)+	**1.4527e+1**(3.15e-1)
SMOP8	5.7014e+2(9.16e+0)+	**1.5853e+1**(2.17e-1)

$+$ means K-RVEA(FS) shows a statistically better performence.
$-$ means K-RVEA(FS) shows a statistically worse performence.
$\approx$ means K-RVEA(FS) doesn't show a statistically diffrent performence.

and diversity to evaluate the performance of compared algorithms. The results are shown in Table IV, where a Wilcoxon rank sum test at a significance level of 0.05 [30] is applied to the results and K-RVEA(FS) is set as the control algorithm. Also Fig. 4 shows the obtained Pareto fronts of NSGA-II, SparseEA, K-RVEA, and K-RVEA(FS) with median IGD values on eight problems.

From Fig. 4 and Table IV, we can see that K-RVEA(FS) obtains the best Pareto fronts and IGD values on SMOP1-8. First of all, using the proposed feature selection operation, the dimensions of the SMOPs are reduced correctly. In the optimization process of sparse MOPs, the searching resources of MOEAs without feature selection are consumed by the searching for zero dimensions. In other words, high-dimensional decision space can be reduced with the heuristic information.

From the poor performance of K-RVEA, we could see that the Kriging model is hard to deal with high-dimensional problems due to the approximation error. The proposed feature selection operation selects non-zero dimensions and search in such a small subspace will be efficient. Prior knowledge of non-zero dimensions will also help the dimension recovery and we could obtain the proper solutions to the original problems.

Compared with SparseEA which also utilizes the heuristic information, we successfully reduced the number of required function evaluations by introducing a surrogate model. When we deal with the expensive function evaluations, the cost of SparseEA, which requires a large number of function evaluations, will become high and even unaffordable. Therefore, by effectively using a surrogate model, K-RVEA(FS) can obtain a better performing results with less computing resources than SparseEA.

Further, we compare the execution time of K-RVEA and K-RVEA(FS) in Table IV-C. K-RVEA(FS) uses significantly shorter time than K-RVEA. It is clear that the proposed feature selection operation can save time for both model building and evolutionary search.

Therefore, we can obtain three observations: 1) the proposed algorithm can effectively select non-zero dimensions; 2) using surrogate model can save the number of expensive function evaluations, which makes it applicable to many real-world sparse MOPs; 3) the proposed algorithm can obtain results with satisfactory convergence and diversity within the limited computing resources.

V. CONCLUSIONS

To address large-scale expensive sparse MOPs, we propose a Kriging-assisted MOEA with a non-dominated sorting based feature selection. By using the proposed feature selection operator, we select out the non-zero dimensions in the Pareto set. Based on the selected decision variables, we perform K-RVEA on the reformulated problem. After we obtain the

optimal solutions to the reformulated problem, we add the zero dimensions to obtained solutions as the solutions to the original problem. The proposed algorithm has been tested on eight benchmark problems with limited expensive function evaluations. Comparing with other existing representative MOEAs, the proposed algorithm obtains the best result based on the IGD metric.

For large-scale expensive sparse multi-objective optimization problems, dimension reduction plays a key role for the optimization process. Feature selection is performed before evolutionary search and the non-zero dimensions are fixed in our method. However, the feature selection method may not be reliable. Therefore, dynamic feature selection [31] or dimension-reduction methods can be adopted to improve accuracy of selection for non-zero dimensions. In addition, for the real sparse problems, the domain knowledge could provide the information for the feature selection and model management to improve the performance of algorithms, especially the modification of model management on different optimization problems is worthy of attention.

REFERENCES

[1] H. Wang, L. Jiao, and X. Yao, "Two_arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 524–541, 2014.

[2] Y. Tian, X. Zhang, C. Wang, and Y. Jin, "An evolutionary algorithm for large-scale sparse multi-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, 2019.

[3] J. E. Fieldsend and S. Singh, "Pareto evolutionary neural networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 338–354, 2005.

[4] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen *et al.*, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007, vol. 5.

[5] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.

[6] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[7] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2013.

[8] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *IEEE Computational Intelligence Magazine*, vol. 4, no. 3, pp. 62–76, 2009.

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

[10] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: an overview and case studies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 442–458, 2018.

[11] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by moea/d with gaussian process model," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 456–474, 2009.

[12] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 129–142, 2016.

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

[14] A. Globerson and N. Tishby, "Sufficient dimensionality reduction," *Journal of Machine Learning Research*, vol. 3, no. Mar, pp. 1307–1331, 2003.

[15] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[16] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[17] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial intelligence*, vol. 97, no. 1-2, pp. 245–271, 1997.

[18] K. Kira, L. A. Rendell *et al.*, "The feature selection problem: Traditional methods and a new algorithm," in *Aaai*, vol. 2, 1992, pp. 129–134.

[19] H. Almuallim and T. G. Dietterich, "Learning with many irrelevant features." in *AAAI*, vol. 91. Citeseer, 1991, pp. 547–552.

[20] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[21] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[22] P. Mitra, C. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 3, pp. 301–312, 2002.

[23] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[24] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[25] J. A. Cornell, *Experiments with mixtures: designs, models, and the analysis of mixture data*. John Wiley & Sons, 2011, vol. 403.

[26] H. Wang and X. Yao, "Corner sort for pareto-based many-objective optimization," *IEEE transactions on cybernetics*, vol. 44, no. 1, pp. 92–102, 2013.

[27] Y. Tian, H. Wang, X. Zhang, and Y. Jin, "Effectiveness and efficiency of non-dominated sorting for evolutionary multi-and many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 4, pp. 247–263, 2017.

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

[29] C. A. C. Coello and N. C. Cortés, "Solving multiobjective optimization problems using an artificial immune system," *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, 2005.

[30] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in statistics*. Springer, 1992, pp. 196–202.

[31] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Dynamic feature space and incremental feature selection for the classification of textual data streams," *Knowledge Discovery from Data Streams*, pp. 107–116, 2006.