2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Effects of Initialization Methods on the Performance of Surrogate-based Multiobjective Evolutionary Algorithms

Jinyuan Zhang, Hisao Ishibuchi, Linjun He, and Yang Nan
Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation,
Department of Computer Science and Engineering,
Southern University of Science and Technology, Shenzhen 518055, China.
zhangjy@sustech.edu.cn, hisao@sustech.edu.cn, this.helj@gmail.com, nany@mail.sustech.edu.cn

***Abstract*—Initialization plays a crucial role in surrogate-based multiobjective evolutionary algorithms (MOEAs) when tackling computationally expensive multiobjective optimization problems. During the initialization process, solutions are generated to train surrogate models. Consequently, the accuracy of these surrogate models depends on the quality of the initial solutions, which in turn directly impacts the performance of surrogate-based MOEAs. Despite the widespread use of Latin hypercube sampling as an initialization method in surrogate-based MOEAs, there is a lack of comprehensive research examining the effectiveness of different initialization methods. Additionally, the impact of the number of initial solutions on the performance of surrogate-based MOEAs remains largely unexplored. This paper aims to bridge these research gaps by comparing the usefulness of two commonly employed initialization methods (i.e., random sampling and Latin hypercube sampling) in surrogate-based MOEAs. Furthermore, it investigates how varying the number of initial solutions influences the performance of surrogate-based MOEAs.**

***Index Terms*—Multiobjective evolutionary algorithms, initialization, surrogate models**

I. INTRODUCTION

Many real-world optimization problems involve multiple conflicting objectives that cannot be optimized simultaneously, known as multiobjective optimization problems (MOPs) [1], [2]. To solve MOPs, evolutionary algorithms have become a popular method, with various multiobjective evolutionary algorithms (MOEAs) being proposed in recent decades. These algorithms can be divided into three categories: dominance-based MOEAs [3], [4], indicator-based MOEAs [5], [6], [7], and decomposition-based MOEAs [8], [9]. These MOEAs usually require a large number of function evaluations to solve MOPs [7], [10]. However, evaluating the quality of solutions in real-world MOPs can be computationally expensive or economically costly.

This work was supported by National Natural Science Foundation of China (Grant No. 62106099, 62250710163, 62250710682), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2023A1515011380), Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No. 2017ZT07X386), The Stable Support Plan Program of Shenzhen Natural Science Fund (Grant No. 20200925174447003), Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531). (Corresponding author: H. Ishibuchi.)

To address this issue, surrogate-based MOEAs have been proposed as a more efficient method for solving expensive MOPs [11], [12], [13]. These algorithms construct computationally cheap surrogate models to replace the original objective functions or fitness functions for evaluating the quality of solutions. Surrogate-based MOEAs can be further classified into two categories based on the types of employed surrogate models: regression-based MOEAs and classification-based MOEAs.

In regression-based MOEAs [11], [13], [14], each objective function of an MOP is usually approximated by a separate regression model. The approximated objective functions are then used instead of the original expensive objective functions for evaluation. Knowles [15] used the efficient global optimization (EGO) algorithm [16] and employed a Gaussian process model to approximate MOPs' landscape. Meanwhile, Song et al. [14] combined the Kriging model with a two-archive evolutionary algorithm. A Kriging model is used to approximate each objective function of the MOP. In most cases, the number of constructed regression models is the same as the number of objective functions.

Classification-based MOEAs [31], [32], [33] often build classifiers to learn the relation between solutions [31], e.g., dominance relation. Loshchilov et al. [31] proposed a surrogate model that merges a classifier and a regression model to learn the dominance relation between a new solution and the existing non-dominated solution set. Zhang et al. [33], [34] used a classifier to pre-select promising offspring solutions by learning the dominance relation between solutions. Lin et al. [35] proposed to use a classifier for pre-selecting promising offspring solutions in MOEA/D, where an SVM model was utilized to predict the relation between solutions. Recently, Yuan et al. [27] combined two feedforward neural network models with θ-DEA [36] for solving expensive MOPs. One model was used to predict the Pareto dominance relation between solutions and the other model was built to predict the θ-dominance relation among solutions. Hao et al. [28] proposed a classifier-based relation prediction model for MOPs, where a neural network model was constructed to learn the dominance relation between a pair of solutions. In most

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

TABLE I
TYPICAL SURROGATE-BASED MOEAS AND THEIR INITIALIZATION METHODS.

Algorithm	Year	Test Problem	M	d	Initialization method	NI	$MaxFEs$
ParEGO [15]	2005	KNO1, OKA1, VLMOP2 OKA2 VLMOP3 DTLZ1a DTLZ2a, DTLZ4a, DTLZ7a	2 2 3 2 3	2 3 2 6 8	LHS	$11d-1$	250
SMS-EGO [17]	2008	OKA2, R_ZDT4 R_ZDT1 R_DTLZ2	2 2 $3, 5$	3 6 $6, 6$	LHS	$11d-1$	130
MOEA/D-EGO [18]	2010	KNO1, VLMOP2 ZDT1-4, 6, LZ08-F1-4 DTLZ2	2 2 3	2 8 6	LHS	$11d-1$	300 ($M=3$) 200 ($M=2$)
K-RVEA [19]	2018	DTLZ1-7	$3, 4, 6, 8, 10$	10	LHS	$11d-1$	300
HSMEA [20]	2019	DTLZ1-7 WFG1-9	$3, 4, 6, 8, 10$ $3, 4, 6, 8, 10$	10 $10, 10, 9, 9, 11$	LHS	$11d-1$	300
CSEA [21]	2019	DTLZ1-7 WFG1-9 MaF1-5	$3, 4, 6, 8, 10$ $3, 4, 6, 8, 10$ $3, 4, 6, 8, 10$	10 $10, 10, 9, 9, 11$ -	LHS	$11d-1$	300
HeE-MOEA [22]	2019	DTLZ1-7, WFG1-9	3	$10, 20, 40, 80$	LHS	$11d-1$	$11d+119$
EIR2 [23]	2020	ZDT1, 2 ZDT3 DTLZ2, 5, 7	2 2 3	5	LHS	$MaxFEs/3$	$90, 120, 210$
AB-MOEA [24]	2020	DTLZ1 DTLZ2-6 DTLZ7 UF1-7	3 3 3 2	7 12 22 -	LHS	$11d-1$	300
KTA2 [14]	2021	DTLZ1-7 WFG1-9	$3, 4, 6, 8, 10$ $3, 4, 6, 8, 10$	10 $10, 10, 9, 9, 11$	LHS	$11d-1$	300
PB-NSGAIII [25]	2021	DTLZ1-7	3, 4, 6, 8, 10	10	LHS	$11d-1$	300
EDN-ARMOEA [26]	2022	DTLZ1-7	3 $5, 10$	$20, 40, 60$ 40	LHS	$11d-1$	$11d+119$
θ-DEA-DP [27]	2022	ZDT1-4 DTLZ1 DTLZ2, 4, 7 WFG6, 7	2 $2, 5, 8$ $3, 5, 8$ $3, 5, 8$	10 $6, 10, 10$ $8, 10, 10$ 10	LHS	$11d-1$	250 ($M=2,3$) 300 ($M=5$) 400 ($M=8$)
REMO [28]	2022	DTLZ1-7	$3, 4, 6, 8, 10$ 3 10	10 $30, 50$ $30, 50$	LHS	$11d-1$ 100 100	300
ESF-RVEA [29]	2022	DTLZ1-7, WFG1-9 UF1-9 MaF1-7	3 2 3	$10, 20, 30$	LHS	$11d-1$	$11d+120$
MCEA/D [30]	2022	DTLZ1-7, WFG1-9	$3, 7, 11$	$50, 100, 150$	LHS	100	300

of the aforementioned algorithms, the number of classifiers constructed is fewer than in regression-based algorithms.

Initialization is an important component of surrogate-based MOEAs. The solutions generated during the initialization process, which are evaluated by expensive objective functions, serve as the initial training data for surrogate model building. As a result, these solutions play a crucial role in the accuracy of surrogate models, which in turn affect the performance of the surrogate-based MOEA.

Despite the popularity of the Latin hypercube sampling (LHS) [37] method in surrogate-based MOEAs, the effectiveness of the LHS method has rarely been examined. Different numbers of initial solutions are often used in experiments. Table I summarizes the studies on surrogate-based MOEAs. It includes the proposed surrogate-based MOEAs, the test problems used in the experiments, the number of objectives (M), the initialization method in the proposed algorithm, the setting of the number of initial solutions (NI), and the setting of the maximum number of function evaluations ($MaxFEs$). From the table, we have the following observations.

- Surrogate-based MOEAs in the studies prefer to use the LHS method for initialization.
- The number of initial solutions is generally set to $11d-1$, where d is the number of decision variables.
- In most studies, the number of decision variables is set to a small number, typically less than 10, and the maximum number of function evaluations is set to 300.
- When the number of decision variables is larger than 10, some algorithms fix the number of initial solutions (e.g., 100) and keep the maximum number of function evaluations constant. Other algorithms continue to initialize $11d-1$ solutions but increase the maximum number of

function evaluations as d increases (e.g., $11d + 119$).

- Most of these algorithms are usually examined on DTLZ [38] and WFG [39] test suites.

This paper aims to answer the following two questions:

1) In the literature, researchers in expensive multiobjective optimization have shown a preference for using the LHS method over the random initialization method in surrogate-based MOEAs. However, is the LHS method really better than the random method in terms of usefulness for choosing initial solutions? Is there any other good initialization mechanism?
2) Different algorithms generate initial solutions in varying quantities, but many algorithms produce $11d - 1$ solutions. How does the setting of the initial solution quantity affect the performance of surrogate-based MOEAs? Is there any other appropriate number of initial solutions?

The remainder of this paper is as follows. Section II presents the preliminaries of this paper. The employed surrogate-based MOEAs and examined initialization methods are presented. Section III shows the numerical results and answers the two research questions. The paper is concluded in Section IV.

II. Preliminaries

A. Investigated Surrogate-based MOEAs

To examine the usefulness of the LHS method in selecting initial solutions and determine the impact of the number of initial solutions on the performance of surrogate-based MOEAs, six widely used surrogate-based MOEAs are investigated in this paper. These algorithms are divided into two categories based on the type of employed surrogate models: (1) regression-based MOEAs and (2) classification-based MOEAs.

1) Regression-based MOEAs: Regression-based MOEAs are a type of surrogate-based MOEAs that use regression models to evaluate the quality of solutions. By using regression models to learn the landscape of objective functions, regression-based MOEAs can effectively reduce the required number of function evaluations and thus reduce the computational cost of solving expensive MOPs. Three regression-based MOEAs are briefly introduced as follows.

1) MOEA/D-EGO [18]: This algorithm combines the EGO algorithm with MOEA/D [9] algorithm. It solves MOPs by decomposing them into several single-objective optimization subproblems. Then, a Gaussian model is used to learn the landscape of each subproblem. The quality of offspring solutions is evaluated using the corresponding Gaussian models. The expected improvements of subproblems are then used to select candidate solutions for expensive function evaluations.
2) K-RVEA [19]: This algorithm combines a Kriging model with the reference vector guided evolutionary algorithm (RVEA) [40]. The Kriging model is used to approximate each objective function of MOPs. The algorithm employs uncertainty information from the Kriging model to effectively balance both exploitation and exploration of the solution space.
3) EDN-ARMOEA [26]: This algorithm applies an efficient dropout neural network (EDN) model to the indicator-based MOEA with reference point adaptation (AR-MOEA) [7] algorithm. The EDN model is used to estimate approximate objective values and confidence levels. A model management method based on two sets of reference vectors is used to select candidate solutions for expensive function evaluations.

2) Classification-based MOEAs: Classification-based MOEAs are the other type of surrogate-based MOEAs that use classification models to evaluate the quality of solutions. By using classifiers to learn the dominance or crowding relation between solutions, classification-based MOEAs can balance the trade-off between convergence and diversity of solutions. Three classification-based MOEAs are briefly introduced as follows.

1) CSEA [21]: This algorithm uses a feedforward neural network model to learn the dominance relation between a new solution and reference solutions. The uncertainty information from the model is used to select candidate solutions for expensive function evaluations.
2) MCEA/D [30]: This algorithm applies multiple support vector machine models to MOEA/D algorithm. It decomposes an MOP into a set of single-objective optimization subproblems. Each subproblem is solved using a separate support vector machine model. The distance between the solution and the decision boundary is used to select solutions for expensive function evaluations.
3) DFC-MOEA [41]: This algorithm is based on a dual fuzzy-classifier-based surrogate model. Two fuzzy classifiers are used to evaluate the quality of unevaluated solutions. One fuzzy classifier is used to learn the Pareto dominance relation among solutions, while the other is used to learn the crowdedness of solutions. This way, both convergence and diversity of solutions can be maintained. As a general algorithm framework, any MOEAs can be integrated into DFC-MOEA. This paper conducts experiments by using DFC-SMS-EMOA.

B. Initialization Methods in Surrogate-Based MOEAs

Initialization is a crucial component of MOEAs. Various initialization methods have been used in MOEAs [42]. The random initialization method, which randomly samples a set of solutions as initial solutions, is widely used in conventional MOEAs. Recently, the Latin hypercube sampling (LHS) method has gained popularity for initialization in surrogate-based MOEAs. In the LHS method, a grid of solution positions is defined as a Latin square if there is only one solution in each row and column. The Latin hypercube extends this concept to multiple dimensions, where each solution is the only one in each axis-aligned hyperplane. To sample N solutions, the LHS method first divides the range of each dimension into N uniform intervals. Then, N solutions are sampled to satisfy the Latin hypercube conditions. This makes the LHS method as a space-filling design method that can effectively ensure the accuracy of the constructed surrogate models. Typically, the number of initial solutions sampled using the LHS method is set to $11d - 1$, where d is the number of decision variables. This is based on the recommendation of Jones et al. [16], who proposed that about $10d$ solutions are needed in the initial design. But with a convenient finite-decimal value for solution spacing, they suggested deviating slightly from the "$10d$" rule, for example, using 21 solutions for two dimensions, 33 for three dimensions, and 65 for six dimensions. This $11d - 1$ setting has been widely applied to many surrogate-based MOEAs [15], [18], [19], [26].

III. Numerical Results

This section examines the effectiveness of the LHS method in selecting initial solutions and explores the impact of the number of initial solutions on surrogate-based MOEAs. Six surrogate-based MOEAs, as mentioned in Section 2.1, are examined on two commonly used test suites. The experimental settings are presented in Section 3.1, followed by an examination of the usefulness of the LHS method in generating initial solutions in Section 3.2. Finally, Section 3.3 explores the effect of the number of initial solutions.

A. Experimental Settings

This section outlines the experimental settings for examining the usefulness of the LHS method and the effect of the number of initial solutions on surrogate-based MOEAs. Six surrogate-based MOEAs and two widely used test suites are selected for experiments. The following details outline the specifications for the algorithms, test problems, population size, number of executions, number of decision variables, termination condition, performance metrics, and parameter settings used in experiments.

1) Algorithms studied: Six surrogate-based MOEAs, including three regression-based algorithms (MOEA/D-EGO [18], K-RVEA [19], and EDN-ARMOEA [26]), and three classification-based algorithms (CSEA [21], MCEA/D [30], and DFC-MO-EA [41]).
2) Test problems: The DTLZ1–7 [38] and WFG1–9 [39] test problems with three objectives (i.e., $M = 3$) are considered.
3) Population size: $N = 50$ is used for EDN-ARMOEA, CSEA, and DFC-MOEA, and $N = 45$ is used for K-RVEA and MCEA/D.
4) Number of executions: Each algorithm is run 21 times independently on each test problem.
5) The number of decision variables and termination condition: Experiments are conducted with three values of decision variables, $d = 10$, 30, and 50. The number of maximum function evaluations is 300 for $d = 10$, 500 for $d = 30$, and 800 for $d = 50$.
6) Performance metrics: The performance of each algorithm is evaluated using the inverted generational distance (IGD) metric [43], calculated using reference points from PlatEMO [44].
7) Platform and parameter settings: All experiments are performed on the PlatEMO platform [44] with default parameter settings.

B. The Usefulness of the LHS Method

To examine the usefulness of the LHS method in surrogate-based MOEAs, this section compares the performance of using the LHS method and the random method for initialization. MOEA/D-EGO, K-RVEA, EDN-ARMOEA, CSEA, MCEA/D, and DFC-SMS-EMOA are used in the experiments. Each algorithm with two initialization methods are compared. The resulted algorithms are MOEA/D-EGO-LHS, MOEA/D-EGO-random, K-RVEA-LHS, K-RVEA-random, EDN-AR-MOEA-LHS, EDN-ARMOEA-random, CSEA-LHS, CSEA-random, MCEA/D-LHS, MCEA/D-random, DFC-SMS-EMOA-LHS and DFC-SMS-EMOA-random. The number of decision variable is $d = 10$. The maximum number of function evaluations is 300.

Table II shows the mean and standard deviation (std) IGD values obtained by the six pairs of compared algorithms when the number of initial solutions is $NI = 11d - 1$ after 300 function evaluations on the DTLZ1–7 and WFG1–9 test problems. The Wilcoxon rank-sum test at the 5% significance level is used to evaluate the statistical difference between the LHS-based algorithm and the random-based algorithm. "$+, -, \sim$" indicates that the results obtained by the LHS-based algorithm are better than, worse than, or similar to those obtained by the random-based algorithm, respectively. Statistically better result is shaded. Table II shows that algorithms with the LHS method and random method for initialization have almost similar performance on these test problems when the number of initial solutions is $NI = 11d - 1$.

Fig. S1 in the supplementary file[1] plots the mean IGD values versus the number of function evaluations obtained by K-RVEA-LHS and K-RVEA-random on DTLZ1–7 test problems. The results in Fig. S1 also show that K-RVEA-LHS and K-RVEA-random have similar performance on most test problems.

To further examine the effectiveness of the two compared initialization methods. The six pairs of compared algorithms are examined when the number of initial solutions is the same as the population size ($NI = N$). Table S1 in the supplementary file shows the mean and standard deviation (std) values obtained by the six pairs of compared algorithms after 300 function evaluations on the 16 test problems. The results also show that the performance of the compared algorithms are almost same on these test problems.

The results in Table II and Table S1 show that the surrogate-based MOEAs use the LHS method and the random method for initialization have similar performance on the DTLZ and WFG test suites. From these results, we can conclude that we do not have to use LHS for initialization since (almost) the same results can be obtained from random initialization.

To investigate whether there are any other good initialization mechanisms for surrogate-based MOEAs, we examine

[1] https://github.com/HisaoLabSUSTC/SSCI-2023

TABLE II

THE $mean(std)$ IGD VALUES OF THE COMPARED ALGORITHMS WHEN THE NUMBER OF INITIAL SOLUTIONS IS $11d - 1$.

Problem	M	d	MOEA/D-EGO-LHS	MOEA/D-EGO-random	K-RVEA-LHS	K-RVEA-random	EDN-ARMOEA-LHS	EDN-ARMOEA-random
DTLZ1	3	10	1.7170e+2 (3.95e+1) $\approx$	1.5931e+2 (3.29e+1)	1.7478e+2 (3.90e+1) $\approx$	1.6499e+2 (3.84e+1)	2.0036e+2 (3.78e+1) $\approx$	2.1339e+2 (4.45e+1)
DTLZ2	3	10	3.3145e-1 (2.69e-2) $\approx$	3.2168e-1 (2.42e-2)	1.5677e-1 (2.71e-2) $\approx$	1.4351e-1 (3.47e-2)	3.1030e-1 (2.22e-2) $\approx$	2.9986e-1 (3.04e-2)
DTLZ3	3	10	1.9679e+2 (2.12e+1) $\approx$	2.0740e+2 (2.88e+1)	2.1631e+2 (3.68e+1) $\approx$	2.1780e+2 (4.22e+1)	3.0266e+2 (5.87e+1) $\approx$	3.0419e+2 (5.47e+1)
DTLZ4	3	10	6.2757e-1 (6.38e-2) $\approx$	6.4940e-1 (9.19e-2)	4.0708e-1 (9.31e-2) $\approx$	3.7859e-1 (8.11e-2)	2.3313e-1 (3.75e-2) $\approx$	2.6871e-1 (8.55e-2)
DTLZ5	3	10	3.3433e-1 (4.86e-2) $\approx$	3.2335e-1 (4.37e-2)	9.7996e-2 (2.56e-2) $\approx$	9.7750e-2 (2.65e-2)	1.9190e-1 (3.11e-2) $\approx$	1.9116e-1 (4.02e-2)
DTLZ6	3	10	2.4845e+0 (7.26e-1) $\approx$	2.1658e+0 (7.15e-1)	3.0742e+0 (4.81e-1) $\approx$	3.1184e+0 (6.08e-1)	6.2788e+0 (3.31e-1) $\approx$	6.3453e+0 (3.29e-1)
DTLZ7	3	10	1.6368e-1 (5.51e-2) $\approx$	1.9386e-1 (9.77e-2)	7.9201e-2 (1.00e-2) $\approx$	8.5212e-2 (1.34e-2)	6.4129e-1 (1.64e-1) $\approx$	6.6864e-1 (1.95e-1)
WFG1	3	10	8.6936e-1 (6.67e-2) $\approx$	8.7119e-1 (5.90e-2)	6.6011e-1 (4.80e-2) $\approx$	6.7678e-1 (4.53e-2)	7.6512e-1 (2.76e-2) $\approx$	7.6639e-1 (1.95e-2)
WFG2	3	10	2.2665e-1 (1.30e-2) $\approx$	2.1934e-1 (1.68e-2)	1.3068e-1 (2.68e-2) $\approx$	1.4136e-1 (3.02e-2)	2.2124e-1 (1.24e-2) $\approx$	2.1805e-1 (1.87e-2)
WFG3	3	10	2.9572e-1 (1.98e-2) $\approx$	2.9843e-1 (1.85e-2)	1.8915e-1 (3.83e-2) $\approx$	1.9915e-1 (3.66e-2)	3.1168e-1 (2.05e-2) $\approx$	3.0485e-1 (1.95e-2)
WFG4	3	10	1.5440e-1 (3.97e-3) $\approx$	1.5287e-1 (9.92e-3)	1.4366e-1 (6.29e-3) $\approx$	1.4180e-1 (6.38e-3)	1.4452e-1 (4.33e-3) $\approx$	1.4623e-1 (6.09e-3)
WFG5	3	10	1.5608e-1 (8.63e-3) $\approx$	1.5679e-1 (8.24e-3)	9.5690e-2 (9.37e-3) $\approx$	9.4900e-2 (9.47e-3)	1.7504e-1 (1.01e-2) $\approx$	1.7992e-1 (9.77e-3)
WFG6	3	10	2.1414e-1 (8.52e-3) $\approx$	2.1396e-1 (1.29e-2)	2.1628e-1 (7.27e-3) $\approx$	2.1201e-1 (1.27e-2)	2.4417e-1 (8.45e-3) $\approx$	2.4216e-1 (7.77e-3)
WFG7	3	10	1.8979e-1 (5.23e-3) $\approx$	1.8731e-1 (6.38e-3)	1.8453e-1 (6.39e-3) $\approx$	1.8150e-1 (8.88e-3)	1.9119e-1 (5.43e-3) $\approx$	1.9114e-1 (4.68e-3)
WFG8	3	10	2.5283e-1 (1.01e-2) $\approx$	2.4856e-1 (1.08e-2)	2.1067e-1 (1.05e-2) $\approx$	2.1019e-1 (7.52e-3)	2.3587e-1 (6.44e-3) $\approx$	2.4023e-1 (8.81e-3)
WFG9	3	10	2.2651e-1 (1.41e-2) $\approx$	2.2583e-1 (1.70e-2)	2.0105e-1 (1.96e-2) $\approx$	1.9741e-1 (2.11e-2)	2.1683e-1 (2.26e-2) $\approx$	2.2681e-1 (1.93e-2)
$+/-/\approx$			0/0/16		0/0/16		0/0/16	
Problem	M	D	CSEA-LHS	CSEA-random	MCEA/D-LHS	MCEA/D-random	DFC-SMS-EMOA-LHS	DFC-SMS-EMOA-random
DTLZ1	3	10	1.1134e+2 (3.76e+1) $\approx$	1.1829e+2 (2.33e+1)	1.3901e+2 (4.37e+1) $\approx$	1.2230e+2 (4.21e+1)	1.8054e+2 (3.74e+1) $\approx$	1.5834e+2 (4.72e+1)
DTLZ2	3	10	2.2791e-1 (3.15e-2) $\approx$	2.2881e-1 (3.42e-2)	2.1922e-1 (3.12e-2) $\approx$	2.2239e-1 (3.79e-2)	2.3882e-1 (2.55e-2) $\approx$	2.3772e-1 (2.27e-2)
DTLZ3	3	10	1.4681e+2 (3.92e+1) $\approx$	1.4775e+2 (4.36e+1)	1.5813e+2 (2.50e+1) $\approx$	1.4043e+2 (5.51e+1)	1.9971e+2 (4.87e+1) $+$	2.3654e+2 (5.45e+1)
DTLZ4	3	10	3.8930e-1 (1.21e-1) $\approx$	4.5935e-1 (1.66e-1)	7.0096e-1 (1.64e-1) $\approx$	6.3977e-1 (1.76e-1)	4.3593e-1 (1.21e-1) $+$	5.6660e-1 (1.64e-1)
DTLZ5	3	10	1.5189e-1 (2.86e-2) $\approx$	1.5331e-1 (3.81e-2)	9.9108e-2 (2.51e-2) $\approx$	9.8916e-2 (2.57e-2)	1.7866e-1 (4.84e-2) $\approx$	2.0418e-1 (4.18e-2)
DTLZ6	3	10	6.0482e+0 (6.11e-1) $\approx$	6.0515e+0 (4.94e-1)	2.9375e+0 (9.67e-1) $\approx$	2.9853e+0 (7.34e-1)	6.5606e+0 (2.92e-1) $\approx$	6.4608e+0 (3.98e-1)
DTLZ7	3	10	7.2587e-1 (2.17e-1) $\approx$	7.4885e-1 (2.96e-1)	9.4096e-1 (3.30e-1) $\approx$	9.5360e-1 (3.02e-1)	1.2911e+0 (2.12e-1) $\approx$	1.2436e+0 (2.04e-1)
WFG1	3	10	6.6952e-1 (3.52e-2) $\approx$	6.6501e-1 (5.48e-2)	8.3935e-1 (2.14e-2) $\approx$	8.4161e-1 (2.46e-2)	8.3333e-1 (5.04e-2) $\approx$	8.3473e-1 (5.68e-2)
WFG2	3	10	1.7167e-1 (1.88e-2) $+$	1.8469e-1 (2.02e-2)	2.0961e-1 (2.72e-2) $\approx$	1.9997e-1 (2.15e-2)	1.9155e-1 (1.99e-2) $\approx$	1.9249e-1 (2.27e-2)
WFG3	3	10	2.5575e-1 (1.92e-2) $\approx$	2.4738e-1 (2.01e-2)	2.2948e-1 (3.03e-2) $\approx$	2.3914e-1 (2.89e-2)	2.4593e-1 (2.67e-2) $\approx$	2.4338e-1 (3.00e-2)
WFG4	3	10	1.3339e-1 (7.96e-3) $\approx$	1.3246e-1 (1.04e-2)	1.4839e-1 (1.03e-2) $\approx$	1.4930e-1 (1.09e-2)	1.3390e-1 (1.04e-2) $\approx$	1.3294e-1 (8.15e-3)
WFG5	3	10	1.3866e-1 (1.13e-2) $\approx$	1.3400e-1 (1.14e-2)	1.4212e-1 (1.60e-2) $\approx$	1.4835e-1 (1.54e-2)	1.8177e-1 (9.40e-3) $\approx$	1.8431e-1 (1.19e-2)
WFG6	3	10	2.0191e-1 (1.18e-2) $+$	2.1245e-1 (1.44e-2)	2.0560e-1 (1.15e-2) $\approx$	2.0809e-1 (1.27e-2)	2.1177e-1 (1.55e-2) $\approx$	2.1730e-1 (1.20e-2)
WFG7	3	10	1.6518e-1 (9.82e-3) $\approx$	1.6605e-1 (9.44e-3)	1.5818e-1 (8.95e-3) $\approx$	1.6149e-1 (1.05e-2)	1.5650e-1 (6.98e-3) $\approx$	1.6115e-1 (9.75e-3)
WFG8	3	10	2.3651e-1 (1.03e-2) $\approx$	2.3565e-1 (1.32e-2)	2.5020e-1 (1.55e-2) $-$	2.3917e-1 (1.62e-2)	2.3209e-1 (9.28e-3) $-$	2.2579e-1 (1.07e-2)
WFG9	3	10	1.9332e-1 (3.11e-2) $\approx$	1.9751e-1 (2.49e-2)	1.7943e-1 (1.81e-2) $\approx$	1.7493e-1 (1.28e-2)	1.8603e-1 (1.75e-2) $\approx$	1.8096e-1 (1.63e-2)
$+/-/\approx$			2/0/14		0/1/15		2/1/13	

three other initialization mechanisms. The improved Latin hypercube sampling (LHSmax) method [45], the Sobol sequence (Sobol) [46] method, and the opposition-based learning (OBL) [47] are used for examination. Detailed explanations of these three methods are shown in the supplementary file. The three methods and the random method are applied to MOEA/D-EGO, K-RVEA, CSEA, and MCEA/D for experiments, respectively. Each algorithm with three initialization methods is compared to the algorithm with the LHS method.

Table S2–Table S5 in the supplementary file show the mean and standard deviation (std) IGD values obtained by four groups of compared algorithms when the number of initial solutions is $NI = 11d - 1$ after 300 function evaluations on the DTLZ1–7 and WFG1–9 test problems. The Wilcoxon rank-sum test at the 5% significance level is used to evaluate the statistical difference between the LHS-based algorithm and the other initialization method-based algorithms. "$+, -, \sim$" indicates that the results obtained by other initialization method-based algorithms are better than, worse than, or similar to those obtained by the LHS-based algorithm, respectively. Table III summarized the statistical results of Table S2–Table S5. These results show that algorithms with the random, LHSmax, OBL, and LHS method for initialization have similar performance on these test problems, where algorithms with the Sobol method outperform the LHS method on more test problems. The reason is that the Sobol method can generate some solutions close to the Pareto set of these test problems (e.g., DTLZ1). From the above results, we can use the Sobol method for initialization in surrogate-based MOEAs since it can obtain better results.

TABLE III

STATISTICAL TEST RESULTS OF THE WILCOXON RANK-SUM TEST OF MOEA/D-EGO, K-RVEA, CSEA, AND MCEA/D WITH DIFFERENT INITIALIZATION METHODS.

Basic Algorithm	random	LHSmax	Sobol	OBL
MOEA/D-EGO	0/0/16	1/0/15	9/1/6	0/0/16
K-RVEA	0/0/16	1/1/14	5/1/10	1/1/14
CSEA	0/2/14	0/1/15	8/0/7	1/0/14
MCEA/D	1/0/15	1/0/15	5/1/10	1/1/14

C. The Effect of the Number of Initial Solutions

To examine the effect of the number of initial solutions on surrogate-based MOEAs, this section compares the performance of surrogate-based MOEAs with different numbers of initial solutions. The LHS method is used for initialization. MOEA/D-EGO, K-RVEA, CSEA, and MCEA/D are used for experiments.

We compare the performance of each algorithm with 20, 40, 60, 80, 100, $11d - 1(109)$, 120, 140, 160, 180, 200, 220, 240, 260, 280, 300 initial solutions. The number of decision

TABLE IV

THE $mean$ RANK VALUES OF EACH ALGORITHM WITH 20, 40, 60, 80, 100, $11d-1(109)$, 120, 140, 160, 180, 200, 220, 240, 260, 280, 300 INITIAL SOLUTIONS ON THE DTLZ AND WFG TEST SUITES WHEN $d = 10$ AND THE MAXIMUM NUMBER OF FUNCTION EVALUATIONS IS 300.

Basic algorithm	Probelms	20	40	60	80	100	$11d-1$	120	140	160	180	200	220	240	260	280	300
MOEA/D-EGO	DTLZ	10.00	10.00	7.71	4.86	7.71	7.71	6.14	8.86	5.57	7.43	7.14	8.71	9.43	9.29	12.14	13.29
	WFG	13.00	11.78	10.44	7.56	6.33	7.78	5.33	5.22	5.22	6.67	6.22	8.00	7.89	10.11	11.67	12.78
K-RVEA	DTLZ	12.14	8.14	6.43	5.14	6.57	4.86	4.86	6.29	6.86	5.29	7.43	9.43	10.71	12.57	14.00	15.29
	WFG	12.67	10.89	9.33	7.11	5.22	4.22	4.78	4.67	4.89	5.89	7.67	8.22	10.56	11.67	13.11	15.11
CSEA	DTLZ	3.00	3.86	4.14	4.00	4.71	5.14	6.43	7.86	8.86	8.29	10.14	11.86	12.86	13.86	15.00	16.00
	WFG	2.33	3.22	3.11	4.00	4.67	5.89	7.67	7.33	8.44	9.11	10.33	12.11	13.00	14.00	14.78	16.00
MCEA/D	DTLZ	7.14	3.86	4.43	4.29	5.43	6.57	6.14	7.57	8.43	10.00	9.71	9.86	11.71	12.43	13.43	15.00
	WFG	12.00	4.33	2.00	2.33	4.33	5.78	4.56	6.89	7.11	9.22	9.89	11.33	13.33	12.78	14.67	15.44

variables is $d = 10$. The maximum number of function evaluations is 300. The mean IGD values obtained by MOEA/D-EGO, K-RVEA, CSEA, and MCEA/D are shown in Table S6, Table S7, Table S8, and Table S9 in the supplementary file, respectively. The IGD values obtained by algorithms with each number of initial solutions are ranked. The rank values are shown in the tables. The best result for each test problem is shaded. Table IV shows the mean rank value of each algorithm with each number of initial solutions on DTLZ and WFG test suites.

From Table S6, Table S7, and Table IV, we can see that the regression-based MOEAs with a small number of initial solutions ($NI \leq 100$) and a large number of initial solutions ($NI \geq 160$) perform worse when $d = 10$. When the number of initial solutions is between 100 and 160, the algorithms usually have similar performance. The reason is that regression-based MOEAs usually build regression models to approximate the objective functions. When using a small number of initial solutions as training data, the regression models may have low approximation accuracy. However, when the number of initial solutions is large, many solutions are used for model building at the beginning of optimization. The number of generated new solutions is limited. A large number of initial solutions may also lead to overlap and reduce the generalization of models.

The results in Table S8, Table S9, and Table IV show that classification-based MOEAs with a small number of initial solutions usually outperform algorithms with a large number of initial solutions. The reason is that classification-based MOEAs usually build classification models to predict the dominance relation between solutions. A small number of training data may be enough for constructing the classification boundary, while a large number of training data may reduce the performance of model.

To further examine the performance of surrogate-based MOEAs with different numbers of initial solutions, we conduct two experiments based on K-RVEA as follows.

- We compare the performance of K-RVEA with 100, 150, 200, 250, 300, $11d-1(329)$, 350, 400, 450, 500 initial solutions. The number of the decision variables is $d = 30$. The maximum number of function evaluations is 500.
- We also compare the performance of K-RVEA with 100, 150, 200, 250, 300, 350, 400, 450, 500, $11d-1(549)$, 600, 650, 700, 750, 800 initial solutions. The number of the decision variables is $d = 50$. The maximum number of function evaluations is 800.

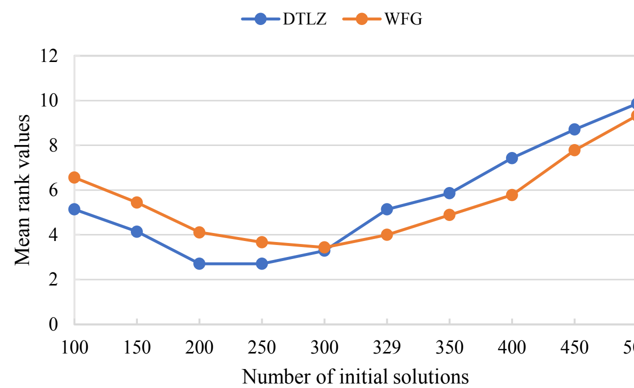

Fig. 1. The $mean$ rank values of K-RVEA with 100, 150, 200, 250, 300, $11d-1(329)$, 350, 400, 450, 500 initial solutions when $d = 30$ and the maximum number of function evaluations is 500 on DTLZ and WFG test suites.

The mean IGD values obtained from the above two experiments are shown in Table S10 and Table S11 in the supplementary file. The rank value of K-RVEA with each number of initial solutions on each test problem is presented. Fig. 1 and Fig. 2 plot the mean rank value of K-RVEA with each number of initial solutions on DTLZ and WFG test suites, respectively. The experimental results in Fig. 1 and Fig. 2 also show that K-RVEA with a small number of initial solutions ($NI \leq 200$) and a large number of initial solutions ($NI \geq 300$ when $d = 30$ and $NI \geq 500$ when $d = 50$) perform worse. The better results are obtained with 250, 300 initial solutions when $d = 30$ and with 300, 350, 400, 450, initial solutions when $d = 50$.

To further examine the performance of classification-based MOEAs with different numbers of initial solutions, we compare the performance of MCEA/D with 20, 40, 60, 80, 100, 250, $11d-1(329)$, 400, 500 initial solutions. The number of the decision variables is $d = 30$. The maximum number of function evaluations is 500. The mean IGD values obtained by MCEA/D with each number of initial solutions are shown in Table S12 in the supplementary file. The rank value of MCEA/D with each specification on each test problem is calculated. Fig. 3 plots the mean rank value of MCEA/D

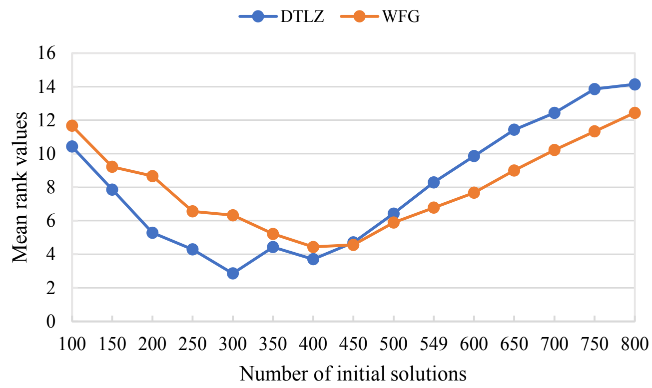

Fig. 2. The $mean$ rank values of K-RVEA with 100, 150, 200, 250, 300, 350, 400, 450, 500, $11d-1(549)$, 600, 650, 700, 750, 800 initial solutions when $d = 50$ and the maximum number of function evaluations is 800 on DTLZ and WFG test suites.

with each number of initial solutions on DTLZ and WFG test suites, respectively. The results show that MCEA/D with 40, 60, and 80 initial solutions obtain better results than other specifications when $d = 30$.

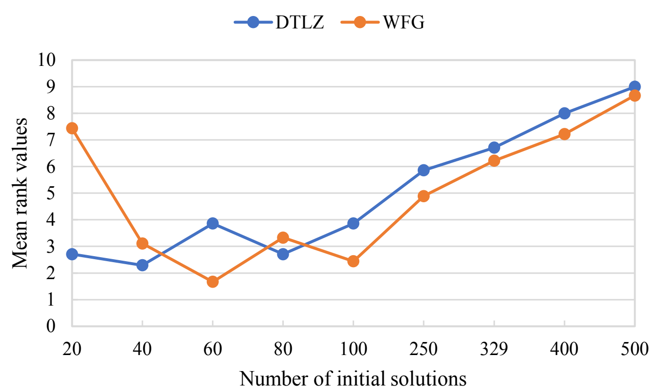

Fig. 3. The $mean$ rank values of MCEA/D with 20, 40, 60, 80, 100, 250, $11d-1(329)$, 400, 500 initial solutions when $d = 30$ and the maximum number of function evaluations is 500 on DTLZ and WFG test suites.

From the above results, we can conclude that regression-based algorithms have bad performance when the number of initial solutions is too large or too small. They usually have good performance when the number of initial solutions is around half of the maximum number of function evaluations. Classification-based algorithms usually have good performance when the number of initial solutions is small. Based on these experimental results, we suggest that the appropriate number of initial solutions for regression-based MOEAs can be $MaxFEs/2$ for $MaxFEs \in [100, 800]$ where MaxFEs is the total number of examined solutions. The appropriate number of initial solutions for classification-based algorithms can be $MaxFEs/10 + 10$.

IV. Conclusion

In this paper, we analyzed the effects of the LHS method and the number of initial solutions on six commonly used surrogate-based MOEAs, including three regression-based MOEAs and three classification-based MOEAs. The experiments were conducted on two widely used test suites and the results were evaluated using the IGD metric. First, the findings showed that the LHS method and the random method produced similar performance results for surrogate-based MOEAs. The experiments of using other initialization methods (LHSmax, Sobol, and OBL method) showed that Sobol performs better than LHS method and could be used for initialization. Then, the findings showed that regression-based MOEAs performed better when the number of initial solutions was close to half of the maximum number of function evaluations, while classification-based MOEAs performed better with a smaller number of initial solutions. We suggest to use $MaxFEs/2$ as the number of initial solutions for regression-based MOEAs and $MaxFEs/10+10$ for classification-based MOEAs. Since classification-based algorithms need less initial solutions, it seems that they are more useful under a very limited computational resources (e.g., 100 solution evaluations in total) in comparison with regression-based algorithms. For the same reason, classification-based algorithms may be more useful for large-scale problems with many decision variables (e.g., 100, 500, 1000) and many objectives (e.g., 5 and 10). For example, in [30], DTLZ and WFG with 11 objectives and 150 decision variables are used to demonstrate the usefulness of a classification-based algorithm. Based on these results, future research can focus on developing new and improved surrogate-based algorithms by using other initialization methods and using other number of initial solutions.

References

[1] L. Jia, Y. Wang, and L. Fan, "Multiobjective bilevel optimization forproduction-distribution planning problems using hybrid genetic algorithm," *Integrated Computer-Aided Engineering*, vol. 21, no. 1, pp. 77–90, Jan. 2014.

[2] T. Chugh, K. Sindhya, K. Miettinen, Y. Jin, T. Kratky, and P. Makkonen, "Surrogate-assisted evolutionary multiobjective shape optimization of anair intake ventilation system," in *Proceedings of the 2017 IEEE Congresson Evolutionary Computation (CEC 2017)*, Donostia, Spain, Jul. 2017, pp. 1541–1548.

[3] 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, Apr. 2002.

[4] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, Tech. Rep. 103, 2001.

[5] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, Sep. 2007.

[6] E. Zitzler and S. Kunzli, "Indicator-based selection in multiobjective search," in *Proceedings of International Conference on Parallel Problem Solving from Nature (PPSN 2004)*, Birmingham, UK, Sep. 2004, pp. 832–842.

[7] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 609–622, Aug. 2018.

[8] T. Murata and H. Ishibuchi, "MOGA: multi-objective genetic algorithms," in *Proceedings of 1995 IEEE Congress on Evolutionary Computation (CEC 1995)*, Perth, Australia, Nov. 1995, pp. 289–294.

[9] 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, Dec. 2007.

[10] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

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

[12] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms," *Soft Computing*, vol. 23, no. 9, pp. 3137–3166, Dec. 2019.

[13] K. Deb, R. Hussein, P. C. Roy, and G. Toscano-Pulido, "A taxonomy for metamodeling frameworks for evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 104–116, Feb. 2019.

[14] Z. Song, H. Wang, C. He, and Y. Jin, "A Kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 6, pp. 1013–1027, Dec. 2021.

[15] J. Knowles, "ParEGO: A hybrid algorithm with on-line landscape approximationfor expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, Feb. 2006.

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

[17] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, "Multiobjective optimization on a limited budget of evaluations using model-assisted S-metric selection," in *Proceedings of International Conference on Parallel Problem Solving from Nature (PPSN 2008)*, Dortmund, Germany, Sep. 2008, pp. 784–794.

[18] 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, Jun. 2009.

[19] 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, Feb. 2018.

[20] A. Habib, H. K. Singh, T. Chugh, T. Ray, and K. Miettinen, "A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 1000–1014, Dec. 2019.

[21] L. Pan, H. Cheng, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 74–88, Feb. 2019.

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

[23] D. Han and J. Zheng, "A Kriging model-based expensive multiobjective optimization algorithm using R2 indicator of expectation improvement," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–16, Jun. 2020.

[24] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, "An adaptive Bayesian approach to surrogate-assisted evolutionary multi-objective optimization," *Information Sciences*, vol. 519, pp. 317–331, May 2020.

[25] Z. Song, H. Wang, and H. Xu, "A framework for expensive many-objective optimization with Pareto-based bi-indicator infill sampling criterion," *Memetic Computing*, vol. 14, pp. 179–191, Nov. 2022.

[26] 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 Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 4, pp. 2084–2097, Apr. 2022.

[27] Y. Yuan and W. Banzhaf, "Expensive multi-objective evolutionary optimization assisted by dominance prediction," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 1, pp. 159–173, Feb. 2022.

[28] H. Hao, A. Zhou, H. Qian, and H. Zhang, "Expensive multiobjective optimization by relation learning and prediction," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 1157–1170, Oct. 2022.

[29] 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 Transactions on Evolutionary Computation*, vol. 26, no. 4, pp. 631–645, Aug. 2022.

[30] T. Sonoda and M. Nakata, "Multiple classifiers-assisted evolutionary algorithm based on decomposition for high-dimensional multi-objective problems," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 6, pp. 1581–1595, Dec. 2022.

[31] I. Loshchilov, M. Schoenauer, and M. Sebag, "A mono surrogate for multiobjective optimization," in *Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2010)*, Portland, USA, Jul. 2010, pp. 471–478.

[32] S. Bandaru, A. H. Ng, and K. Deb, "On the performance of classification algorithms for learning Pareto-dominance relations," in *Proceedings of 2014 IEEE Congress on Evolutionary Computation (CEC 2014)*, Beijing, China, Jul. 2014, pp. 1139–1146.

[33] J. Zhang, A. Zhou, and G. Zhang, "A classification and Pareto domination based multiobjective evolutionary algorithm," in *Proceedings of 2015 IEEE Congress on Evolutionary Computation (CEC 2015)*, Sendai, Japan, May 2015, pp. 2883–2890.

[34] J. Zhang, A. Zhou, K. Tang, and G. Zhang, "Preselection via classification: A case study on evolutionary multiobjective optimization," *Information Sciences*, vol. 465, pp. 388–403, Oct. 2018.

[35] X. Lin, Q. Zhang, and S. Kwong, "A decomposition based multiobjective evolutionary algorithm with classification," in *Proceedings of 2016 IEEE Congress on Evolutionary Computation (CEC 2016)*, Vancouver, Canada, Jul. 2016, pp. 3292–3299.

[36] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2016.

[37] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[38] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*, London, UK, 2005, pp. 105–145.

[39] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Proceedings of International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, Guanajuato, Mexico, March 2005, pp. 280–295.

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

[41] J. Zhang, L. He, and H. Ishibuchi, "Dual fuzzy classifier-based evolutionary algorithm for expensive multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, 2022 (Early Access).

[42] B. Kazimipou, X. Li, and A. K. Qin, "A review of population initialization techniques for evolutionary algorithms," in *Proceedings of 2014 IEEE Congress on Evolutionary Computation (CEC 2014)*, Beijing, China, Jul. 2014, pp. 2585–2592.

[43] C. A. C. Coello and M. R. Sierra, "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm," in *Proceedings of Mexican International Conference on Artificial Intelligence (MICAI 2004)*, Mexico City, Mexico, Apr. 2004, pp. 688–697.

[44] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, Nov. 2017.

[45] B. Beachkofski and R. Grandhi, "Improved distributed hypercube sampling," in *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Denver, USA, Apr. 2002, pp. 580–586.

[46] X. Wang and I. H. Sloan, "Low discrepancy sequences in high dimensions: How well are their projections distributed?" *Journal of Computational and Applied Mathematics*, vol. 213, no. 2, pp. 366–386, Apr. 2008.

[47] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, Feb. 2008.