

Conjugate Surrogate for Expensive Multiobjective Optimization

Qi-Te Yang, *Student Member, IEEE*, Liu-Yue Luo, *Student Member, IEEE*, Xin-Xin Xu, Chun-Hua Chen, *Member, IEEE*, (Corresponding Author), Hua Wang, *Senior Member, IEEE*, Jun Zhang, *Fellow, IEEE*, Zhi-Hui Zhan, *Senior Member, IEEE* (Corresponding Author)

Abstract—The Kriging surrogate (KS) has been widely used in surrogate-assisted multiobjective evolutionary algorithms (SAMOEAs) for solving expensive multiobjective optimization problems (EMOPs). Typically, when tackling an M -objective EMOP, a KS consists of M Kriging models, in which each model is used to approximate one objective function to replace the expensive fitness evaluation. Since such a KS is only efficient in solving low-dimensional EMOPs, the dimension reduction method has been adopted to construct the reduction surrogate (RS) to reduce training costs. However, both KS and RS can only approximate the mapping from variables to different objectives (i.e., objective function) but ignore the potential relationship between objectives. For practical applications, it is necessary to take into account the mapping between different objectives for the reliability of the surrogate. Therefore, we for the first time propose the concept of the conjugate surrogate (CS) and construct a simple CS to realize the approximated mapping from objectives to objectives. Different from KS or RS, all models in CS are conjugate symbiosis. In collaboration with RS, CS can not only benefit the light training cost, but also improve the convergence speed. Compared with five state-of-the-art SAMOEAs, the CS-assisted algorithm shows its effectiveness and competitiveness in solving EMOPs.

Keywords—Kriging, Surrogate-assisted, Expensive Multiobjective Optimization, Evolutionary Computation

I. INTRODUCTION

After decades of development, a large number of multiobjective evolutionary algorithms (MOEAs) have been proposed for solving multiobjective optimization problems (MOPs) [1]-[5]. These MOEAs can find a set of well-performed solutions from both convergence and diversity perspectives whereas a lot of real objective fitness evaluation (FE) is required to support this optimization process. However, such real FE in many practical multiobjective optimization applications is time-consuming or financially costly, and those

This work was supported in part by the National Key Research and Development Program of China under Grant 2022ZD0120001 and in part by the National Natural Science Foundations of China (NSFC) under Grant 62176094.

Qi-Te Yang, Liu-Yue Luo, and Zhi-Hui Zhan are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, P. R. China and also with the College of Artificial Intelligence, Nankai University, Tianjin 300350, P. R. China. Email: zhanapollo@163.com.

Xin-Xin Xu is with the School of Computer Science and Technology, Ocean University of China, Qingdao 266100, P. R. China

Chun-Hua Chen is with the School of Software Engineering, South China University of Technology, Guangzhou 510006, P. R. China.

Hua Wang is with the Institute for Sustainable Industries and Liveable Cities, Victoria University, Melbourne, VIC 8001, Australia.

Jun Zhang is with the College of Artificial Intelligence, Nankai University, Tianjin 300350, P. R. China and also with Hanyang University, Ansan 15588, South Korea.

MOPs whose FE has this property are called expensive MOPs (EMOPs) [6][7]. For these EMOPs, large amounts of FE are not available to support the optimization process of MOEAs. Surrogate-assisted MOEAs (SAMOEAs) [8][9][10] have thus been widely studied by scholars in recent years.

Different from canonical MOEAs, SAMOEAs gather some solutions that have been real-evaluated and use them to train a surrogate. During the optimization process, the objective values of each newly generated solution are predicted by this surrogate to ease the reliance on real FE. Such a surrogate is usually composed of machine learning models like radial basis function network, support vector machine, or feedforward neural network [11]. Kriging model [12], which is also known as the Gaussian process, is one of the widely used models for building surrogates in SAMOEAs. This is due to the ease of building a surrogate with Kriging models for EMOPs. For an EMOP with M objectives, one Kriging model is often used to approximate each objective function f_m , where $m \in \{1, \dots, M\}$, and the fusion of these Kriging models in all objectives form the surrogate. Herein, we call it a Kriging surrogate (KS). Therefore, each Kriging model ks_m in a KS is the approximated regression mapping from all variables to objective f_m , as:

$$ks_m = F(\{x_1, \dots, x_D\}) \rightarrow f_m \quad (1)$$

where $F(a) \rightarrow b$ is the mapping from a to b .

However, many studies have shown that this KS is only suitable for problems with variable dimensions below 20 [13][14], since the performance of an ideal Kriging model is correlated with the quantity of data, and the higher the variable dimension of data, the more data is required [15]. Such a large number of data will also increase the training time cost. Therefore, some studies tried to pre-process database (DB) through the strategy of dimension reduction [16][17]. Specifically, by mathematical or random means, only several variables are selected to participate in one Kriging model training, so that fewer data are required and the training time cost can also be reduced. Herein, the surrogate in which each model is trained through this dimension reduction method is called a reduction surrogate (RS). Each model rs_m in an RS is the approximated regression mapping from some selected variables in X_m to objective f_m , as:

$$rs_m = F(X_m) \rightarrow f_m \quad (2)$$

where X_m is a variable subset selected from all variables $\{x_1, x_2, \dots, x_D\}$ and $0 < |X_m| < D$. It is worth noting that such an RS may suffer from misfit caused by the insufficiency of selected variables. Thus RS is usually not used alone. For example, in [16], several different RSs are adapted to form an ensemble surrogate to alleviate the performance deterioration.

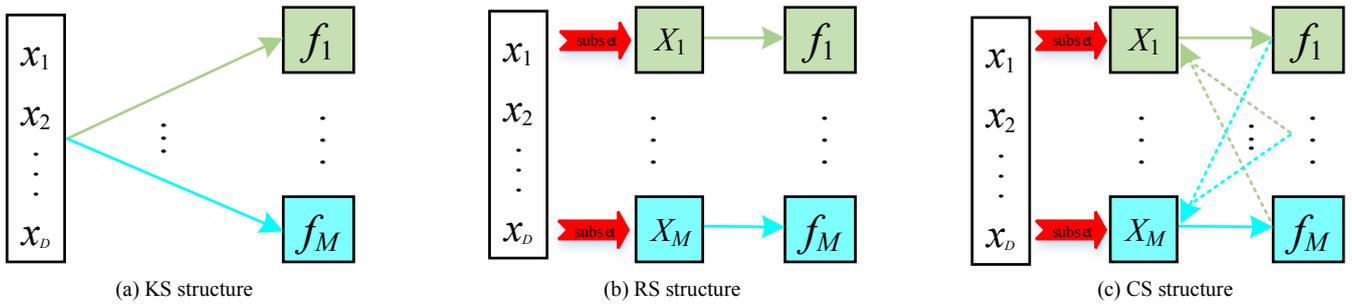


Fig. 1. The basic structures of KS, RS, and CS in SAMOEAs.

As can be seen from the above introduction, all Kriging models in KS/RS are completely independent of each other, and each of them can only fit the approximated mapping from variables to a single objective. However, such a model structure in KS/RS cannot explore the relationship between different objectives, but there may be some connection among objectives in real-world EMOPs. This connection may be linear or non-linear, positively or negatively correlated. For example, the car cab design application is a real-world EMOP with nine objectives including fuel economy, acceleration time, and car weight [18]. In this application, there must be some connection between these three objectives intuitively, such as a heavier car inevitably leads to a longer acceleration time.

Therefore, in this paper, we explore a new surrogate structure to fit the approximated objectives-to-objective mapping. For such a surrogate, models in it are not independent. In other words, the output of one model may be the input for another model, and vice versa. Therefore, models in this surrogate are conjugate with each other, and thus this surrogate is called a conjugate surrogate (CS).

In conclusion, the core of CS is to achieve the approximated fitting of the entanglement among objectives, and the causation of this entanglement is the sharing of variables. Therefore, the mapping from other objectives f_n ($n = 1, \dots, M$ and $n \neq m$) to objective f_m can be divided into two steps: the inverse mapping from other objectives f_n to shared variables (i.e., the inverse form of Eq. (1)) and the mapping from shared variables to f_m . In this way, these shared variables are used as intermediates to realize the connection between the two steps. Fig. 1 shows the basic structures of KS, RS, and CS. The arrows of each color represent the mapping paths to the corresponding color objective, in which the solid arrows mean the mapping paths from variables to one objective and the dotted arrows mean the inverse mapping paths from objectives to variables. X_m ($m \in 1, \dots, M$) is a variable subset selected from $\{x_1, x_2, \dots, x_D\}$ to participate in the model training for f_m , and it is also treated as the shared variable subset when constructing mapping from other objectives f_n to f_m in Fig. 1 (c).

The main contributions of this paper can be summarized as follows:

- We propose a new method for building an RS, in which the variable dimension is reduced through feature selection.
- We for the first time propose the concept of CS and gives the basic structure of CS for expensive multiobjective optimization. In addition, we also propose a method for building a simple CS.

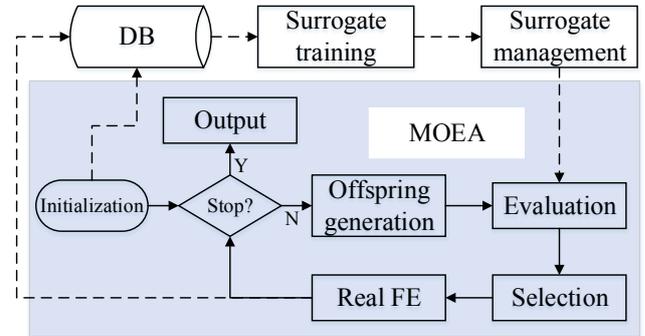


Fig. 2. The flowchart of a SAMOEA.

- We propose a surrogate management method to coordinate RS and CS working in SAMOEAs. Theoretically, this surrogate management method can be implemented into any SAMOEAs that use Kriging models to approximate the objective functions.

The rest of this paper is organized as follows. First, we briefly introduce some background information about EMOPs and SAMOEAs in Section II. Then in Section III, we illustrate the details of our proposed surrogate method including the RS, CS, and surrogate management. Section IV presents the experimental results and analysis. Finally, Section V gives the conclusion of this paper.

II. BACKGROUNDS

MOPs are optimization problems that involve multiple conflicting objectives simultaneously. A MOP with M objectives to be minimized can be defined as:

$$\begin{aligned} \min : \mathbb{F} &= \{f_1(\mathbf{x}), \dots, f_M(\mathbf{x})\} \\ \text{s.t. } \mathbf{x} &= (x_1, \dots, x_D)^T \in \Omega \end{aligned} \quad (3)$$

where Ω is the search space. EMOPs have the same expression as MOPs, except that the FE of each objective f_m ($m = 1, \dots, M$) in EMOPs is expensive.

SAMOEAs are a kind of algorithm that can solve these EMOPs efficiently in recent years. Fig. 2 shows the basic flowchart of a SAMOEA. When faced with an EMOP to be solved, the SAMOEA first initializes some solutions and re-evaluates them. Then the flow will be split into two (i.e., solid arrows and dotted arrows). The solid arrows represent the flow of an MOEA, and the dotted arrows represent the flow of data and the surrogate. Next, we will focus on how the surrogate works during the MOEA process. More specifically, these evaluated solutions form the DB, after which a surrogate is trained through the DB. Then, this well-trained surrogate will replace the real FE and give the predicted objective values to each un-evaluated offspring solution via the surrogate management. Thus, newborn solutions can

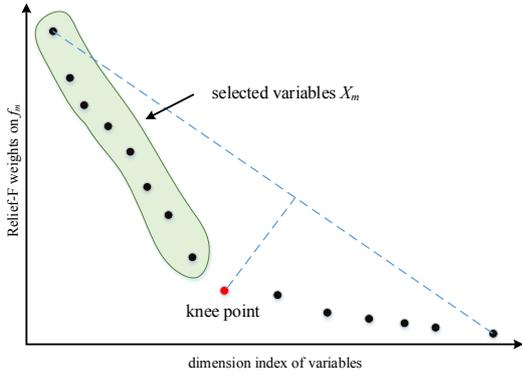


Fig. 3. Variable selection for rs_m in RS.

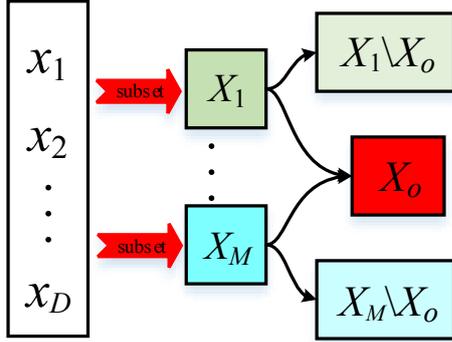


Fig. 4. Variable partition in our proposed CS.

avoid expensive FE so as to reduce computational costs. After the MOEA has completed a search for a certain period, some solutions will be selected for real FE to ensure that the search trend is convergent. These real-evaluated solutions will be added to the DB to further update the surrogate. Therefore, the surrogate training in SAMOEAs can be seen as an incremental learning process.

III. PROPOSED SURROGATE METHOD

In this section, we illustrate the details of our proposed surrogate method for expensive multiobjective optimization.

A. Reduction Surrogate

Considering that the dimension reduction for each rs_m in RS is to select some variables X_m from $\{x_1, x_2, \dots, x_D\}$, the correlation between selected variables and f_m will affect the fitting performance of rs_m . In other words, if the selected variables in X_m have a higher correlation with f_m , then the final fitting degree on f_m of rs_m will be closer to ks_m , thus minimizing the performance deterioration caused by dimension reduction.

Based on the above analysis, dimension reduction for rs_m is more like a feature selection operation [19]. Therefore, we propose a new dimension reduction method through feature selection for building an RS, as shown in Fig. 3. In detail, for objective f_m , the correlations between variables x_d ($d = 1, \dots, D$) and f_m are calculated by Relief-F [20] based on all samples in DB. Then all variables are sorted in descending order according to their Relief-F weights, and a line is further obtained by connecting variables with the largest and smallest Relief-F weights. The variable having the largest distance to this line is regarded as the knee point. Finally, variables whose Relief-F weights are larger than that of this knee point are selected for participating in the training of rs_m . Notice that this method of dimension reduction is derived from [21]. However, if the number of variables finally selected for rs_m is

Algorithm 1 Surrogate training

Input: DB ;
Output: RS and CS .
Begin
1: **For** $m = 1 : M$ **do**
2: calculating Relief-F weights between variables and f_m in DB ;
3: sorting weights in descending order, and finding knee point;
4: $X_m =$ variables whose weights are larger than knee point;
5: **If** $|X_m| < 0.5D$ **then**
6: $X_m =$ first half of variables with high Relief-F weights;
7: **End if**
8: training rs_m through $DB(\langle X_m, f_m \rangle)$;
9: **End for**
10: $RS = (rs_1, \dots, rs_M)$;
11: $X_o = X_1 \cap X_2 \cap \dots \cap X_M$;
12: **For** $m = 1 : M$ **do**
13: training cs_m through $DB(\langle \{f_1, \dots, f_{m-1}, f_{m+1}, \dots, f_M, X_m \setminus X_o\}, f_m \rangle)$;
14: **End for**
15: $CS = (cs_1, \dots, cs_M)$;
End

Algorithm 2 Surrogate-assisted FE

Input: un-evaluated solution \mathbf{x} , RS , CS ;
Output: $\{f_i(\mathbf{x}), \dots, f_M(\mathbf{x})\}$ and $\{e_1(\mathbf{x}), \dots, e_M(\mathbf{x})\}$
Begin
1: **For** $m = 1 : M$ **do**
2: $\{rf_m(\mathbf{x}), re_m(\mathbf{x})\} = rs_m(X_m)$;
3: **End for**
4: **For** $m = 1 : M$ **do**
5: $\{cf_m(\mathbf{x}), ce_m(\mathbf{x})\} = cs_m(rf_1(\mathbf{x}), \dots, rf_{m-1}(\mathbf{x}), rf_{m+1}(\mathbf{x}), \dots, rf_M(\mathbf{x}), X_m \setminus X_o)$;
6: **If** $ce_m(\mathbf{x}) < re_m(\mathbf{x})$ **then**
7: $f_m(\mathbf{x}) = cf_m(\mathbf{x})$; $e_m(\mathbf{x}) = ce_m(\mathbf{x})$;
8: **Else**
9: $f_m(\mathbf{x}) = rf_m(\mathbf{x})$; $e_m(\mathbf{x}) = re_m(\mathbf{x})$;
10: **End if**
11: **End for**
End

lower than a threshold K (K is set as $0.5D$ in this paper), then the first K variables with high Relief-F weights after sorting are selected. After dimension reduction, the variable subset for training rs_m is denoted as X_m . As a result of the above approach, the dimension of variables adopted for training rs_m in the RS is reduced from D to $|X_m|$, where $K \leq |X_m| < D$.

B. Conjugate Surrogate

Although the form of RS makes surrogate training less costly and suitable for higher-dimensional EMOPs, it does so at the expense of prediction performance. In this paper, we take a new approach and propose the idea of building a CS in expensive multiobjective optimization, and the collaboration between CS and RS can alleviate the performance deterioration in RS.

In Fig. 1 (c), we discuss the basic structure of CS, but it can be seen that too many shared variable subsets need to be set if CS is built following that structure. This greatly increases the difficulty and complexity of building CS. Therefore, herein, we propose a method to build a simpler CS, in which only a single shared variable subset is used for the mapping among all objectives.

Referring to the concept of inverse model, we can find the approximated inverse mapping from f_m to variables X_m by a model. That is, $F'(f_m) \rightarrow X_m$, where $F'(b) \rightarrow a$ is the inverse function of $F(a) \rightarrow b$. In addition, there must be an intersection among all X_m ($m = 1, \dots, M$), as:

$$X_o = X_1 \cap \dots \cap X_M \quad (4)$$

Therefore, $F'(f_m) \rightarrow X_m$ can be further divided into $F'(f_m) \rightarrow X_o$ and $F'(f_m) \rightarrow X_m \setminus X_o$. Furthermore, $F(X_m) \rightarrow f_m$ can be

TABLE I
THE STATISTICAL IGD RESULTS BETWEEN CS-RVEA AND OTHER SAMOEAS

Problem	M	CS-RVEA	MOEA/D-EGO	GP-ARMOEA	HeE-MOEA	AB-SAEA	K-RVEA	RS-RVEA
DTLZ1	3	3.23e+2 (4.48e+1)	3.19e+2 (6.09e+1) =	3.54e+2 (3.50e+1) +	3.70e+2 (3.49e+1) +	3.30e+2 (3.84e+1) =	3.17e+2 (4.82e+1) =	3.21e+2 (2.07e+1) =
	5	2.46e+2 (3.46e+1)	2.24e+2 (3.53e+1) =	2.74e+2 (2.58e+1) +	2.62e+2 (2.85e+1) =	2.47e+2 (3.99e+1) =	2.38e+2 (3.28e+1) =	2.51e+2 (2.78e+1) =
	8	1.49e+2 (2.97e+1)	1.44e+2 (2.02e+1) =	1.80e+2 (2.63e+1) +	1.89e+2 (2.33e+1) +	1.55e+2 (3.72e+1) =	1.46e+2 (2.79e+1) =	1.67e+2 (2.33e+1) =
DTLZ2	3	6.04e-1 (1.00e-1)	5.96e-1 (4.47e-2) =	6.71e-1 (1.34e-1) =	3.51e-1 (3.09e-2) -	6.37e-1 (9.19e-2) =	7.13e-1 (6.64e-2) +	5.88e-1 (8.81e-2) =
	5	7.24e-1 (7.23e-2)	7.98e-1 (8.12e-2) +	9.03e-1 (4.98e-2) +	5.48e-1 (3.44e-2) -	7.83e-1 (5.35e-2) +	8.20e-1 (5.26e-2) +	7.30e-1 (6.05e-2) =
	8	7.51e-1 (7.93e-2)	9.66e-1 (3.79e-2) +	9.40e-1 (5.72e-2) +	8.20e-1 (3.72e-2) +	8.45e-1 (8.70e-2) +	7.64e-1 (5.07e-2) =	7.66e-1 (6.62e-2) =
DTLZ3	3	9.41e+2 (8.67e+1)	7.05e+2 (2.15e+2) -	1.07e+3 (1.09e+2) +	1.04e+3 (1.17e+2) +	9.80e+2 (1.45e+2) =	8.78e+2 (1.03e+2) -	8.91e+2 (1.57e+2) =
	5	7.78e+2 (1.17e+2)	5.21e+2 (1.28e+2) -	9.14e+2 (8.86e+1) +	9.08e+2 (1.02e+2) +	8.30e+2 (1.05e+2) =	8.30e+2 (7.27e+1) =	7.81e+2 (8.24e+1) =
	8	5.27e+2 (6.83e+1)	4.32e+2 (1.12e+2) -	6.97e+2 (1.11e+2) +	6.77e+2 (6.54e+1) +	6.17e+2 (1.09e+2) +	5.31e+2 (7.46e+1) =	5.39e+2 (8.30e+1) =
DTLZ4	3	7.91e-1 (1.22e-1)	1.16e+0 (9.44e-2) +	1.15e+0 (1.68e-1) +	1.03e+0 (2.37e-2) +	9.08e-1 (1.78e-1) +	1.01e+0 (1.41e-1) +	9.60e-1 (1.27e-1) +
	5	9.41e-1 (1.02e-1)	1.21e+0 (7.15e-2) +	1.24e+0 (9.99e-2) +	1.14e+0 (2.83e-2) +	1.04e+0 (1.28e-1) +	1.06e+0 (1.30e-1) +	1.02e+0 (1.13e-1) +
	8	9.54e-1 (8.05e-2)	1.20e+0 (5.32e-2) +	1.20e+0 (4.30e-2) +	1.18e+0 (4.36e-2) +	1.09e+0 (7.54e-2) +	1.02e+0 (7.63e-2) +	1.07e+0 (6.39e-2) +
DTLZ5	3	4.68e-1 (9.40e-2)	5.64e-1 (7.94e-2) +	5.72e-1 (1.64e-1) +	2.63e-1 (3.42e-2) -	5.45e-1 (9.58e-2) =	5.67e-1 (1.23e-1) +	4.08e-1 (7.83e-2) -
	5	3.96e-1 (1.01e-1)	5.36e-1 (6.82e-2) +	6.14e-1 (1.30e-1) +	2.37e-1 (2.91e-2) -	4.91e-1 (1.18e-1) +	4.83e-1 (8.42e-2) +	3.84e-1 (8.33e-2) =
	8	3.06e-1 (5.75e-2)	4.25e-1 (5.61e-2) +	4.80e-1 (6.73e-2) +	1.82e-1 (2.16e-2) -	3.76e-1 (7.06e-2) +	3.40e-1 (7.73e-2) =	2.96e-1 (6.71e-2) =
DTLZ6	3	1.07e+1 (6.83e-1)	8.49e+0 (1.35e+0) -	1.35e+1 (5.73e-1) +	1.53e+1 (2.20e-1) +	1.31e+1 (8.90e-1) +	1.03e+1 (7.93e-1) =	1.10e+1 (9.41e-1) =
	5	1.00e+1 (9.45e-1)	7.61e+0 (1.16e+0) -	1.26e+1 (3.71e-1) +	1.36e+1 (1.54e-1) +	1.12e+1 (8.04e-1) +	1.02e+1 (7.51e-1) =	1.07e+1 (6.93e-1) +
	8	9.14e+0 (6.27e-1)	6.29e+0 (1.20e+0) -	1.00e+1 (3.47e-1) +	1.10e+1 (1.18e-1) +	9.90e+0 (4.98e-1) +	8.23e+0 (5.89e-1) -	9.23e+0 (4.21e-1) =
WFG1	3	1.86e+0 (1.06e-1)	2.22e+0 (6.81e-2) +	2.13e+0 (7.36e-2) +	2.28e+0 (4.59e-2) +	1.96e+0 (1.35e-1) +	1.85e+0 (1.57e-1) =	1.86e+0 (1.10e-1) =
	5	2.32e+0 (9.05e-2)	2.56e+0 (7.46e-2) +	2.50e+0 (5.01e-2) +	2.63e+0 (6.68e-2) +	2.42e+0 (1.18e-1) +	2.36e+0 (7.55e-2) =	2.35e+0 (8.72e-2) =
	8	2.92e+0 (7.04e-2)	3.09e+0 (4.43e-2) +	3.00e+0 (4.08e-2) +	3.13e+0 (3.71e-2) +	3.03e+0 (7.77e-2) +	2.92e+0 (5.30e-2) =	2.96e+0 (5.45e-2) =
WFG2	3	6.69e-1 (3.60e-2)	7.99e-1 (5.30e-2) +	7.37e-1 (5.84e-2) +	7.64e-1 (5.15e-2) +	7.63e-1 (1.03e-1) +	7.11e-1 (3.11e-2) +	6.87e-1 (3.62e-2) =
	5	1.07e+0 (6.40e-2)	1.45e+0 (2.26e-1) +	1.18e+0 (9.96e-2) +	1.62e+0 (2.20e-1) +	1.04e+0 (2.22e-1) -	1.06e+0 (9.58e-2) =	1.08e+0 (8.91e-2) =
	8	1.60e+0 (1.61e-1)	2.45e+0 (2.64e-1) +	1.85e+0 (2.37e-1) +	2.86e+0 (3.77e-1) +	1.99e+0 (5.77e-1) +	1.57e+0 (9.77e-2) =	1.65e+0 (1.05e-1) =
WFG3	3	6.60e-1 (3.34e-2)	6.81e-1 (1.84e-2) +	6.89e-1 (2.68e-2) +	4.94e-1 (2.17e-2) -	6.42e-1 (3.37e-2) =	6.93e-1 (2.42e-2) +	6.50e-1 (3.97e-2) =
	5	9.31e-1 (4.06e-2)	9.50e-1 (2.83e-2) =	9.74e-1 (2.38e-2) +	6.87e-1 (2.85e-2) -	9.45e-1 (5.63e-2) =	9.51e-1 (5.02e-2) =	9.40e-1 (4.12e-2) =
	8	1.34e+0 (4.96e-2)	1.30e+0 (4.57e-2) -	1.36e+0 (4.60e-2) =	9.65e-1 (6.33e-2) -	1.34e+0 (6.42e-2) =	1.34e+0 (5.73e-2) =	1.37e+0 (4.57e-2) =
WFG4	3	5.07e-1 (1.23e-2)	5.73e-1 (5.75e-2) +	5.36e-1 (2.17e-2) +	6.18e-1 (3.79e-2) +	5.16e-1 (3.56e-2) =	5.26e-1 (1.40e-2) +	5.33e-1 (1.90e-2) +
	5	1.34e+0 (4.72e-2)	1.92e+0 (2.46e-1) +	1.52e+0 (1.89e-1) +	2.32e+0 (1.71e-1) +	1.46e+0 (2.71e-1) =	1.40e+0 (1.07e-1) +	1.36e+0 (5.11e-2) +
	8	4.27e+0 (4.83e-1)	4.94e+0 (1.07e+0) =	4.22e+0 (3.77e-1) =	6.57e+0 (3.01e-1) +	4.17e+0 (5.40e-1) =	4.10e+0 (4.13e-1) =	4.33e+0 (2.84e-1) =
WFG5	3	5.91e-1 (3.58e-2)	6.79e-1 (3.26e-2) +	6.62e-1 (1.46e-2) +	7.62e-1 (1.35e-2) +	6.77e-1 (2.42e-2) +	5.62e-1 (4.06e-2) -	5.83e-1 (3.56e-2) =
	5	1.35e+0 (2.43e-2)	1.56e+0 (7.77e-2) +	1.51e+0 (9.93e-2) +	1.67e+0 (3.77e-2) +	1.44e+0 (3.67e-2) +	1.38e+0 (4.34e-2) =	1.37e+0 (3.99e-2) =
	8	3.44e+0 (1.03e-1)	4.58e+0 (3.11e-1) +	4.11e+0 (5.20e-1) +	4.35e+0 (1.45e-1) +	3.59e+0 (2.74e-1) +	3.54e+0 (1.96e-1) =	3.59e+0 (2.31e-1) +
WFG6	3	8.13e-1 (2.65e-2)	8.95e-1 (5.97e-2) +	8.69e-1 (1.46e-2) +	7.94e-1 (2.67e-2) -	8.43e-1 (2.50e-2) +	8.07e-1 (2.90e-2) =	8.19e-1 (1.59e-2) =
	5	1.52e+0 (1.85e-2)	1.79e+0 (6.62e-2) +	1.58e+0 (3.09e-2) +	1.96e+0 (7.07e-2) +	1.79e+0 (6.13e-2) +	1.55e+0 (3.70e-2) =	1.54e+0 (3.36e-2) =
	8	3.41e+0 (7.23e-2)	4.41e+0 (4.28e-1) +	3.63e+0 (1.06e-1) +	5.20e+0 (2.30e-1) +	4.42e+0 (3.85e-1) +	3.49e+0 (9.38e-2) +	3.51e+0 (1.20e-1) +
WFG7	3	6.53e-1 (1.67e-2)	6.77e-1 (1.84e-2) +	6.82e-1 (1.41e-2) +	6.07e-1 (1.51e-2) -	6.49e-1 (2.28e-2) =	6.88e-1 (1.47e-2) +	6.86e-1 (1.15e-2) +
	5	1.46e+0 (6.76e-2)	1.79e+0 (9.25e-2) +	1.43e+0 (3.20e-2) =	1.81e+0 (9.38e-2) +	1.61e+0 (9.04e-2) +	1.53e+0 (7.29e-2) +	1.56e+0 (9.46e-2) +
	8	3.80e+0 (2.51e-1)	5.25e+0 (3.78e-1) +	3.82e+0 (2.25e-1) =	5.53e+0 (2.76e-1) +	4.27e+0 (3.57e-1) +	4.09e+0 (6.46e-1) =	4.32e+0 (4.47e-1) +
WFG8	3	7.17e-1 (2.37e-2)	8.28e-1 (1.82e-2) +	7.28e-1 (2.19e-2) +	7.70e-1 (2.83e-2) +	7.50e-1 (3.41e-2) +	7.06e-1 (2.71e-2) =	7.15e-1 (2.18e-2) =
	5	1.54e+0 (2.63e-2)	1.89e+0 (6.07e-2) +	1.58e+0 (3.26e-2) +	2.04e+0 (6.48e-2) +	1.86e+0 (8.41e-2) +	1.57e+0 (4.13e-2) +	1.56e+0 (3.49e-2) =
	8	3.65e+0 (9.08e-2)	5.04e+0 (3.05e-1) +	3.85e+0 (1.08e-1) +	5.41e+0 (1.85e-1) +	4.76e+0 (2.54e-1) +	3.70e+0 (1.09e-1) =	3.79e+0 (2.50e-1) +
WFG9	3	8.28e-1 (4.45e-2)	8.66e-1 (4.96e-2) +	8.74e-1 (2.72e-2) +	7.72e-1 (5.10e-2) -	8.69e-1 (4.99e-2) +	8.58e-1 (4.20e-2) =	8.60e-1 (4.16e-2) +
	5	1.77e+0 (7.10e-2)	1.97e+0 (1.06e-1) +	1.83e+0 (8.84e-2) +	1.89e+0 (1.12e-1) +	1.87e+0 (9.94e-2) +	1.87e+0 (9.38e-2) +	1.89e+0 (1.00e-1) +
	8	4.40e+0 (3.00e-1)	5.22e+0 (4.43e-1) +	4.71e+0 (4.32e-1) +	4.88e+0 (2.74e-1) +	4.58e+0 (4.91e-1) =	4.57e+0 (4.17e-1) =	4.87e+0 (4.80e-1) +
+/-/=		DTLZ	8/6/4	17/0/1	12/5/1	11/0/7	7/2/9	4/1/13
		WFG	24/1/2	22/0/5	21/6/0	18/1/8	10/1/16	11/0/16
		Total	32/7/6	39/0/6	33/11/1	29/1/15	17/3/25	15/1/29

rewritten as $F(\{X_o, X_m \setminus X_o\}) = F(\{F'(f_1), \dots, F'(f_{m-1}), F'(f_{m+1}), \dots, F'(f_M), X_m \setminus X_o\}) = F(\{f_1, \dots, f_{m-1}, f_{m+1}, \dots, f_M, X_m \setminus X_o\}) \rightarrow f_m$. Herein, we use a model cs_m to rewrite the approximated mapping from X_m to f_m , as:

$$cs_m = F(\{f_1, \dots, f_{m-1}, f_{m+1}, \dots, f_M, X_m \setminus X_o\}) \rightarrow f_m \quad (5)$$

The fusion of all cs_m ($m = 1, \dots, M$) consists of the CS. Fig. 4 shows the variables partition in CS. For each cs_m , only variables in $X_m \setminus X_o$ are directly involved in the training of cs_m . The intersection variable set X_o works as the shared variable subset to fulfill the mapping from f_n ($n = 1, \dots, M$ and $n \neq m$) to f_m .

When we compare CS with RS, we can find that every model rs_m in RS is independent of each other, while cs_m in CS parasitically interacts with other cs_n ($n = 1, \dots, M$, and $n \neq m$).

In other words, all models in CS are conjugate symbiosis since the output of each objective depends on the other objectives as inputs, and at the time participates in the output of the other objectives as inputs.

C. Surrogate Management

In this section, we illustrate the details of our proposed surrogate management method.

Algorithm 1 gives the pseudo-code of surrogate training. Once DB is input, the Relief-F weights between variables and each objective f_m are calculated according to the solutions in DB (line 2). Then a variable subset X_m is chosen (lines 3-7), and only these X_m instead of all variables in DB participate in the training of RS (lines 8 and 10). After that, we obtain the intersection of all variable subsets (line 11). Finally, the CS

is trained (lines 12-14). In our surrogate, the Kriging model is adopted for building RS and CS.

Algorithm 2 gives the method of using our well-trained surrogate to replace real FE during the optimization process. When an un-evaluated offspring solution \mathbf{x} is generated, it is first fed to the RS to get the predicted objective values (e.g., $rf_1(\mathbf{x})$ to $rf_M(\mathbf{x})$) and the prediction errors (e.g., $re_1(\mathbf{x})$ to $re_M(\mathbf{x})$) (lines 1-3). Then \mathbf{x} is fed to the CS to get another predicted value and error (e.g., $cf_m(\mathbf{x})$ and $ce_m(\mathbf{x})$) for each objective (line 5). Notice that the prediction values of \mathbf{x} from RS will participate as input in the prediction process by CS. After that, if $ce_m(\mathbf{x})$ is smaller than $re_m(\mathbf{x})$, then the m -th objective value of \mathbf{x} (e.g., $f_m(\mathbf{x})$) and its prediction error (i.e., $e_m(\mathbf{x})$) are provided by the model cs_m from CS; otherwise, it is estimated by the model rs_m from RS (lines 6-10).

IV. EXPERIMENTAL STUDIES

A. Experimental Settings

Theoretically, our proposed surrogate can be implemented into any SAMOEAs that use Kriging models to approximate the objective functions for expensive multiobjective optimization. Herein, we use it to replace the Kriging surrogate in K-RVEA [22], and the K-RVEA variant adopting our surrogate is denoted as CS-RVEA. Comparison experiments are taken with five state-of-the-art SAMOEAs (MOEA/D-EGO [8], GP-ARMOEA [9], HeE-MOEA [14], AB-SAEA [10], and K-RVEA) on DTLZ [23] and WFG [24] benchmark suites. In addition, to verify the effectiveness of CS, a CS-RVEA variant in which only the RS used (denoted as RS-RVEA) is also conducted.

All algorithms run 20 independent times on each test problem. The inverted generalized distance (IGD) [25][26] is adopted as the performance metric. Wilcoxon's rank-sum test at a 5% significance level is conducted between CS-RVEA and other compared algorithms. The symbols of '+', '-', and '=' represent that CS-RVEA is significantly better than, worse than, or similar to the corresponding compared algorithm, respectively. For each independent experiment, population size $N = 50$, variable dimension $D = 20$. The Latin hypercube sampling is adopted to initialize $11D - 1$ solutions as the initialized DB. The maximum number of FE is set as 339 (e.g., $11D - 1 + 120$). Except for these, other parameters of all compared algorithms are set according to their original papers for a fair comparison.

B. Experimental Results and Analysis

Table I shows the statistical IGD results between CS-RVEA and other compared SAMOEAs on DTLZ and WFG with $M = 3, 5, \text{ and } 8$, respectively. It can be seen that CS-RVEA performs worse than some SAMOEAs on only a few test problems, whereas outperforms them on most other problems. Specifically, CS-RVEA obtains significantly better IGD results than other compared SAMOEAs on 32, 39, 33, 29, 17, and 15 out of 45 instances, respectively. These results confirm the competitiveness of CS-RVEA in dealing with EMOPs.

In addition, from the comparison of the results in Table I, the efficiency of CS on WFG seems to be better than that on DTLZ. This is probably due to the differences in the way the two benchmark test suites are constructed. DTLZ used a simple addition form to define the objective functions, which makes the objectives themselves lack sufficient symbiosis. This makes it difficult for CS to play a role in these kinds of

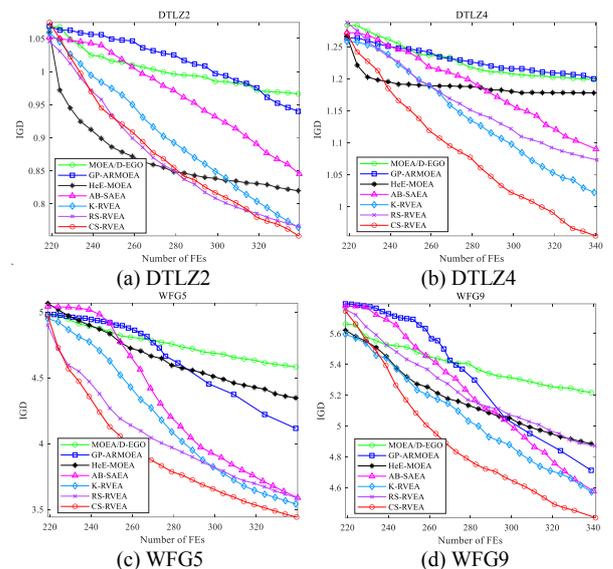


Fig. 5. The mean IGD curves of SAMOEAs on four test problems ($M = 8$).

TABLE II
AVERAGE TRAINING TIME (S) FOR A SINGLE SURROGATE ON DTLZ2

M	KS	RS	CS
3	3.20160	1.77366	0.95435
5	5.61238	3.00935	3.29620
8	8.02504	3.65827	7.88033

problems. On the contrary, the structure of WFG is relatively more complex, and the objectives have stronger symbiosis, making the importance of CS more prominent in these problems. This result shows that the use of CS also depends on the property of EMOPs. CS is more suitable for problems where there is a certain connection between objectives, even if such a connection cannot be mathematically determined.

To further explore the acceleration effect of CS in convergence, Fig. 5 plots the mean IGD curves of SAMOEAs over the number of real FEs on different problems. It can be seen that the IGD curve by CS-RVEA declines significantly faster than that by the other compared SAMOEAs. Here we focus on comparing the IGD curves among K-RVEA, RS-RVEA, and CS-RVEA, and we can find that the curve of RS-RVEA is worse than that of K-RVEA in some cases. These results further confirm our previous discussion that a single RS will cause a deterioration in performance, although it can reduce training costs. However, with the help of CS, CS-RVEA can not only avoid this performance deterioration caused by RS but also improve the convergence speed.

C. Runtime Cost Analysis

In this section, we explore the difference in runtime cost between the three different surrogates. The experiments are implemented in Matlab on an identical computer with Inter(R) Core (TM) i5-11400 2.60 GHz and 16GB of RAM. Table II shows the average training time for a single surrogate on DTLZ2 with different objectives. It can be seen that the training time of RS and CS is less than that of KS, which indicates that they also have an advantage over KS in time cost. Although the simultaneous use of RS and CS will increase the training time, it is possible to reduce it through parallel training techniques.

V. CONCLUSION

The existing Kriging surrogate in SAMOEAs works as the replacement for real expensive fitness evaluation by

approximating the mapping from variables to each objective but neglects the possible relationship among different objectives. In this paper, we for the first time explore the possibility of objectives-to-objective mapping and propose the CS to approximate such mapping for expensive multiobjective optimization. In addition, the collaboration between this CS and RS can also alleviate the prediction performance deterioration caused by dimension reduction, so as to improve the convergence speed without high training cost.

Our proposed CS can be implemented into any SAMOEA in which the Kriging model is adopted to approximate each objective function for solving EMOPs, especially for those EMOPs when there is a potential connection between objectives. In the experiment, a CS-assisted SAMOEA called CS-RVEA is proposed, and the comparison results between several state-of-the-art SAMOEAs confirm its effectiveness and competitiveness in dealing with EMOPs. In our future work, we will propose several strategies to enhance the performance of CS and extend it to solve real-world expensive multiobjective optimization applications[27]-[30].

REFERENCES

- [1] Z. -H. Zhan, L. Shi, K. C. Tan, and J. Zhang, "A survey on evolutionary computation for complex continuous optimization," *Artif. Intell. Rev.*, vol. 55, pp. 59–110, 2022.
- [2] Z. -H. Zhan, J. -Y. Li, S. Kwong, and J. Zhang, "Learning-aided evolution for optimization," *IEEE Trans. Evol. Comput.*, 2022, DOI: 10.1109/TEVC.2022.3232776.
- [3] X. -F. Liu, Z. -H. Zhan, Y. Gao, J. Zhang, S. Kwong, and J. Zhang, "Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 587–602, 2019.
- [4] S. -C. Liu, Z. -H. Zhan, K. C. Tan, and J. Zhang, "A multi-objective framework for many-objective optimization," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13654–13668, 2022.
- [5] Q. -T. Yang, Z. -H. Zhan, S. Kwong, and J. Zhang, "Multiple populations for multiple objectives framework with bias sorting for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 5, pp. 1340–1354, 2023.
- [6] J. -Y. Li, Z. -H. Zhan, and J. Zhang, "Evolutionary computation for expensive optimization: A survey," *Mach. Intell. Res.*, vol. 19, no. 1, pp. 3–23, 2022.
- [7] 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, 2019.
- [8] Q. Zhang, W. Liu, and E. Tsang, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, 2010.
- [9] D. Guo, X. Wang, K. Gao, Y. Jin, J. Ding, and T. Chai, "Evolutionary optimization of high-dimensional multiobjective and many-objective expensive problems assisted by a dropout neural network," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 52, no. 4, pp. 2084–2097, 2021.
- [10] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, "An adaptive Bayesian approach to surrogate-assisted evolutionary multi-objective optimization," *Info. Sci.*, vol. 519, pp. 317–331, 2020.
- [11] Q. -T. Yang, Z. -H. Zhan, Y. Li, and J. Zhang, "Social learning particle swarm optimization with two-surrogate collaboration for offline data-driven multiobjective optimization", in *Proc. ACM Genet. Evol. Comput. Conf.*, 2022, pp. 49–57.
- [12] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, and N. De Freitas, "Taking the human out of loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, 2015.
- [13] T. Chugh, Y. Jin, K. Miettinen, and J. Hakanen, "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, 2018.
- [14] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1012–1025, 2018.
- [15] M. T. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single-and multiobjective evolutionary optimization assisted by Gaussian random field meta-models," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 421–439, 2006.
- [16] Q. Lin, X. Wu, L. Ma, J. Li, M. Gong, and C. A. C. Coello, "An ensemble surrogate-based framework for expensive multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 4, pp. 631–645, 2022.
- [17] J. Lin, C. He, and R. Cheng, "Dimension dropout for evolutionary high-dimensional expensive multiobjective optimization," in *Proc. Evol. Multi-Criterion Optim.*, 2021, pp. 567–579.
- [18] S. Qin, C. Sun, Q. Liu, and Y. Jin, "A performance indicator based infill criterion for expensive multi-/many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 4, pp. 1085–1099, 2023.
- [19] J. -Q. Yang *et al.*, "Bi-directional feature fixation-based particle swarm optimization for large-scale feature selection," *IEEE Trans. Big Data*, vol. 9, no. 3, pp. 1004–1017, 2023.
- [20] M. R. Šikonja and I. Kononenko, "Theoretical and empirical analysis of reliefF and RReliefF," *Mach. Learn.*, vol. 53, nos. 1–2, pp. 23–69, 2003.
- [21] K. Chen, B. Xue, M. Zhang, and F. Zhou, "An evolutionary multitasking-based feature selection method for high-dimensional classification," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 7172–7186, 2022.
- [22] T. Chugh, Y. Jin, K. Miettinen, and J. Hakanen, "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, 2018.
- [23] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Proc. Evol. Multi-Criterion Optim.*, 2005, pp. 105–145.
- [24] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, 2006.
- [25] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, 2003.
- [26] J. -Y. Li, Z. -H. Zhan, Y. Li, and J. Zhang, "Multiple tasks for multiple objectives: A new multiobjective optimization method via multitask optimization," *IEEE Trans. Evol. Comput.*, 2023, DOI:10.1109/TEVC.2023.3294307.
- [27] Z. -H. Zhan, J. -Y. Li, and J. Zhang, "Evolutionary deep learning: A survey," *Neurocomputing*, vol. 483, pp. 42–58, 2022.
- [28] Y. Jiang, Z. -H. Zhan, K. C. Tan, and J. Zhang, "Knowledge learning for evolutionary computation," *IEEE Trans. Evol. Comput.*, 2023, DOI: 10.1109/TEVC.2023.3278132.
- [29] J. -Y. Li, Z. -H. Zhan, J. Xu, S. Kwong, and J. Zhang, "Surrogate-assisted hybrid-model estimation of distribution algorithm for mixed-variable hyperparameter optimization in convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 5, pp. 2338–2352, 2023.
- [30] Z. -G. Chen, Z. -H. Zhan, S. Kwong, and J. Zhang, "Evolutionary computation for intelligent transportation in smart cities: A survey," *IEEE Comput. Intell. Mag.*, vol. 17, no. 2, pp. 83–102, 2022.