

# Evolutionary Large-Scale Multiobjective Optimization: Benchmarks and Algorithms

Songbai Liu, *Student Member, IEEE*, Qiuzhen Lin, *Member, IEEE*, Ka-Chun Wong, Qing Li, *Senior Member, IEEE*, Kay Chen Tan, *Fellow, IEEE*

**Abstract**—Evolutionary large-scale multiobjective optimization (ELMO) has received increasing attention in recent years. This study has compared various existing optimizers for ELMO on different benchmarks, revealing that both benchmarks and algorithms for ELMO still need significant improvement. Thus, a new test suite and a new optimizer framework are proposed to further promote the research of ELMO. More realistic features are considered in the new benchmarks, such as mixed formulation of objective functions, mixed linkages in variables, and imbalanced contributions of variables to the objectives, which are challenging to the existing optimizers. To better tackle these benchmarks, a variable group-based learning strategy is embedded into the new optimizer framework for ELMO, which significantly improves the quality of reproduction in large-scale search space. The experimental results validate that the designed benchmarks can comprehensively evaluate the performance of existing optimizers for ELMO and the proposed optimizer shows distinct advantages in tackling these benchmarks.

**Index Terms**—Evolutionary algorithm, Large-Scale Optimization, Multiobjective Optimization, Benchmarks.

## I. INTRODUCTION

Large-scale multiobjective optimization problems (LMOPs) have widely existed in many real-world applications, e.g., complex community detection [1], capacitated vehicle routing [2], and deep neural architecture search [3]. Mathematically, an unconstrained and continuous LMOP can be formulated as follows:

$$\text{Minimize } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})), \mathbf{x} \in \Omega, \quad (1)$$

where  $\mathbf{x} = (x_1, \dots, x_n)$  indicates a solution vector of the LMOP with  $n$  often interacting variables in the search space  $\Omega$  ( $n \geq 100$ ) and  $f_1(\mathbf{x}), \dots, f_m(\mathbf{x})$  represent  $m$  often mutually conflicting objective functions ( $m \geq 2$ ). In the case of  $n < 100$ , an LMOP is often called a multiobjective optimization problem when  $m = 2$

---

Manuscript received xx. xx. 2021; revised xx. xx. 2021; accepted xx. xx. 2021. This work is supported in part by the National Key Research and Development Project, Ministry of Science and Technology, China (Grant No. 2018AAA0101301), and in part by the National Natural Science Foundation of China (NSFC) under Grant 61876162 and Grant 61876110, and in part by the Research Grants Council of the Hong Kong SAR under Grant No. PolyU11202418 and Grant No. PolyU11209219, and in part by the Shenzhen Scientific Research and Development Funding Program under Grant JCYJ20190808164211203. (Corresponding authors: Qiuzhen Lin; Kay Chen Tan).

S.B. Liu and K.C. Wong are with the Department of Computer Science, City University of Hong Kong, Hong Kong (email: songbai209@qq.com).

Q.Z. Lin is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China.

Q. Li and K.C. Tan are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR (emails: csqli@comp.polyu.edu.hk, kctan@polyu.edu.hk).

or 3, or a many-objective optimization problem when  $m > 3$ , which are all abbreviated as MOP in this paper. Similar to MOPs, no single optimal solution exists for LMOPs, but a set of equally optimal solutions will be found in the search space  $\Omega$  [4], which is called Pareto-optimal set (PS). The mapping of PS in the objective space is called Pareto-optimal front (PF).

During the last decades, a number of multi/many-objective evolutionary algorithms (MOEAs) have been reported to solve MOPs. Based on the environmental selection strategies in the objective space, most of MOEAs can be roughly classified into three categories: Pareto-based MOEAs [5]–[6], decomposition-based MOEAs [7]–[8], and indicator-based MOEAs [9]–[10]. These MOEAs mainly focus on improving the environmental selection run in the objective space, which have shown promising performance for solving general MOPs. However, they will encounter more challenges in tackling LMOPs, as the properties of LMOPs (e.g., multimodality, interaction between variables, and landscape) will become complicated due to the increased number of variables [13]. Moreover, most MOEAs usually use traditional evolutionary operators (e.g., polynomial mutation [5], simulated binary crossover [6], particle swarm optimization [11], and differential evolution [8]) to search the variable space, which cannot effectively generate high-quality solutions for solving LMOPs [12].

Under this context, evolutionary large-scale multiobjective optimization (ELMO) has received increasing attention in recent years and a number of large-scale MOEAs (LMOEAs) have been proposed for solving LMOPs. Based on their specific mechanisms, most of LMOEAs can be roughly classified into the following four types: LMOEAs based on the cooperative coevolution framework (CCF) [12]–[19], LMOEAs with a problem reformulation mechanism (PRM) [20]–[24], LMOEAs via an enhanced reproduction strategy (ERS) [25]–[34], and LMOEAs with a dimensional reduction technique (DRT) [35]–[37]. However, the above four types of LMOEAs have their own pitfalls in solving LMOPs, the details of which will be introduced in Section II. Moreover, some well-known test suites with variable scalabilities for MOPs, e.g., ZDT [38], DTLZ [39], WFG [40], and UF [41], are used to evaluate the performance of most LMOEAs. Here, the variable scalability of an MOP refers to that the number of variables  $n$  in (1) can be set any arbitrary positive integer. However, some common features of real-world LMOPs, e.g., different objective functions related to different groups of variables, nonuniform grouping of variables, mixed overlapping between different variable groups, mixed separability of variables, and various correlations between variables, are not studied in these MOPs.

In addition, a new test suite [42] is proposed for ELMO, but it still has limitations to approximate real-world LMOPs. As

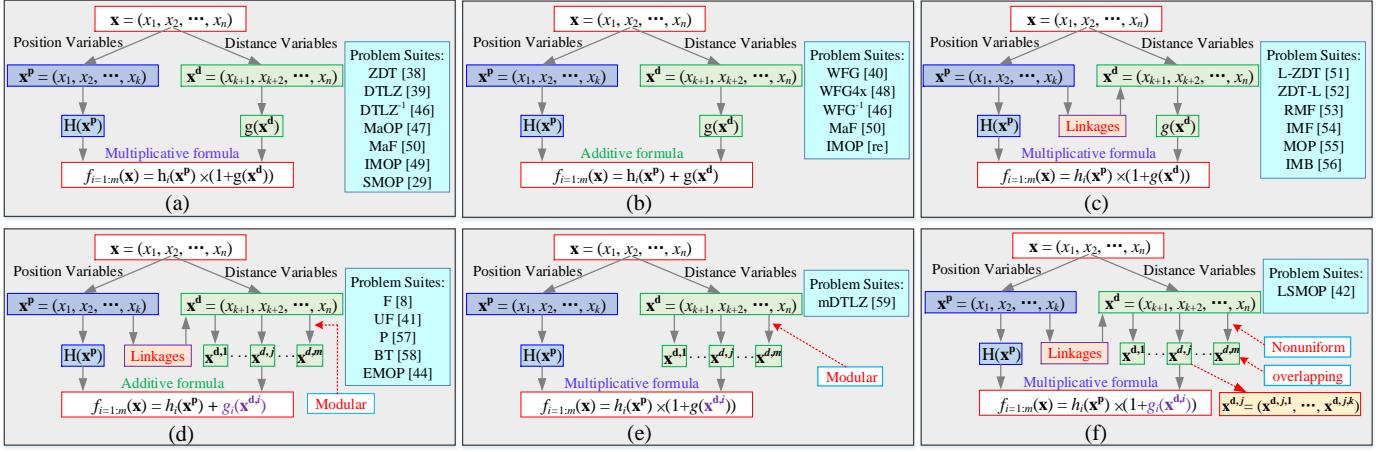


Fig. 1 Illustration of existing test LMOPs based on formulation model and variable linkages.

experimentally studied in Section II, both the benchmarks and algorithms are still not satisfactory for the study of ELMO. To further promote the advancement of ELMO, this paper has made the following three contributions:

- (1) This paper presents an in-depth review of the existing test LMOPs and LMOEAs, which has investigated their limitations for ELMO in different aspects.
- (2) This paper designs a new test suite for ELMO, which shows more features to approximate real-world LMOPs, e.g., mixed formulation of objective functions, mixed linkages between variables, and imbalanced contributions of different variables to the objectives.
- (3) This paper proposes a new LMOEA framework based on a variable group-based learning strategy, which can achieve a proper balance between convergence speed and computational burden when reproducing offspring solutions with robust qualities.

The rest of this paper is organized as follows. At first, a review of test LMOPs and LMOEAs is given in Section II. Then, a new LMOP test suite and a new LMOEA framework are introduced in Section III and Section IV, respectively. In Section V, a number of experiments are given to study the performance of our proposed LMOEA on the designed test LMOPs. At last, the conclusions and future work are summarized in Section VI.

## II. A REVIEW OF TEST LMOPS AND LMOEAS

In most of existing studies about ELMO [12]-[18], the MOP test suites with variable scalability are usually used as benchmarks by directly setting their dimension of variables  $n \geq 100$  to study the performance of LMOEAs. However, it is debatable whether such a direct and simple modification can reflect the real nature of LMOPs and whether they can be directly used to study the merits and demerits of a newly proposed LMOEA. Thus, a review of existing test LMOPs with respect to their design principles and characteristics is given in Section II.A, and then a review of existing LMOEAs is provided in Section II.B. Finally, some discussions of ELMO with respect to benchmarks and algorithms are given in Section II.C.

### A. Review of Existing Test LMOPs

The bottom-up method [43] is the most classical strategy to

design test MOPs. In this method,  $n$  variables of a solution  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  are classified into two groups: a position-related group  $\mathbf{x}^p$  with  $k$  variables ( $x_1^p, \dots, x_k^p$ ) and a distance-related group  $\mathbf{x}^d$  with  $l$  variables ( $x_1^d, \dots, x_l^d$ ), where  $n = k + l$ . Then, a set of  $m$  shape functions  $\mathbf{H}(\mathbf{x}^p) = (h_1(\mathbf{x}^p), \dots, h_m(\mathbf{x}^p))$  is defined on  $\mathbf{x}^p$  to determine the conflict among objective functions and the shapes of PF, while a set of  $m$  landscape functions  $\mathbf{G}(\mathbf{x}^d) = (g_1(\mathbf{x}^d), \dots, g_m(\mathbf{x}^d))$  is defined on  $\mathbf{x}^d$  to determine the interaction among variables and the complexity of PS. Most test LMOPs are extended from the test MOPs designed based on the above method. In addition, the  $m$  objective functions of an LMOP are further formulated by the following two models:

#### Model 1: Multiplication-based formula model

$$\min f_{i=1:m}(\mathbf{x}) = h_i(\mathbf{x}^p) \times (1 + g_i(\mathbf{x}^d)) \quad (2)$$

#### Model 2: Addition-based formula model

$$\min f_{i=1:m}(\mathbf{x}) = h_i(\mathbf{x}^p) + g_i(\mathbf{x}^d) \quad (3)$$

where  $g_i(\mathbf{x}^d) \geq 0$  ( $i = 1, 2, \dots, m$ ). These models can produce the solutions to spread in different positions [44]. Thus, the design of an LMOP by this bottom-up method is transferred to customizing its three main components, i.e., the shape functions, the landscape functions, and the formula model. Because of its flexible plasticity in constructing various PFs and PSs [45], the bottom-up method has been also used to design a number of test suites for ELMO as shown in Fig. 1.

As shown in Fig. 1(a), the first test suite based on **Model 1** is ZDT [38]. Its most obvious limitation is the lack of objective scalability, i.e.,  $m$  is fixed to 2 in ZDT. In general, the scalabilities on objectives and variables are respectively determined by the design of  $\mathbf{H}(\mathbf{x}^p)$  and  $\mathbf{G}(\mathbf{x}^d)$ . In this context, DTLZ [39] was presented based on **Model 1** with objective scalability by tailoring  $\mathbf{H}(\mathbf{x}^p)$  inspired from spherical coordinates. In addition, the first LMOP test suite based on **Model 2** is WFG [40] as shown in Fig. 1(b) and its decision vector  $\mathbf{x}$  with  $n$  variables is transferred into an  $m$  dimensional manipulative vector via a series of transition vectors. Specifically,  $\mathbf{x}^p$  in WFG is transformed to an  $m-1$  dimensional vector and thus  $k$  can be set to a multiple of  $m-1$ , while  $k$  in both ZDT and DTLZ is fixed to  $m-1$ .

Undoubtedly, ZDT, DTLZ, and WFG are the three most widely used test suites in the research of MOEAs. However, they have three main limitations: 1) The PF shapes designed

based on  $\mathbf{H}(\mathbf{x}^p)$  are relatively regular, and the PSs determined mainly by  $\mathbf{G}(\mathbf{x}^d)$  are relatively simple; 2)  $\mathbf{H}(\mathbf{x}^p)$  and  $\mathbf{G}(\mathbf{x}^d)$  can be optimized separately as they are mutually independent and variables in  $\mathbf{x}^p$  (or in  $\mathbf{x}^d$ ) are nearly uncorrelated or weakly dependent [40]; and 3) all the  $m$  objective functions share the same landscape function  $g(\mathbf{x}^d)$ . In recent years, there are some test suites proposed with various irregular PF shapes by formulating  $H(\mathbf{x}^p)$ , e.g., DTLZ<sup>-1</sup> [46], MaOP [47] based on **Model 1**, WFG4x [48], WFG<sup>-1</sup> [46] based on **Model 2**, IMOP [49], MaF [50] based on **Model 1** and **Model 2**, as listed in Fig.1(a) and Fig.1(b). Furthermore, the linkages between variables (or variable dependencies) are further embedded into benchmarks [51]-[56], which can be roughly divided into two types below.

**Type1:** the variables of  $\mathbf{x}^p$  in L-ZDT [51], ZDT-L [52], and some benchmarks (called RMF [53] and IMF [54] here) are directly linked to the variables of  $\mathbf{x}^d$  with linear or nonlinear mapping in the original ZDT or DTLZ test suites. Moreover, other imbalanced test suites, e.g., MOP [55] and IMB [56], are customized via introducing biased mapping in benchmarks. These test suites are listed in Fig.1(c) based on **Model 2**.

**Type2:** the variables of  $\mathbf{x}^d$  are further divided into  $m$  subgroups ( $\mathbf{x}^{d,1}, \dots, \mathbf{x}^{d,m}$ ) via a modular method [57] based on **Model 1**. Thus, each objective function  $f_i(\mathbf{x})$  has its own unique landscape function  $g_i(\mathbf{x}^{d,i})$  defined on the corresponding  $\mathbf{x}^{d,i}$  ( $i = 1, \dots, m$ ), which is shown in Fig. 1(d). As pointed out in [45], this method has realized a direct correlation between  $\mathbf{H}(\mathbf{x}^p)$  and  $\mathbf{G}(\mathbf{x}^d)$ . Moreover, in the F [8], UF [41], and P [57] test suites, variable linkages are further used to enhance the dependency between variables and complicated PS shapes. In the BT [58] and EMOP [44] test suites, biased mapping is specially designed to link  $\mathbf{x}^p$  and  $\mathbf{x}^d$ , which can increase the difficulty of approximating the whole PFs. Analogously, for the mDTLZ [59] test suit shown in Fig. 1(e), the division of  $\mathbf{x}^d$  with modular method is also applied to modify DTLZ, leading to hardly dominated boundaries in PF. As indicated in [45], the strategies in this category are more realistic.

The two types of benchmarks introduced above also have three obvious limitations: 1) every subgroup  $\mathbf{x}^{d,i}$  has almost the same number of variables, which means that this is a uniform division of  $\mathbf{x}^d$  via the modular method; 2) each function  $g_i(\mathbf{x}^{d,i})$  is almost completely separable; 3) there is no intersection between any two subgroups, i.e., this is a non-overlapping grouping of  $\mathbf{x}^d$ , which is inadequate to better resemble characteristics of real-world problems, especially for LMOPs [42]. Strictly speaking, the test suites introduced above are simply extended from the well-known benchmarks for general MOPs, which are lack of some common features of real-world LMOPs, e.g., different objective functions related to different groups of variables, nonuniform grouping of variables, mixed overlapping between different variable groups, mixed separability of variables, and various correlations between variables.

The LSMOP [42] test suit shown in Fig. 1(f) may be the first one specifically designed for ELMO, which uses a logistic map to nonuniformly divide the variables of  $\mathbf{x}^d$  and adopts different correlation matrixes to classify the variables of  $\mathbf{x}^d$  into overlapped groups. Moreover, each subgroup  $\mathbf{x}^{d,i} \in \mathbf{x}^d$  is further evenly divided into  $K$  subcomponents, i.e.,  $\mathbf{x}^{d,i} =$

$(\mathbf{x}^{d,i,1}, \dots, \mathbf{x}^{d,i,K})$ , which can introduce mixed separability in LSMOP. After that, there have more test suites specially designed for ELMO. For example, an SMOP test suite based on **Model 1** is proposed in [29] to challenge the performance of LMOEAs for handling sparse LMOPs and a real-world LMOP test suite called TREE is designed in [65] to simulate the time-varying ratio error estimation problem for the voltage transformers. Except for the above bottom-up methods, there also have other strategies that can be used to design LMOPs, e.g., in [66]-[67], the strategy of distance minimization from multiple points is used to expand the variable space of multi-objective polygon test problems.

Please note that the detailed features and limitations of  $\mathbf{H}(\mathbf{x}^p)$  and  $\mathbf{G}(\mathbf{x}^d)$  in the existing LMOPs can refer to [45], and some concepts in benchmarks including overlapping, linkages, and separability between variables are further introduced in the supplementary file of this paper.

### B. Review of Existing LMOEAs

As introduced in Section I, according to the adopted search strategies, most of existing LMOEAs can be roughly classified into four categories: 1) CCF-based LMOEAs [12]-[19]; 2) PRM-based LMOEAs [20]-[24]; 3) ERS-based LMOEAs [25]-[33]; and 4) DRT-based LMOEAs [34]-[36], which are summarized in Table SI of the supplementary file to introduce their characteristics, the used test suites, the variable dimensions and the maximal allowable function evaluations ( $FE_{max}$ ).

In CCF-based LMOEAs, the target LMOP is first divided into  $M$  exclusive low-dimensional subproblems via a variable grouping method, followed by selecting an existing solver to separately optimize each subproblem [12]. Here, a solution of each subproblem is encoded by the variables in its associated variable group, which is a partial vector of the complete solution to the target LMOP. Thus, its function evaluation needs to cooperate with other  $M-1$  partial solutions from their corresponding subproblems [15]. As variables of an LMOP are often interacted with each other, the performance of CCF strongly depends on the accuracy of variable grouping [18]. That is to say, some related variables classified into different groups may induce a misleading search direction [19].

For PRM-based LMOEAs, the target LMOP is reformulated by transferring the search space from the original large-scale variable space to a low-dimensional weighted variable space, aiming to accelerate the convergence speed [21]. Then, the original variables are updated according to the change of their associated weight values. Thus, an original variable in PRM cannot be changed independently and the variables associated with the same weight are updated on the same way, which substantially limits the search to certain reachable original spaces [34].

On ERS-based LMOEAs, they aim to reproduce offspring directly and efficiently in the large-scale search space without any preprocess of the target LMOP. In this case, the quality of the offspring would be very poor if the search was still conducted using the traditional evolutionary operators, e.g., SBX crossover and polynomial mutation [4]. Therefore, they often design new and effective search strategies or improve

TABLE I  
SETTINGS OF THREE EXPERIMENTAL TESTS: E1, E2, AND E3

No. Test	$n$	$m$	$N$	$FE_{\max}$	Test LMOPs	Considered LMOEAs
E1	50, 100, 500, 1000	2,3	100	$10^5$	DTLZ, UF, IMF, LSMOP	MOEA/DVA, WOF, LSMOF, LMOCSO, DGEA
E2	$10^2, 300, 500, 800, 10^3, 2000$	2,3	$100 \times m$	$10^6$	IMF, LSMOP	MOEA/DVA, WOF, LMEA, LSMOF, LMOCSO, DGEA
E3	100, 300, 600, 1000	2	200	$n \times 10^4$	UF, LSMOP	MOEA/DVA, WOF, LMEA, LSMOF, LMOCSO

Notes:  $n$  and  $m$  correspondingly indicate the number of variables and the number of objectives set for the target LMOP.  $N$  and  $FE_{\max}$  respectively represent the population size and the maximum function evaluations set in the corresponding LMOEA. E1, E2, and E3 indicate three different experimental setups, respectively.

traditional search strategies to achieve the efficient reproduction of offspring [25]-[27]. However, when computational resource is limited, their performance is extremely sensitive to the variable dimensionality due to their slow convergence speed, which will be further investigated in Section II.C.

Regarding DRT-based LMOEAs, they assume that redundant variables may exist in the large-scale variable space of LMOPs, so machine learning methods can be used to achieve dimensionality reduction of the search space [35]. However, whatever DRT models used for dimensionality reduction will encounter two main challenges: 1) how to ensure the rationality of the training and test data, as the data selected are partially distributed to poorly match the PS/PF during the evolutionary process; 2) how to avoid the high computational cost due to the online running of machine learning methods. This kind of LMOEAs is often customized for solving specific types of LMOPs, e.g., SMOPs [29].

### C. Discussions about ELMO

#### 1) Discussions about LMOP benchmarks

Proverbially, benchmarks are of great importance to algorithm analysis, which can help to analyze the merit and demerit of the target algorithms. As summarized in Table SI, the test suites that were originally designed as benchmarks for general MOPs, e.g., ZDT, DTLZ, WFG, IMF, and UF, are widely used to assess the performance of existing LMOEAs. However, as discussed in [26], the test suites listed in Fig. 1(a)-(b) are not suitable for assessing the performance of LMOEAs. In these benchmarks,  $\mathbf{H}(\mathbf{x}^p)$  and  $\mathbf{G}(\mathbf{x}^d)$  without variable linkages can be optimized separately, and  $\mathbf{H}(\mathbf{x}^p)$  is always defined based on a few variables ( $k = m - 1$ ), which only requires the algorithms with strong convergence ability to solve them. The algorithms focusing on convergence, e.g., PRM-based LMOEAs, show distinct advantages in solving the problems of these test suites. Regarding the test suites listed in Fig. 1(c)-(d), although the correlations between  $\mathbf{H}(\mathbf{x}^p)$  and  $\mathbf{G}(\mathbf{x}^d)$  have been considered, their three limitations described in Section II.A still make them disqualified as LMOP benchmarks. Although the LSMOP test suite is specifically designed for ELMO, it still has several limitations, which are discussed as follows.

First, as pointed out in [60], the Griewank's function should not be considered in an LMOP due to the fact that the function becomes easier with the increased variable dimension. However, this function is included in both LSMOP2 and LSMOP4 problems. Second, the PF shapes involved in LSMOP are too regular [45], which can be easily solved by LMOEAs guided with reference vectors. Third, the number of variables in  $\mathbf{x}^p$  is still fixed as  $k = m - 1$ , which is very few for LMOPs. Fourth, it is not so flexible or realistic by using correlation matrixes to

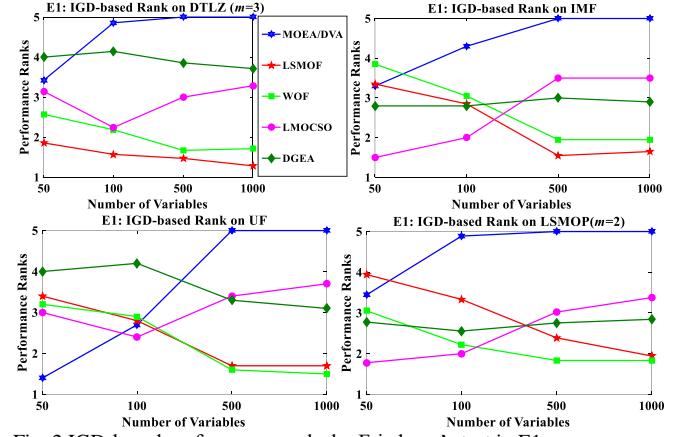


Fig. 2 IGD-based performance ranks by Friedman's test in E1

control overlapping variables between different groups of  $\mathbf{x}^d$ . At last, each group  $\mathbf{x}^{d,i} \in \mathbf{x}^d$  is evenly divided into  $K$  sub-components, and the imbalanced contributions of different subcomponents to the corresponding objective function are not considered in LSMOP, while such imbalanced feature occurs frequently in real-world problems [60]. Therefore, it is reasonable to conclude that benchmarks for ELMO are still not so satisfactory. In this context, this paper is motivated to propose a new LMOP test suite in Section III, which should include more features to closely approximate real-world LMOPs.

#### 2) Discussions about LMOEAs

As shown in the last two columns of Table SI, the settings of key parameters, i.e.,  $n$  and  $FE_{\max}$ , in the experimental design of existing ELMO studies are extremely different. In fact, almost every reported LMOEA was validated to be the best under its used parameter setting. Thus, in order to analyze the strengths and shortcomings of various LMOEAs, it is necessary to make a comprehensive comparison for them under a fair and diverse experimental setting. Here, the experimental comparisons are made by using three parameter settings (i.e., E1, E2, and E3) as listed in Table I, which gives the number of variables  $n$ , the number of objectives  $m$ , the population size  $N$ , the maximal function evaluations  $FE_{\max}$ , the test LMOPs and the compared LMOEAs in each comparison.

The mean results of Inverted Generational Distance (IGD) [61] are collected from 20 independent runs for each LMOEA on each test problem. Due to page limitations, these IGD results are provided in the supplementary file. Then, all the compared LMOEAs are ranked by Friedman's test in KEEL [62] for each experiment, which quantifies how each optimizer performs on the corresponding test suites. Please note that the lower rank indicates the better performance of LMOEA.

In Fig. 2, the IGD-based performance ranks obtained by

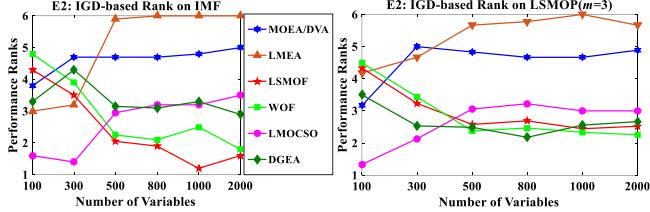


Fig. 3 IGD-based performance ranks by Friedman's test in E2

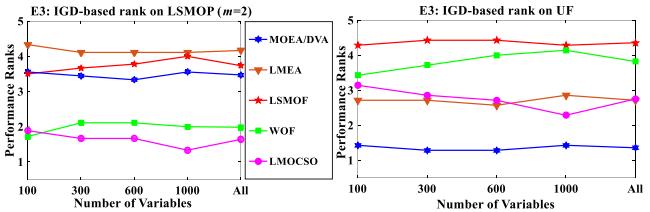


Fig. 4 IGD-based performance ranks by Friedman's test in E3

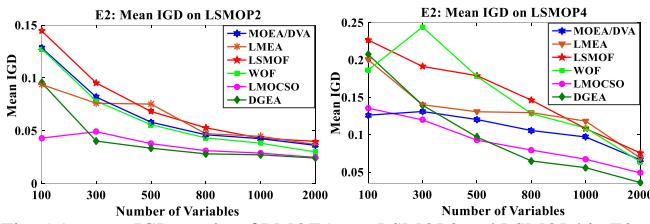


Fig. 5 Average IGD results of LMOEAs on LSMOP2 and LSMOP4 in E2.

using Friedman's test are plotted when running five different LMOEAs (MOEA/DVA [13], WOF [20], LSMOF [21], LMOCSO [27], DGEA [34]) to solve four test suites (DTLZ [39], UF [41], IMF [54], LSMOP [42]) with the parameter setting E1. Based on their performance ranks, the two PRM-based LMOEAs (WOF and LSMOF) have shown significant advantages on these four test suites in the cases of  $n = 500$  and  $n = 1000$ , but they are always beaten by the two ERS-based LMOEAs (LMOCSO and DGEA) in the case of  $n \leq 100$  except on DTLZ. In addition, the CCF-based LMOEA (MOEA/DVA) obtains the worst overall performance, which can hardly beat any other LMOEAs on all considered test suites in the cases of  $n \geq 100$ . As shown in Fig. 3 with the parameter setting E2, six LMOEAs are adopted to solve two test suites (IMF and LSMOP). Similarly, the two ERS-based LMOEAs perform better than the two PRM-based LMOEAs in the cases of  $n \leq 300$ , but they are defeated by the PRM-based LMOEAs in the cases of  $n \geq 500$ , while the two CCF-based LMOEA (MOEA/DVA and LMEA) are the worst in the cases of  $n \geq 300$ . Here, the main difference between E1 and E2 is to increase the given  $FE_{\max}$  from  $10^5$  in E1 to  $10^6$  in E2. To have a further verification,  $FE_{\max}$  is set to  $n \times 10^4$  in E3, which is unaffordable when  $n$  is extremely large. As shown in Fig. 4, the performance ranks of the five LMOEAs on LSMOP and UF in E3 are almost completely different from that in E1 and E2, MOEA/DVA that has the worst performance on all considered test suites in both E1 and E2 performs best on UF in E3. In addition, the rank curves in Fig. 4 remain stable with only minor fluctuations, and LMOCSO is always the best on LSMOP with the parameter setting E3. From the above ex-

perimental results, the following observations and conclusions can be found.

First, when a limited  $FE_{\max}$  is given, e.g.,  $FE_{\max}=10^5$  in E1 and  $FE_{\max} = 10^6$  in E2, the performance of two CCF-based LMOEAs and two ERS-based LMOEAs is degraded with the increase of  $n$ . Accountably, the limited  $FE_{\max}$  is only enough for MOEA/DVA (or LMEA) to do variable analysis, as few resources are left to optimize LMOPs when  $n$  is increased to a certain value (e.g.,  $n=100$  in E1 and  $n=500$  in E2). For the two ERS-based LMOEAs, they perform well on LMOPs with a relatively low value of  $n$  (e.g.,  $n \leq 100$  in E1 and  $n \leq 500$  in E2), but become ineffective on the same LMOPs with other higher values of  $n$  based on the limited  $FE_{\max}$ . On the two PRM-based LMOEAs (WOF and LSMOF), they speed up the convergence by converting the search into a low-dimensional weight space, but they may perform worse than the other two types of LMOEAs when  $FE_{\max}$  is relatively sufficient (e.g.,  $FE_{\max}=10^5$  at  $n \leq 100$  in E1,  $FE_{\max} = 10^6$  at  $n \leq 500$  in E2, and  $FE_{\max}=n \times 10^4$  in E3). This is because the original variables of the target LMOP in WOF and LSMOF cannot be changed independently, which substantially limits the search to certain optimal original spaces.

Second, the CCF-based LMOEAs (like MOEA/DVA and LMEA) have more advantages in solving LMOPs with a separable  $\mathbf{G}(\mathbf{x}^d)$ , e.g., UF test problems in E3, while the PRM-based LMOEAs (like WOF and LSMOF) are good at solving LMOPs that are more dependent on convergence, e.g., DTLZ in E1.

Third, it is true that LSMOP2 and LSMOP4 are easier to be solved with higher dimensions. As shown in Fig. 5, given the same  $FE_{\max}$  in E2, the mean IGD values of LSMOP2 and LSMOP4 are decreased with the increased value of  $n$ . E1, E2, and E3 are all implemented on the PlatEMO [63], and other parameters involved in the compared LMOEAs are set by the default values on the PlatEMO, which are basically the recommended settings in their corresponding references.

Therefore, to effectively design an LMOEA, balancing the allocation of computational resources between variable analysis and evolutionary optimization is important when handling LMOPs. Moreover, when designing a reproduction strategy, we should not only emphasize its ability to balance exploration and exploitation, but also consider to get an appropriate convergence speed. As motivated by the above observations and discussions, a new LMOEA framework is proposed in Section IV, aiming to achieve a proper balance between the convergence speed and the computational burden when reproducing offspring with higher quality.

### III. THE PROPOSED LMOP TEST SUITES

To further advance the development of research on LMOP, a new LMOP test suite is proposed in this section, called LMF, which follows the basic design principle for MOPs, i.e., test problems are designed by a uniform formulation model, which are both scalable on objectives and variables with known PFs [43]. Moreover, LMF inherits the common features in LSMOP test suite [42] that are specially designed to mimic real-world LMOPs, e.g., nonuniform grouping of variables, different

objective functions on different groups of variables, mixed separability between variables, and variable linkages between  $\mathbf{x}^p$  and  $\mathbf{x}^d$ . Then, LMF makes the following five improvements.

- (1) Introducing a hybrid formulation model.
- (2) Designing hybrid linkages between  $\mathbf{x}^p$  and  $\mathbf{x}^d$ .
- (3) Proposing more flexible grouping of variables.
- (4) Strengthening the shape functions by adding correlations between position-related variables.
- (5) Creating imbalanced contribution of each subcomponent to the corresponding objective function.

### A. Hybrid Formulation and Linkages

In all existing LMOP test suites as reviewed in Section II.A, the uniform formulation for constructing each objective function is either additive or multiplicative. However, different objectives may be constructed with different formula models in real-world MOPs [68]. Therefore, it is essential to include such a realistic characteristic into an LMOP by tailoring each of its objectives with different formula models. To make LMF more realistic, the following formula model with hybrid multiplication and addition is proposed:

**Model 3:** Hybrid multiplication and addition model

$$\min \begin{cases} f_{i=1:2:m}(\mathbf{x}) = h_i(\mathbf{x}^p) \times (1 + g_i(\mathbf{x}^{d,i})) \\ f_{i=2:2:m}(\mathbf{x}) = h_i(\mathbf{x}^p) + g_i(\mathbf{x}^{d,i}) \end{cases} \quad (4)$$

where each objective function  $f_i$  has its own shape function  $h_i$  and landscape function  $g_i$  ( $i = 1, \dots, m$ ). Therefore, by using this hybrid model, mixed position spread of solutions can be obtained, and the functions  $h_1(\mathbf{x}^p), \dots, h_m(\mathbf{x}^p)$  together determine the shape of the PF, which can be used to test the ability of maintaining a diverse population. The functions  $g_1(\mathbf{x}^d), \dots, g_m(\mathbf{x}^d)$  together define the fitness landscape, which can be used to test the ability of converging to the PF.

Moreover, variable linkages between  $\mathbf{x}^p$  and  $\mathbf{x}^d$  are defined either linear or nonlinear in existing LMOPs. Generally, all variables in  $\mathbf{x}^d$  are linked with the first variable  $x_1^p$  of  $\mathbf{x}^p$  with the same way. However, variable linkages in real-world MOPs are always intricate, i.e., variables are closely related to each other in different ways [68]. Thus, to better reflect this realistic characteristic, the following variable linkage function with linear and nonlinear hybridization is proposed:

$$L(\mathbf{x}^{p,i}, \mathbf{x}^{d,i}) = \begin{cases} L_1 : \text{linear linkage} & \text{if } i \bmod 2 = 0 \\ L_2 : \text{nonlinear linkage} & \text{otherwise} \end{cases} \quad (5)$$

where  $L_1$  and  $L_2$  are the linkage functions defined in [42]. Different from [42] that only consider the first parameter  $x_1^p$  in the linkage design, our method in (5) links the variables in the first subgroup  $\mathbf{x}^{p,1}$  to variables in  $\mathbf{x}^{d,i}$  with a hybrid way ( $i = 1, \dots, m$ ). Detailed grouping of variables in  $\mathbf{x}^p$  and  $\mathbf{x}^d$  is elaborated in the next subsection, and the detailed definition of  $L(\mathbf{x}^{p,i}, \mathbf{x}^{d,i})$  is given in the supplementary file of this paper.

### B. Flexible Grouping of Variables

In LMF, for getting a flexible grouping of variables to make the defined objective functions more realistic, three types of variables are defined here, i.e., unique variable ( $u_v$ ), overlapped variable ( $o_v$ ), and shared variable ( $s_v$ ). To be specific,  $u_v$  refers to this variable that can be only found in one group,  $o_v$  means that this variable exists only in two groups, while  $s_v$  can

TABLE II SUMMARY OF THE SIX-BASE SINGLE-OBJECTIVE FUNCTIONS			
Base Functions	Mark	Modality	Separability
Sphere	$bf_1(\mathbf{x})$	Unimodal	Separable
Schwefel 1.2	$bf_2(\mathbf{x})$	Unimodal	Non-separable
Schwefel 2.21	$bf_3(\mathbf{x})$	Unimodal	Non-separable
Ackley	$bf_4(\mathbf{x})$	Multi-modal	Separable
Rastrigin	$bf_5(\mathbf{x})$	Multi-modal	Separable
Rosenbrock	$bf_6(\mathbf{x})$	Multi-modal	Non-separable

TABLE III  
SUMMARY OF THE PROPOSED TWELVE LMF BENCHMARKS

Problem	$\mathbf{H}(\mathbf{x}^p)$	$m$	$g_j(\mathbf{x}^{d,i,j}) = \sum b_j(\mathbf{x}^{d,i,j})$
LMF1	$H_1$ : concave	$\geq 2$	$b_1(\mathbf{x}^{d,i,j}) : [bf_1, bf_2], [bf_1, bf_3]$
LMF2	$H_2$ : convex	$\geq 2$	$b_1(\mathbf{x}^{d,i,j}) : [bf_1, bf_2], [bf_1, bf_4]$
LMF3	$H_3$ : linear	$\geq 2$	$b_1(\mathbf{x}^{d,i,j}) : [bf_3, bf_4], [bf_2, bf_5]$
LMF4	$H_4$ : concave	$\geq 2$	$b_1(\mathbf{x}^{d,i,j}) : [bf_2, bf_1], [bf_1, bf_6]$
LMF5	$H_2$ : convex	$\geq 2$	$b_1(\mathbf{x}^{d,i,j}) : [bf_1, bf_2, bf_3], [bf_1, bf_4, bf_5]$
LMF6	$H_3$ : linear	$\geq 2$	$b_1(\mathbf{x}^{d,i,j}) : [bf_1, bf_2, bf_3], [bf_1, bf_3, bf_5]$
LMF7	$H_1$ : concave	$\geq 2$	$b_1(\mathbf{x}^{d,i,j}) : [bf_1, bf_2, bf_6], [bf_2, bf_3, bf_6]$
LMF8	$H_3$ : linear	$\geq 2$	$b_1(\mathbf{x}^{d,i,j}) : [bf_1, bf_4, bf_6], [bf_3, bf_5, bf_6]$
LMF9	$H_4$ : inverted concave	$\geq 3$	$b_1(\mathbf{x}^{d,i,j}) : [bf_1, bf_2], [bf_2, bf_3], [bf_3, bf_4]$
LMF10	$H_5$ : inverted linear	$\geq 3$	$b_1(\mathbf{x}^{d,i,j}) : [bf_2, bf_4], [bf_2, bf_5], [bf_2, bf_3]$
LMF11	$H_4$ : inverted concave	$\geq 3$	$b_1(\mathbf{x}^{d,i,j}) : [bf_4, bf_5], [bf_1, bf_3], [bf_1, bf_2]$
LMF12	$H_5$ : inverted linear	$\geq 3$	$b_1(\mathbf{x}^{d,i,j}) : [bf_1, bf_6], [bf_3, bf_5], [bf_2, bf_4]$

be found in all groups. Thus, by controlling the number of these three kinds of variables in each position-related group (or distance-related group), the correlation between variables in  $\mathbf{x}^p$  (or in  $\mathbf{x}^d$ ) can be flexibly designed. Specifically, the variables in  $\mathbf{x}^p$  are evenly divided into  $m-1$  groups ( $\mathbf{x}^{p,1}, \dots, \mathbf{x}^{p,m-1}$ ), where each group has three types of position-related variables described above. Details of the grouping of  $\mathbf{x}^p$  are provided in the supplementary file of this paper.

In addition, the variables in  $\mathbf{x}^d$  are nonuniformly decomposed into  $m$  groups ( $\mathbf{x}^{d,1}, \dots, \mathbf{x}^{d,m}$ ) based on the logistic map method introduced in [45], marked as  $LM(c_1, r, N_r)$ , which can generate  $N_r$  chaos-based pseudo random numbers ( $c_1, c_2, \dots, c_{N_r}$ ) by the following formula:

$$c_i = r \times c_{i-1} (1 - c_{i-1}), i > 1 \quad (6)$$

In different independent runs, these  $N_r$  numbers are constant under the same input of  $c_1$  and  $r$ . In this way, a set of  $N_r$  weights ( $w_1, w_2, \dots, w_{N_r}$ ) can be generated below:

$$w_i = \frac{c_i}{\sum_{i=1}^{N_r} c_i} \quad (7)$$

where  $\sum_{i=1}^{N_r} w_i = 1$ . Thus, the  $m$  groups ( $\mathbf{x}^{d,1}, \dots, \mathbf{x}^{d,m}$ ) can be obtained as the following three steps:

**Step 1:** get  $m$  weights based on  $LM(c_1, r, N_r)$  via (6) and (7) with  $N_r = m$ ,  $r = 3.8$ , and  $c_1 = 0.342$ ;

**Step 2:** divide  $\mathbf{x}^d$  into  $m+1$  parts in order, compute the number of variables in the first  $m$  parts by  $N_i = \lfloor w_i \times (l - l_{sv}) \rfloor$  ( $i = 1, 2, \dots, m$ ), and then get the number of variables in the last part as  $l - \sum_{i=1}^m N_i$ , where  $l$  indicates the number of distance variables and  $l_{sv}$  is used to control the number of shared variables.

**Step 3:** variables in the first  $m$  parts in Step 2 are assigned to group  $\mathbf{x}^{d,1}, \dots, \mathbf{x}^{d,m}$ , respectively. Then, the parameter  $\rho_{ov}$  is set to control the proportion of  $o_v$  in each group, followed by distributing it to its neighbor group on the right. Finally, the variables in the last part in Step 2 are regarded as distance

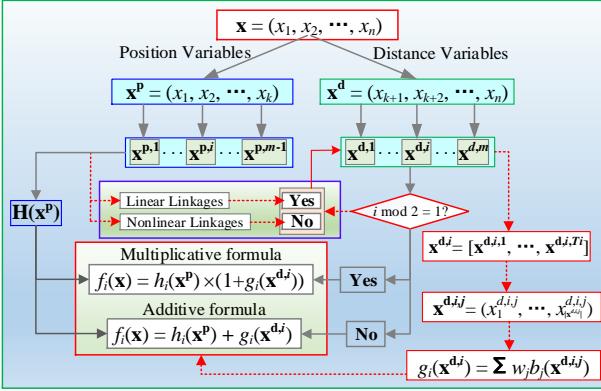


Fig. 6 Illustration of the general process in designing problems of LMF.

related  $s_v$ , which are added into each group  $\mathbf{x}^{d,i}$ .

After that, each  $\mathbf{x}^{d,i}$  is further nonuniformly divided into  $T_i$  subcomponents  $(\mathbf{x}^{d,i,1}, \dots, \mathbf{x}^{d,i,T_i})$ , where  $|\mathbf{x}^{d,i,1}|, \dots, |\mathbf{x}^{d,i,T_i}|$  is an arithmetic-like sequence ( $i = 1, 2, \dots, m$ ). Thus, the nonuniformity of  $|\mathbf{x}^{d,i,1}|, \dots, |\mathbf{x}^{d,i,T_i}|$  can be flexibly controlled by setting its first term and the common difference. More details of the grouping of  $\mathbf{x}^d$  are provided in the supplementary file.

### C. Shape Functions

Generally, the shape of the PF is also an important issue that should be considered in LMOP benchmarks [45]. Until now, a lot of PF shapes have been designed in existing LMOPs, such as the concave, convex, linear, inverted, degenerated, disconnected, and other mixed or irregular shapes. As introduced in Section I, the research of MOEAs focuses on solving the MOPs with irregular PFs. As discussed in [69], the inverted PF shape is more realistic than the regular triangular PF shape when designing MOPs. Moreover, as pointed out in [68], many real-world MOPs are featured with convex PF shapes. However, when designing LMOPs with very complex PFs, the searching of entire PFs may be the key factor to solve these LMOPs, which may decrease the importance of properties in large-scale search space. Considering the above reason, only five basic PF shapes are adopted here, including concave ( $H_1$ ), convex ( $H_2$ ), linear ( $H_3$ ), inverted concave ( $H_4$ ) and inverted linear ( $H_5$ ), which can be found in the MaF test suite [50]. More details about the definition of these five shape functions are provided in the supplementary file of this paper.

### D. Imbalanced Contributions in Landscape Functions

After getting the  $m$  groups  $\mathbf{x}^{d,1}, \dots, \mathbf{x}^{d,m}$ , the landscape function in the  $i$ th objective  $f_i(\mathbf{x})$  can be defined on  $\mathbf{x}^{d,i}$  by the combination of  $T_i$  base single-objective functions, which are respectively defined on its subcomponents  $(\mathbf{x}^{d,i,1}, \dots, \mathbf{x}^{d,i,T_i})$  ( $i = 1, \dots, m$ ). In the existing LMOPs listed in Fig. 1, the same base function is used on all subcomponents with equal size. However, as pointed out in [60], many practical problems can be simulated by combining a set of different base functions, and each of them naturally introduces different contributions to the global objective function. Thus, introducing imbalanced contribution of each subcomponent to the objective function is essential in approximating real-world problems. There are two common ways to create such imbalanced contribution for each

---

### Algorithm 1 General Framework of GLEA

---

**Input:** an LMOP with  $m$  objectives and  $n$  variables

**Output:** the final population  $\mathbf{P}$

```

1: Initialization:  $FE_{\max}$ ,  $FE = 0$ ,  $\mathbf{P}$  with  $N$  random solutions
2: while  $FE \leq FE_{\max}$  do
3:    $\mathbf{Q} = Reproduction\_GLS(\mathbf{P})$ 
4:    $\mathbf{P} = Environmental\_Selection(\mathbf{P}, \mathbf{Q})$ 
5:    $FE += N$ 
6: end while
7: return  $\mathbf{P}$ 

```

---

### Algorithm 2 Reproduction GLS( $\mathbf{P}$ )

---

```

1: initialize the offspring population  $\mathbf{Q} = \emptyset$ 
2:  $(\mathbf{S}_1, \mathbf{S}_2) = Clustering(\mathbf{P})$ 
3: Get  $M$  variable groups  $\mathbf{v}_g = (g^1, \dots, g^M)$  via linear grouping
4:  $\mathbf{Q}_1 = GroupSelfSupervisedLearning(\mathbf{S}_1, \mathbf{v}_g)$ 
5:  $\mathbf{Q}_2 = GroupBroadLearning(\mathbf{S}_1, \mathbf{S}_2, \mathbf{v}_g)$ 
6:  $\mathbf{Q} = \mathbf{Q}_1 + \mathbf{Q}_2$ 
7: return  $\mathbf{Q}$ 

```

---

subcomponent. The first way is to set a non-uniform subcomponent size, which is achieved in  $(\mathbf{x}^{d,i,1}, \dots, \mathbf{x}^{d,i,T_i})$  with an arithmetic-like sequence in Section III.B. The second way is to weight each subcomponent differently as follows:

$$g_i(\mathbf{x}^{d,i}) = \sum_{j=1}^{T_i} w_j b_j(\mathbf{x}^{d,i,j}), \quad (8)$$

where  $b_j(\mathbf{x}^{d,i,j})$  is a base single-objective function defined on  $\mathbf{x}^{d,i,j}$ , and  $w_j$  can be generated with (6) and (7) introduced in Section III.B by setting  $N_r = T_i$  and initializing  $c_1$  and  $r$ . Here,  $c_1 = 0.23$  and  $r = 3.7$  are applied. Besides, six different base single-objective functions are considered in the construction of LMF test suite, which are summarized in Table II, and their corresponding detailed definitions can be found in the supplementary file of this paper.

For ease of understanding, the design process of our LMF test suite is clearly shown in Fig. 6. Finally, twelve instances are designed in LMF, termed LMF1-LMF12, which are summarized in Table III. The function  $b_j(\mathbf{x}^{d,i,j})$  in Table III is selected from the six base functions  $b_1$  to  $b_6$  in Table II based on the value of  $(i, j)$ . Please read the details about this in the supplementary file of this paper.

Table SII of the supplementary file provides a comparison between LMF and seven well-known existing LMOPs, which includes all the types of test suites listed in Fig. 1. In Table SII, the mark  $\otimes$  (“ $\checkmark$ ”) represents the absence (presence) of the corresponding property in each LMOP suite. Besides, the “onefold” means only linear linkage or nonlinear linkage is considered in one problem. As observed from Table SII, LMF embraces a variety of realistic features that are not considered in existing LMOPs. In fact, it is difficult to develop a perfect test suite that can simulate and fully reflect all possible features of those highly complex and diverse real-world LMOPs. The proposed LMF is more able to approximate real-world LMOPs than the existing LMOPs, which is important to fairly analyze the performance of different MOEAs. Finally, both the Matlab and Java source codes of LMF are downloadable at the following website: <https://github.com/songbai-liu/LMF>.

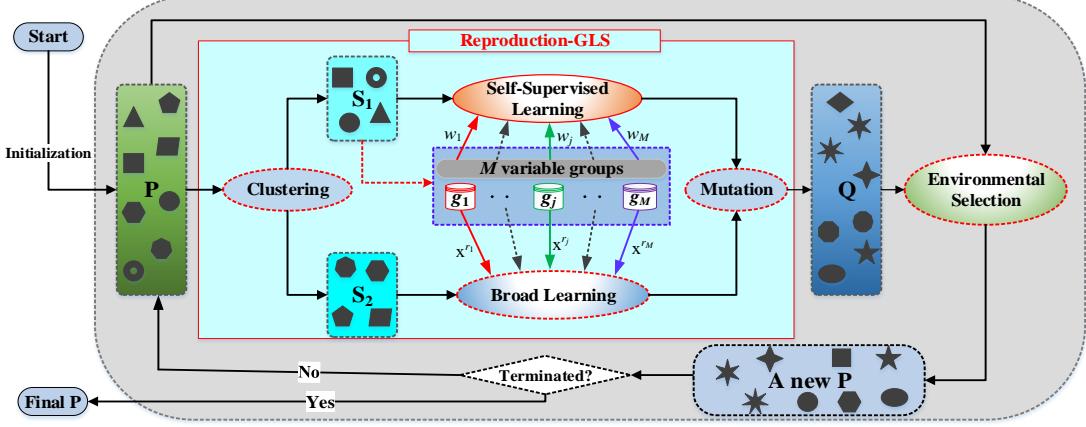


Fig. 7 Illustration of the general flow diagram of the proposed GLEA

#### IV. THE PROPOSED LMOEA FRAMEWORK

To further promote the research of ELMO, a new LMOEA framework is proposed in this section, termed GLEA, which follows the general framework of a traditional MOEA with three components, i.e., initialization, reproduction, and environmental selection, as shown in **Algorithm 1**. In the initialization, a parent population  $\mathbf{P}$  with  $N$  randomly sampled solutions of the target LMOP is obtained, the maximum number of function evaluations  $FE_{\max}$  is given, and the function evaluation counter  $FE$  is set to 0 (line 1). On the reproduction, an offspring population  $\mathbf{Q}$  is generated by searching the variable space via a group-based learning strategy (GLS) on the population  $\mathbf{P}$  (line 3), aiming to balance exploration and exploitation during the evolution. Regarding the environmental selection,  $N$  elite solutions with balanceable convergence and diversity are selected from the combined population of  $\mathbf{P}$  and  $\mathbf{Q}$  via a clustering-based strategy proposed in [64] to update  $\mathbf{P}$  (line 4). From the above description, the main contribution of the proposed GLEA is to improve the reproduction by GLS. As observed from the pseudocode of GLS given in **Algorithm 2**, the GLS-based reproduction includes the following four main components.

**Clustering the population  $\mathbf{P}$**  in line 2: solutions in  $\mathbf{P}$  are classified into two different solution sets ( $\mathbf{S}_1$  and  $\mathbf{S}_2$ ) by a clustering method, where the solutions in  $\mathbf{S}_1$  are considered to be better than that in  $\mathbf{S}_2$  in terms of convergence and diversity. Here, the reference vector-guided clustering proposed in [64] is applied to divide  $\mathbf{P}$ , and the details of the clustering process are provided in the supplementary file.

**Grouping the variables** in line 3: for simplicity, the  $n$  variables of the target LMOP are randomly divided into  $M$  equal groups. Specifically, this random grouping of variables is run in a period of 100 generations during the evolutionary process. Here, the number of evaluations per generation is equal to the population size  $N$ . Besides, let  $\mathbf{v}_g = (\mathbf{g}^1, \dots, \mathbf{g}^M)$  store these  $M$  variable groups.

**Group-based self-supervised learning** in line 4: Since the solutions in  $\mathbf{S}_1$  are considered to be dominant when compared with those in  $\mathbf{S}_2$ , a strategy of self-supervised learning in a low dimensional weighted space is customized for the solutions in  $\mathbf{S}_1$  to effectively generate child solutions, aiming at acceler-

ating the convergence speed. Concretely, each solution  $\mathbf{x}^i \in \mathbf{S}_1$  learns from other two different random solutions  $(\mathbf{x}^{r_1}, \mathbf{x}^{r_2}) \in \mathbf{S}_1$  in a weighted space based on variable groups in  $\mathbf{v}_g$  to generate its child  $\mathbf{x}^c$ , via the following four steps:

**Step 1:** construct a latent weighted space and a weight vector  $\mathbf{w}^i = (\mathbf{w}_1^i, \dots, \mathbf{w}_M^i)$  for  $\mathbf{x}^i$  is obtained as

$$\mathbf{w}_j^i = \frac{1}{|\mathbf{g}^j|} \sum_{k=1}^{|\mathbf{g}^j|} \frac{\mathbf{x}_{g_k^j}^i - LB_{g_k^j}}{UB_{g_k^j} - LB_{g_k^j}} \quad (9)$$

where  $\mathbf{x}^i$  is the  $i$ th solution in  $\mathbf{S}_1$ ,  $g_k^j$  is the  $k$ -th element in  $\mathbf{g}^j$  ( $j \in \{1, 2, \dots, M\}$ ),  $LB$  and  $UB$  are respectively the lower bound and upper bound of the variable space of the target LMOP. Analogously, two weight vectors  $(\mathbf{w}^{r_1}, \mathbf{w}^{r_2})$  corresponding to  $(\mathbf{x}^{r_1}, \mathbf{x}^{r_2})$  are obtained with the same way by (9),  $r_1$  and  $r_2$  are two randomly values selected from  $\{1, 2, \dots, |\mathbf{S}_1|\}$  ( $r_1 \neq r_2 \neq i$ ).

**Step 2:** learn a meta weight vector  $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M)$  in the  $M$ -dimensional weight space, as follows:

$$\mathbf{w}_j = \mathbf{w}_j^i + L \times (\mathbf{w}_j^{r_1} - \mathbf{w}_j^{r_2}) \quad (10)$$

where  $j \in \{1, 2, \dots, M\}$  and  $L$  indicates the learning rate randomly sampled within  $[0, 1]$ .

**Step 3:** adjust the learned weight vector  $\mathbf{w}$  as follows:

$$\mathbf{w}_j = \begin{cases} \alpha \times \mathbf{w}_j^i & \text{if } \mathbf{w}_j < \alpha \times \mathbf{w}_j^i \\ \beta \times \mathbf{w}_j^i & \text{if } \mathbf{w}_j > \beta \times \mathbf{w}_j^i \\ \mathbf{w}_j & \text{otherwise} \end{cases} \quad (11)$$

where  $\alpha$  and  $\beta$  are two parameters used to control the learned scope of  $\mathbf{w}$ , where  $\alpha \in (0, 1)$  and  $\beta > 1$ . In this paper,  $\alpha = 0.3$ ,  $\beta = 3$ , and the sensitivity analysis of  $\alpha$  and  $\beta$  in the proposed GLEA can be found in the supplementary file of this paper.

**Step 4:** produce the child solution  $\mathbf{x}^c$  as follows:

$$\mathbf{x}_t^c = \frac{\mathbf{w}_j \times |\mathbf{g}^j| \times \frac{\mathbf{x}_t^i - LB_t}{UB_t - LB_t}}{\sum_{k=1}^{|\mathbf{g}^j|} \frac{\mathbf{x}_{g_k^j}^i - LB_{g_k^j}}{UB_{g_k^j} - LB_{g_k^j}}} \quad \text{for each } t \in \mathbf{g}^j \quad (12)$$

where  $j \in \{1, 2, \dots, M\}$ . At last, each child solution  $\mathbf{x}^c$  is saved in the offspring population  $\mathbf{Q}$ .

**Group-based broad learning** in line 5: each solution in  $\mathbf{S}_2$  broadly learns from  $M$  solutions randomly selected from  $\mathbf{S}_1$  in the corresponding variable group constrained space, aiming to produce its child with better quality. Particularly, each solution

$\mathbf{y}^i \in \mathbf{S}_2$  produces its child  $\mathbf{y}^c$  as follows:

$$\mathbf{y}_j^c = \mathbf{y}_j^i + \begin{cases} L_1 \times (\mathbf{x}_j^{r1} - \mathbf{y}_j^i) & \text{if } j \in \mathbf{g}^1 \\ L_2 \times (\mathbf{x}_j^{r2} - \mathbf{y}_j^i) & \text{if } j \in \mathbf{g}^2 \\ \dots & \dots \\ L_M \times (\mathbf{x}_j^{rM} - \mathbf{y}_j^i) & \text{if } j \in \mathbf{g}^M \end{cases} \quad (13)$$

where  $\mathbf{x}^{r1}$  to  $\mathbf{x}^{rM}$  are  $M$  randomly selected solutions from  $\mathbf{S}_1$ ,  $L_1$  to  $L_M$  indicate the  $M$  learning rates, and  $j \in \{1, 2, \dots, n\}$ . Please note that  $L_1$  to  $L_M$  are random values sampled within  $[0, 1]$ . Also, each child solution  $\mathbf{y}^c$  is added into  $\mathbf{Q}$ .

In particular, all  $N$  child solutions undergo a polynomial mutation before being added to  $\mathbf{Q}$ , aiming to escape from local optima. In order to give a clear overview of the proposed GLEA, its general flow diagram is depicted in Fig. 7.

## V. EXPERIMENTAL STUDIES

In this section, several experiments are conducted to study the performance of the proposed GLEA and the hardness of the proposed test suite LMF. First, GLEA is compared with five LMOEAs in solving the LMF problems with  $m = (2, 3)$  and  $n = (256, 512, 1024)$ . Then, the LMF is tailored by introducing four adjustable parameters, which can study the sensitivity of all considered LMOEAs to the main features of LMF. Finally, an ablation study of GLEA is conducted to investigate the effectiveness of its main components.

### A. Experimental Design and Parameters Settings.

In this work, the proposed test suite LMF including twelve problems, i.e., LMF1 to LMF12, are used to assess the performance of various LMOEAs. Here, the number of decision variables ( $n$ ) varies from 256 to 1024 for all problems. In addition, the numbers of unique variables, overlapped variables, and shared variables in the position-related grouping are set to 2, 2, and 1, respectively. The parameters  $l_{sv}$  and  $\rho_{ov}$  used in the distance-related grouping are respectively set to 2 and 0.2. Finally, the first term and the common difference of the arithmetic-like sequence in the distance-related grouping are set to 5 and 2, respectively.

To concurrently reflect the convergence and diversity of the final solution sets obtained by the optimizers, the well-known IGD [61] and HV [70] metrics are used as the indicators in the performance comparison, where a smaller (larger) IGD (HV) value indicates a better performance of the optimizer in approximating the true PF of the target LMF problem. When calculating these two indicators, a large number of sample points evenly extracted from the true PF are required for IGD, while a reference point dominated by all points of the true PF is required for HV. In our experiments, 5000 points and  $10^4$  points are sampled to calculate the IGD for problems with 2 and 3 objectives, respectively. Besides, the reference point is specified as  $(1.5, 1.5, \dots, 1.5)$  for the calculation of HV. In the following tables, the mean IGD and HV results are collected from 20 independent runs for each optimizer. In each run, the maximum number of function evaluations ( $FE_{\max}$ ) is set as the termination condition. As pointed out in Section II.C, the performance of an optimizer is sensitive to the value of  $FE_{\max}$ . However, it is common that the computational resource is

limited in practical cases. Thus, the  $FE_{\max}$  in our experiment is set to an acceptable fixed value  $5 \times 10^5$  for all cases. Moreover, the population size ( $N$ ) is set as  $N = 100$  and  $N = 153$  for LMF problems with 2 and 3 objectives, respectively.

For the performance comparisons, six LMOEAs of different types, i.e., MOEA/DVA, WOF, LSMOF, LMOCO, DGEA, are included to compare with our proposed GLEA in the experimental studies. The parameters in these compared LMOEAs are set as suggested in their corresponding references in order to have the best performance and guarantee a fair comparison. To be specific, in MOEA/DVA, the numbers of control property analyses and interaction analyses are set to 20 and 6, respectively. LSMOF is embedded into NSGA-II and configured to use 10 reference vectors and 30 solutions for single-objective optimization. WOF is embedded into MOEA/D and configured to use interval transformation and ordered grouping, along with 1000 evaluations for each optimization of the original problem, 500 evaluations for each optimization of the transformed problem, and 4 variable groups. In DGEA, the direction vector size is set to 10. In our proposed GLEA, the number of groups  $M$  is adaptively set as  $M = 5, 10$ , and 15 when  $n = 256, 512$ , and 1024, respectively.

To ensure a statistically sound conclusion, a Wilcoxon rank sum test with a 0.05 significance level is used to show the statistically significant differences in the performance results. Specifically, the symbols “+”, “-” or “=” after the corresponding IGD results in the following tables indicates that the algorithm respectively performs significantly better, worse, or similarly when compared to GLEA in optimizing the benchmark problems with different numbers of variables.

### B. Results on 2-objective LMF1-LMF8 Problems

Here, GLEA is compared to MOEA/DVA, WOF, LSMOF, LMOCO, and DGEA on LMF1-LMF8 with  $n \in \{256, 512, 1024\}$  and  $m = 2$ , and Table SIII of the supplementary file records the average IGD results of these six optimizers. Clearly, GLEA exhibits significantly better overall performance than its five competitors. Specifically, GLEA achieves the best results in 13 out of 24 test problems, whereas MOEA/DVA, WOF, LSMOF, LMOCO, DGEA perform best in 0, 4, 5, 2, and 0 cases, respectively. Based on the Wilcoxon rank-sum test, when respectively compared to MOEA/DVA, WOF, LSMOF, LMOCO, and DGEA, GLEA is better in 22, 10, 13, 13 and 18 of 24 cases and is worse in 0, 7, 8, 6, and 5 cases, which validates the advantages of GLEA in solving these 2-objective LMF problems, especially for the case of  $n = 1024$ . In spite of this, the IGD values obtained by each optimizer indicate that the proposed LMF suite poses a significant challenge to these LMOEAs, as none of the six optimizers is able to efficiently solve the problems in LMF.

For further analysis, the final non-dominated solutions obtained by the LMOEA with the best IGD value in its best run are plotted in Fig. S9 of the supplementary file when solving 2-objective LMF1 to LMF8 problems with  $n = 256$ . From these eight subfigures in Fig. S9, we can see that the final optimal solutions obtained by these involved LMOEAs are very poor in both diversity and convergence when compared to the true

TABLE IV  
PROPERTY OF THE LANDSCAPE OF LMF1-LMF8 IN ITS TWO OBJECTIVES

Problem	$g_1(\mathbf{x}^{d,1})$	$g_2(\mathbf{x}^{d,2})$
LMF1	Partially separable & Unimodal	Partially separable & Unimodal
LMF2	Separable & Multimodal	Separable & Multimodal
LMF3	Partially separable & Multimodal	Partially separable & Multimodal
LMF4	Partially separable & Unimodal	Partially separable & Multimodal
LMF5	Partially separable & Unimodal	Separable & Multimodal
LMF6	Partially separable & Multimodal	Partially separable & Multimodal
LMF7	Partially separable & Multimodal	Partially separable & Multimodal
LMF8	Partially separable & Multimodal	Partially separable & Multimodal

PFs of LMF1 to LMF8. Thus, we will present some discussions about the difficulties of solving these eight LMF problems below.

In the case of  $m = 2$  in LMF1 to LMF8, there's only one group of position-related variables in them, which means that the position function  $\mathbf{H}(\mathbf{x}^p)$  doesn't make the problem harder in this case. Therefore, the increase of difficulty in these problems mainly comes from the specially designed landscape function  $\mathbf{G}(\mathbf{x}^d)$ , the mixed linkage function  $L(\mathbf{x}^p, \mathbf{x}^d)$ , and the mixed formulation  $\mathbf{F}(\mathbf{x})$  for the two objective functions. Specifically, in LMF1 to LMF8, the first objective  $f_1(\mathbf{x})$  is formulated by the multiplicative model with the linear linkage, while  $f_2(\mathbf{x})$  is constructed by the addition model with nonlinear linkage. Besides,  $g_1(\mathbf{x}^{d,1})$  and  $g_2(\mathbf{x}^{d,2})$  respectively in  $f_1(\mathbf{x})$  and  $f_2(\mathbf{x})$  are completely different but closely correlated, as there are both overlapped and unique variables in  $\mathbf{x}^{d,1}$  and  $\mathbf{x}^{d,2}$ . Learning from Table II and Table III, we can summarize the characteristics of  $g_1(\mathbf{x}^{d,1})$  and  $g_2(\mathbf{x}^{d,2})$  in each problem as shown in Table IV, where each landscape fitness is made up of imbalanced contributions by (8) from base functions with different separability and modality. Among these eight problems, LMF2 seems to be the easiest one to be solved, as its landscapes in both two objectives are separable. Nevertheless, the compared LMOEAs used in our studies still struggle to solve this problem and cannot well solve other more complex LMF problems, which indicates that various degrees of difficulties for different objective functions are introduced in the proposed LMF test suite.

### C. Results on 3-objective LMF1-LMF12 Problems

Again, GLEA is compared to MOEA/DVA, WOF, LSMOF, LMOCso, DGEA on LMF1-LMF12 with  $n \in \{256, 512, 1024\}$  and  $m = 3$ , and Table SIV of the supplementary file records the average IGD results of these six optimizers. Likewise, none of them is able to effectively solve these LMF problems according to their obtained IGD values. Besides, these 3-objective problems bring greater challenges to the involved LMOEAs than the 2-objective cases. Nevertheless, GLEA still shows the best performance, as it obtains significantly better IGD results than its five competitors in most problems. Specifically, GLEA achieves the best results in 24 of 36 test problems, whereas MOEA/DVA, WOF, LSMOF, LMOCso, DGEA perform best in 0, 11, 1, 0, and 0 cases, respectively. Based on the Wilcoxon rank-sum test, when respectively compared to MOEA/DVA, WOF, LSMOF, LMOCso, and DGEA, GLEA is better in 36, 19, 24, 27 and 31 out of 36 cases and is only worse in 0, 9, 5, 2, and 0 cases, which further

demonstrates the superiority of GLEA in solving these 3-objective LMF problems.

Compared to LMF1-LMF8 with  $m = 2$ , they become more complex in the case of  $m = 3$ . At first,  $\mathbf{x}^p$  in  $\mathbf{H}(\mathbf{x}^p)$  is divided into two groups of overlapped variables, thus the diversity of the population is harder to maintain. In addition,  $g_1(\mathbf{x}^{d,1})$ ,  $g_2(\mathbf{x}^{d,2})$ , and  $g_3(\mathbf{x}^{d,3})$  respectively in  $f_1(\mathbf{x})$ ,  $f_2(\mathbf{x})$ ,  $f_3(\mathbf{x})$  are also completely different but closely correlated, which makes the landscape more complicated, and thus it's more difficult to converge towards the PS. Moreover, the same base functions are used to construct  $g_1(\mathbf{x}^{d,1})$  and  $g_3(\mathbf{x}^{d,3})$  in LMF1-LMF8, while this is not the case in LMF9-LMF12, where the base functions used in  $g_1(\mathbf{x}^{d,1})$ ,  $g_2(\mathbf{x}^{d,2})$ , and  $g_3(\mathbf{x}^{d,3})$  are completely different, as summarized in Table III. Furthermore, the inverted PFs are introduced in LMF9 to LMF12. As a result, the features of LMF9-LMF12 become more complex and pose greater challenges to existing LMOEAs. As shown in Fig. S9 of the supplementary file, although GLEA shows the best IGD performance on LMF9-LMF12, its obtained optimal solutions are still far from satisfactory.

Finally, to further study the behavior of all LMOEAs on these 3-objective LMF problems, Fig. S10 of the supplementary file plots their evolutionary curves based on twenty-five average IGD values obtained during the overall optimization process with an interval of 20000 evaluations. Obviously, the performance of MOEA/DVA is the worst one, because it only allocates a small part of computing resources for optimization in this study, which is started at the later stage of the evolution (e.g., on LMF1, LMF3-LMF4, and LMF7-LMF12 problems). For LMOCso and DGEA, their evolution curves soon enter a process of slow convergence, in which LMOCso performs better than DGEA because an acceleration operator is introduced in its competitive search. Regarding WOF and LSMOF, their obtained curves show the fast convergence performance due to reformulation of the target LMOP, but they also quickly converge to stagnation (e.g., on LMF2 to LMF8 problems). Similarly, GLEA converges quickly and then stagnates directly on LMF1-LMF2 and LMF9-LMF10 problems because it also has a self-supervised reformulation of these LMFs.

### D. Further Studies on LMF

#### 1) When Computing Resource is Relatively Sufficient

In Section IV. B, the experimental results reported in Table SIII show that even the 2-objective LMF problems with  $n = 256$  still pose great difficulties for all the LMOEAs involved in the study with  $FE_{max}=5 \times 10^5$ . Here, a relatively sufficient computing resource (i.e.,  $FE_{max}=10^6$ ) is allocated to study the performance of compared LMOEAs in solving simpler LMF problems (i.e.,  $n = 100$ ). Both the average IGD and HV results are presented in Table SV of the supplementary file. It can be seen that the IGD results here have a significant improvement over those in Table SIII. Besides, although GLEA still shows certain advantages on LMF8-LMF12 problems, it is obviously premature and locally convergent on LMF1-LMF3 problems. Thus, we will continue to improve the performance of GLEA in our future work. Moreover, the final population obtained by each LMOEA on LMF1-LMF12 with  $n = 100$  is plotted in Fig.

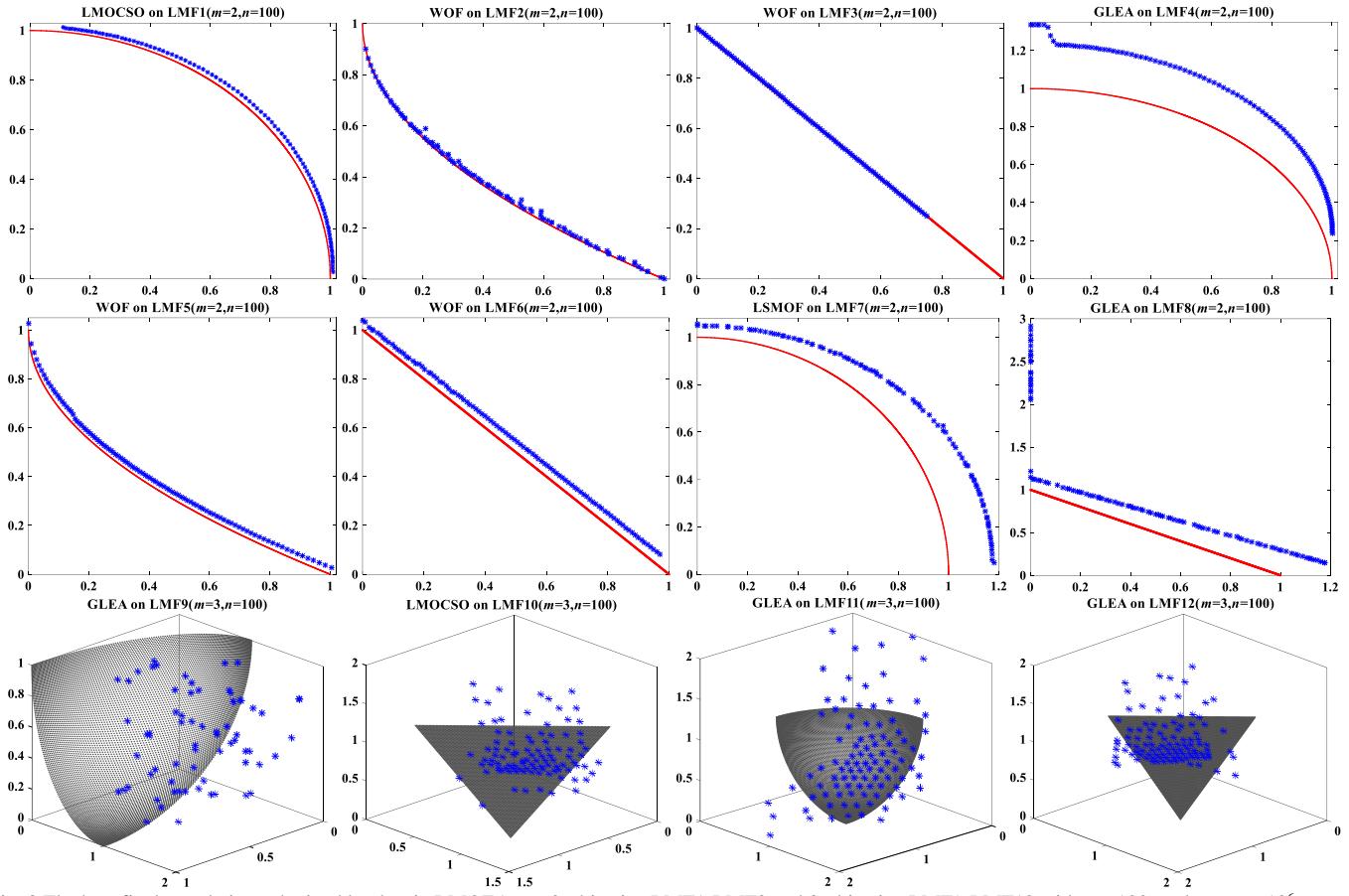


Fig. 8 The best final populations obtained by the six LMOEAs on 2-objective LMF1-LMF8 and 3-objective LMF9-LMF12 with  $n = 100$ , and  $FE_{\max} = 10^6$ .

8. From these subfigures, we can also see that there are still difficulties for these six LMOEAs to solve the 100-D LMF problems when  $FE_{\max}=10^6$ , as the best result obtained by them on each LMF problem is not fully approximate to the true PF.

## 2) Study of the Main Features in LMF

To experimentally study the contributions of the main features in benchmarks, LMF is tailored by further introducing four adjustable parameters  $T_{FM}$ ,  $T_{LK}$ ,  $T_{DG}$ , and  $T_{CV}$ , which are respectively used to control the type of formula model, type of variable linkage, type of deep grouping, and type of component combination in LMF. Concretely,  $T_{FM}$  has three values: 0, 1, and 2, which indicate that LMF is formulated using **Mode 2** by (3), **Mode 1** by (2), and mixed mode by (4), respectively;  $T_{LK}$  also has three values: 0, 1, and 2, which represent that the linkage between  $\mathbf{x}^p$  and  $\mathbf{x}^d$  in LMF is linear, nonlinear, and mixed, respectively. Besides, both  $T_{DG}$  and  $T_{CV}$  are binary numbers, where  $T_{DG} = 0$  and  $T_{CV} = 0$  respectively indicate that  $\mathbf{x}^{d,i}$  is uniformly divided into  $K$  groups and the value of  $w_i$  in (8) is  $1/K$  for LMF ( $K=5$  here as in LSMOP), while  $T_{DG} = 1$  and  $T_{CV} = 1$  respectively represent that the same strategies described in Section III are adopted in LMF for the deep grouping of  $\mathbf{x}^{d,i}$  and the setting of  $w_i$  in (8).

Here, the LMF1-LMF3 problems with  $(m=2, n=100)$  are adopted for the experimental studies, including eight different cases, i.e.,  $(T_{FM}T_{LK}T_{DG}T_{CV}) = \{(0211), (1211), (2211), (2011), (2111), (2201), (2210), (2200)\}$ . Besides, the six LMOEAs introduced in Section IV.A are used to solve these modified

LMF1-LMF3 problems with  $FE_{\max}=10^6$ . The average IGD results are provided in Table S.VI of the supplementary file, which are also plotted in Fig. 9 for a more intuitive effect. As observed from Fig., 9, we can draw the conclusions below.

At first, the corresponding features of LMF1-LMF3 controlled by these four adjustable parameters are of great importance for the performance analysis of LMOEAs, as the IGD results obtained by each LMOEA shown in Fig. 9 are sensitive to the change of a certain characteristic in the LMF problems. Next, for the type of formulation model, using a hybrid model seems to simplify the LMF problem when compared to the other two models, as most LMOEAs perform better when solving LMF1-LMF3 problems in the case of (2211). This may be because such a mixed model can neutralize the extreme effects of only using an addition model or a multiplication model (e.g., the hardly dominated boundaries as reported in [59]). Moreover, for the type of linkages, the mixed type of linkage still does not make the LMF problem more difficult to be solved. Finally, the imbalanced contributions of different components to the objective are jointly controlled by  $T_{DG}$  and  $T_{CV}$ , while these three LMF problems are the most difficult to be solved when setting  $T_{DG}=0$ . The reason behind this may be that  $\mathbf{x}^{d,i}$  is only evenly divided into 5 groups and the number of variables in each group is still large for an LMOP in this case. Thus, the base function defined on this group is hard to be solved when compared to the case of setting  $T_{DG}=1$ . Therefore, we import new features in LMF not to make it more difficult to

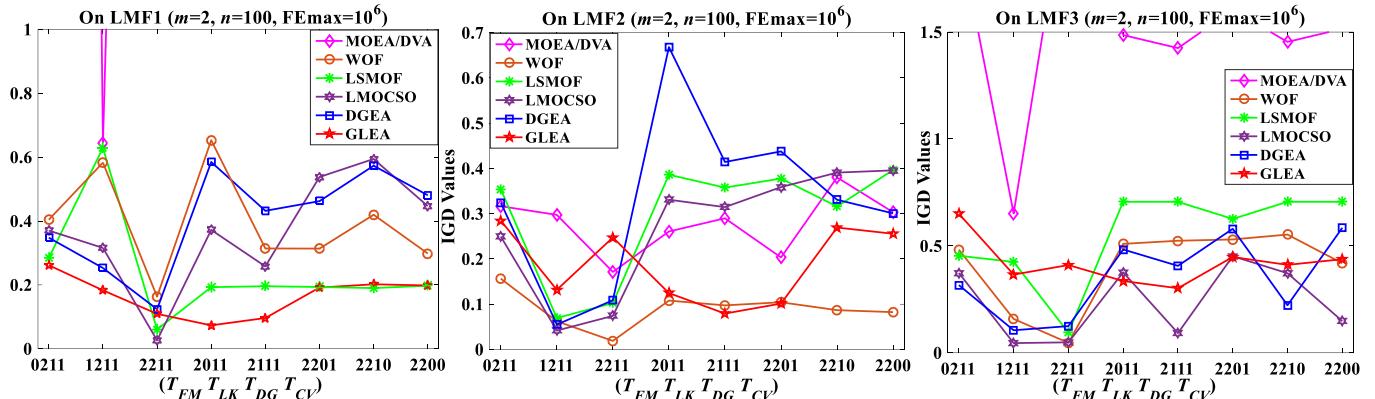


Fig. 9 The average IGD results obtained by six LMOEAs on LMF1-LMF3 problems with eight different settings of parameters ( $T_{FM} T_{LK} T_{DG} T_{CV}$ ).

be solved, but to make the properties of LMF more realistic, because the characteristics in real-world LMOPs are mixed and diversified. Consequently, by properly adjusting these four parameters, the LMF can be tailored into various different test suites to meet different needs.

#### E. Other Experimental Studies of GLEA

Due to page limitation, the following three experimental studies on GLEA are provided in the supplementary file of this paper: 1) comparison study of GLEA and its competitors on the LSMOP test suite in the case of  $FE_{\max} = 10^5$ ,  $m = 2$ ,  $n = (50, 100, 500, 1000)$ ; 2) an ablation study of GLEA in order to investigate the effectiveness of group-based self-supervised learning and group-based broad learning; and 3) parameter sensitivity analysis of GLEA.

## VI. CONCLUSIONS AND FUTURE WORK

To advance the development of ELMO, this paper gives a review of existing LMOPs and LMOEAs to investigate their weakness for ELMO research in terms of benchmarks and algorithms. After that, a new LMOP test suite, termed LMF, and a new LMOEA framework, called GLEA, are respectively developed as inspired from the existing ELMO work. Specifically, LMF is developed by including a variety of features that are rarely considered in literature but frequently exist in real-world problems (e.g., hybrid formulation of objective functions, hybrid linkages between variables, and imbalanced contributions of different variables to the objectives). Further, GLEA is customized by using a variable group-based learning strategy with four main components (i.e., clustering of the population, grouping of the variables, group-based self-supervised learning, and group-based broad learning), which can enhance the effectiveness of reproduction for solving LMOPs. When compared to five competitive LMOEAs, GLEA shows a more promising performance in solving the proposed LMF problems with variable dimensions ranging from 256 to 1024. As observed from the IGD results, we can also conclude that the proposed LMF problems are very challenging to be solved by these LMOEAs.

As observed from the experimental studies, although the proposed GLEA shows some advantages when compared with its competitors, it still has a lot of room for improvement. Thus,

in our future work, we plan to enhance the performance of GLEA from the following three aspects:

- (1) (Improve the grouping of variables) We plan to study a more efficient variable grouping method and to learn an adaptive number of variable groups during the evolutionary process.
- (2) (Improve the group-based self-supervised learning) We plan to construct a more appropriate weighted space and to learn an adaptive parameter setting (e.g., the learning rate in (10) and the  $(\alpha, \beta)$  in (11)).
- (3) (Improve the group-based broad learning) We plan to learn an adaptive learning rate for each group-based subspace and to study how to compare two different solutions in a subspace by a machine learning model, aiming to select  $M$  more eligible solutions to be learned in (13), i.e., each selected solution is relatively prominent in the corresponding subspace rather than in the entire search space of the target LMOP.

Finally, the proposed LMF is still designed based on the bottom-up method. The combinations of position functions and landscape functions in LMF may not fully and exactly formulate the characteristics of various complex real-world applications. Therefore, designing an LMOP test suite based on real-world applications is also in the agenda of our future work.

## REFERENCES

- [1] X. Zhang, et al., “A Network Reduction-Based Multiobjective Evolutionary Algorithm for Community Detection in Large-Scale Complex Networks,” *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 703–716, 2020.
- [2] J. Xiao, T. Zhang, J. Du, X. Zhang, “An Evolutionary Multiobjective Route Grouping-Based Heuristic Algorithm for Large-Scale Capacitated Vehicle Routing Problems,” *IEEE Trans. Cybern.*, early access, 2020.
- [3] Z. Lu *et al.*, “NSGA-NET: A multi-objective genetic algorithm for neural architecture search,” Available: arXiv:1810.03522, 2018.
- [4] S. Liu, Q. Lin, K. Wong, L. Ma, C. A. Coello Coello, D. Gong, “A novel multi-objective evolutionary algorithm with dynamic decomposition strategy,” *Swarm and Evol. Comput.*, vol. 48, pp. 182–200, 2019.
- [5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.
- [6] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, 2014.
- [7] Q. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algo-

- rithm based on decomposition,” *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.
- [8] H. Li and Q. Zhang, “Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 13, pp. 284–302, 2009.
- [9] S. Jiang, J. Zhang, “A Simple and Fast Hypervolume Indicator-Based Multiobjective Evolutionary Algorithm,” *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2202–2213, 2015.
- [10] J. Bader and E. Zitzler, “HypE: an algorithm for fast hypervolume based many-objective optimization,” *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.
- [11] Q. Lin, S. Liu, et al., “Particle Swarm Optimization with a Balanceable Fitness Estimation for Many-Objective Optimization Problems,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 32–46, 2018.
- [12] L. M. Antonio, C. A. Coello Coello, “Use of cooperative coevolution for solving large scale multiobjective optimization problems,” *IEEE Congr. Evol. Comput. (CEC)*, pp. 2758–2765, 2013.
- [13] X. Ma, et al., “A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 275–298, 2016.
- [14] A. Song, Q. Yang, W. Chen, J. Zhang, “A random-based dynamic grouping strategy for large scale multi-objective optimization,” *IEEE Congr. Evol. Comput. (CEC)*, pp. 468–475, 2016.
- [15] B. Cao, J. Zhao, Z. Lv, X. Liu, “A Distributed Parallel Cooperative Coevolutionary Multiobjective Evolutionary Algorithm for Large-Scale Optimization,” *IEEE Trans. Industrial Informatics*, vol. 13, no. 4, pp. 2030–2038, 2017.
- [16] B. Cao, S. Fan, J. Zhao, P. Yang, K. Muhammad, M. Tanveer, “Quantum-enhanced multiobjective large-scale optimization via parallelism,” *Swarm and Evol. Comput.*, vol. 57, 2020.
- [17] X. Zhang, Y. Tian, R. Chen, Y. Jin, “A Decision Variable Clustering-Based Evolutionary Algorithm for Large-Scale Many-Objective Optimization,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 97–112, 2018.
- [18] B. Cao, J. Zhao, Y. Gu, Y. Ling, X. Ma, “Applying graph-based differential grouping for multiobjective large-scale optimization,” *Swarm and Evol. Comput.*, vol. 53, 2020.
- [19] H. Chen, R. Cheng, J. Wen, H. Li, J. Weng, “Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations,” *Inf. Sci.*, vol. 509, pp. 457–469, 2020.
- [20] H. Zille, H. Ishibuchi, S. Mostaghim, Y. Nojima, “A framework for large-scale multiobjective optimization based on problem transformation,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 260–275, 2018.
- [21] C. He, L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin, X. Yao, “Accelerating Large-Scale Multiobjective Optimization via Problem Reformulation,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 949–961, 2019.
- [22] R. Liu, J. Liu, Y. Li, J. Liu, “A random dynamic grouping based weight optimization framework for large-scale multi-objective optimization problems,” *Swarm and Evol. Comput.*, vol. 55, 2020.
- [23] C. He, R. Cheng, Y. Tian, X. Zhang, “Iterated Problem Reformulation for Evolutionary Large-Scale Multiobjective Optimization,” *IEEE Congr. Evol. Comput. (CEC)*, pp. 1–8, 2020.
- [24] D. Tang, Y. Wang, X. Wu, Y. Cheung, “A Symmetric Points Search and Variable Grouping Method for Large-scale Multi-objective Optimization,” *IEEE Congr. Evol. Comput. (CEC)*, pp. 1–8, 2020.
- [25] H. Zille, H. Ishibuchi, S. Mostaghim, Y. Nojima, “Mutation Operators based on Variable Grouping for Multi-objective Large-scale Optimization,” *IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, 2016.
- [26] W. Hong, K. Tang, A. Zhou, H. Ishibuchi, X. Yao, “A Scalable Indicator-Based Evolutionary Algorithm for Large-Scale Multiobjective Optimization,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 525–537, 2019.
- [27] Y. Tian, X. Zheng, X. Zhang, Y. Jin, “Efficient Large-Scale Multi-objective Optimization Based on a Competitive Swarm Optimizer,” *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3696–3708, 2020.
- [28] Y. Zhang, et al., “Enhancing MOEA/D with information feedback models for large-scale many-objective optimization,” *Inf. Sci.*, vol. 522, pp. 1–16, 2020.
- [29] Y. Tian, X. Zhang, C. Wang, Y. Jin, “An Evolutionary Algorithm for Large-Scale Sparse Multi-Objective Optimization Problems,” *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 380–393, 2019.
- [30] C. He, S. Huang, R. Cheng, K. C. Tan, Y. Jin, “Evolutionary Multi-objective Optimization Driven by Generative Adversarial Networks (GANs),” *IEEE Trans. Cybern.*, early access, 2020.
- [31] H. Hiba, A. A. Bidgoli, A. Ibrahim, S. Rahnamayan, “CGDE3: An Efficient Center-based Algorithm for Solving Large-scale Multi-Objective Optimization Problems,” *IEEE Congr. Evol. Comput. (CEC)*, pp. 350–358, 2019.
- [32] L. M. Antonio, et al., “Coevolutionary Operations for Large Scale Multi-Objective Optimization,” *IEEE Congr. Evol. Comput. (CEC)*, pp. 1–8, 2020.
- [33] L. M. Antonio, C. A. Coello Coello, “Decomposition-Based Approach for Solving Large Scale Multi-objective Problems,” *Parallel Problem Solving from Nature (PPSN)*, pp. 525–534, 2016.
- [34] C. He, R. Cheng, D. Yazdani, “Adaptive Offspring Generation for Evolutionary Large-Scale Multiobjective Optimization,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, in press, 2020.
- [35] R. Liu, R. Ren, J. Liu, J. Liu, “A clustering and dimensionality reduction based evolutionary algorithm for large-scale multi-objective problems,” *Applied soft Compt.*, vol. 89, pp. 106–120, 2020.
- [36] Y. Tian, C. Lu, X. Zhang, K. C. Tan, Y. Jin, “Solving Large-Scale Multiobjective Optimization Problems With Sparse Optimal Solutions via Unsupervised Neural Networks,” *IEEE Trans. Cybern.*, early access, 2020.
- [37] Y. Tian, C. Lu, X. Zhang, F. Cheng, Y. Jin, “A Pattern Mining-Based Evolutionary Algorithm for Large-Scale Sparse Multiobjective Optimization Problems,” *IEEE Trans. Cybern.*, early access, 2021.
- [38] E. Zitzler, K. Deb, L. Thiele, “Comparison of Multiobjective Evolutionary Algorithms: Empirical Results,” *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.
- [39] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, “Scalable Test Problems for Evolutionary Multi-Objective Optimization,” *Evolutionary Multiobjective Optimization*, pp. 105–145, 2005.
- [40] S. Huband, P. Hingston, L. Barone, L. While, “A Review of Multi-objective Test Problems and a Scalable Test Problem Toolkit,” *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, 2006.
- [41] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, S. Tiwari, “Multiobjective optimization test instances for the CEC 2009 special session and competition” School of CS & EE, University of Essex, Working Report CES-487, 2009.
- [42] R. Chen, Y. Jin, M. Olhofer, B. Sendhoff, “Test Problems for Large-Scale Multiobjective and Many-Objective Optimization,” *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4108–4121, 2017.
- [43] K. Deb, Multi-objective Genetic Algorithms Problem Difficulties and Construction of Test Problems, *Evol. Comput.*, vol. 7, no. 3, pp. 205–230, 1999.
- [44] Z. Wang, Y. S. Ong, J. Sun, A. Gupta, Q. Zhang, “A Generator for Multiobjective Test Problems with Difficult-to-Approximate Pareto Front Boundaries,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 556–571, 2019.
- [45] S. Z. Martínez, C. A. Coello Coello, H. E. Aguirre, K. Tanaka, “A Review of Features and Limitations of Existing Scalable Multiobjective Test Suites,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 130–142, 2019.
- [46] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, “Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169–190, 2017.
- [47] H. Li, et al., “Comparison between MOEA/D and NSGA-III on a set of novel many and multi-objective benchmark problems with challenging difficulties,” *Swarm Evol. Comput.*, vol. 46, pp. 104–117, 2019.
- [48] R. Wang, R. C. Purshouse, P. J. Fleming, “Preference-inspired coevolutionary algorithms using weight vectors,” *European Journal of Operational Research*, vol. 243, no. 2, pp. 423–441, 2015.
- [49] Y. Tian, R. Cheng, X. Zhang, M. Li, Y. Jin, “Diversity Assessment of Multi-Objective Evolutionary Algorithms: Performance Metric and Benchmark Problems,” *IEEE Comput. Intell. Mag.*, vol. 14, no. 3, pp. 61–74, 2019.
- [50] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, X. Yao, “A benchmark test suite for evolutionary many-objective optimization,” *Complex and Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.
- [51] K. Deb, A. Sinha, S. Kukkonen, “Multi-objective test problems, linkages, and evolutionary methodologies,” *Genetic and Evol. Comput. Conf. (GECCO)*, Seattle, WA, pp. 1141–1148, 2006.
- [52] H. Li, Q. Zhang, “A multi-objective differential evolution based on

- decomposition for multiobjective optimization with variable linkages,” in Proc. *Parallel Problem Solving From Nature* (PPSN IX), Reykjavik, Iceland, pp. 583–592, 2006.
- [53] Q. Zhang, A. Zhou, Y. Jin, “RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 41–63, 2008.
- [54] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, “A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 838–856, Dec. 2015.
- [55] H. Liu, F. Gu, Q. Zhang, “Decomposition of a Multiobjective Optimization Problem into a Number of Simple Multiobjective Subproblems,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, 2014.
- [56] H. Liu, L. Chen, K. Deb, E. D. Goodman, “Investigating the Effect of Imbalance Between Convergence and Diversity in Evolutionary Multiobjective Algorithms,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 408–425, 2017.
- [57] D. K. Saxena, Q. Zhang, J. A. Duro, A. Tiwari, “Framework for Many-objective Test Problems with both Simple and Complicated Pareto-set Shapes,” *Evolutionary Multi-Criterion Optimization (EMO)*, pp. 197–211, 2011.
- [58] H. Li, Q. Zhang, J. Deng, “Biased Multiobjective Optimization and Decomposition Algorithm,” *IEEE Trans. Evol. Comput.*, vol. 47, no. 1, pp. 52–66, 2017.
- [59] Z. Wang, Y. S. Ong, H. Ishibuchi, “On Scalable Multiobjective Test Problems with Hardly-dominated Boundaries,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 217–231, 2019.
- [60] M. N. Omidvar, X. Li, T. Tang, “Designing benchmark problems for large-scale continuous optimization,” *Inf. Sci.*, vol. 316, pp. 419–436, 2015.
- [61] P. A. N. Bosman and D. Thierens, “The balance between proximity and diversity in multiobjective evolutionary algorithms,” *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, 2003.
- [62] J. Alcala-Fdez, et al., “KEEL: a software tool to assess evolutionary algorithms for data mining problems,” *Soft Computing*, vol. 13, pp. 307–318, 2009.
- [63] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, “PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum],” *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017.
- [64] Y. Yuan, H. Xu, B. Wang, and X. Yao, “A new dominance relation based evolutionary algorithm for many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16–37, 2016.
- [65] C. He, R. Cheng, C. Zhang, Y. Tian, Q. Chen, X. Yao, “Evolutionary Large-Scale Multiobjective Optimization for Ratio Error Estimation of Voltage Transformers,” *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 868–881, 2020.
- [66] H. Ishibuchi, M. Yamane, N. Akedo, Y. Nojima, “Many-Objective and Many-Variable Test Problems for Visual Examination of Multiobjective Search,” *IEEE Congr. Evol. Comput. (CEC)*, pp. 1491–1498, 2013.
- [67] H. Ishibuchi, Y. Peng, K. Shang, “A Scalable Multimodal Multiobjective Test Problem,” *IEEE Congr. Evol. Comput.*, pp. 310–317, 2019.
- [68] R. Tanabe, H. Ishibuchi, “An easy-to-use real-world multi-objective optimization problem suite,” *Applied Soft Computing*, vol. 89, 2020.
- [69] H. Ishibuchi, L. He, K. Shang, “Regular Pareto Front Shape is not Realistic,” *IEEE Congr. Evol. Comput. (CEC)*, pp. 2034–2041, 2019.
- [70] L. While, L. Bradstreet and L. Barone, “A Fast Way of Calculating Exact Hypervolumes,” *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 86–95, 2012.



**Songbai Liu** received the B.S. degree from Changsha University and the M.S. degree from Shenzhen University, China, in 2012 and 2018, respectively. He worked for ShenZhen TVT Digital Technology Co., Ltd as a software engineer from 2013 to 2015, and he worked for Shenzhen University as a research assistance from 2018 to 2019.

He is currently a PhD student in Department of Computer Science, City University of Hong Kong, Hong Kong. His current research interests include nature-inspired computation, evolutionary transfer optimization, and evolutionary large-scale optimization.



**Qiužhen Lin** received the B.S. degree from Zhaoqing University and the M.S. degree from Shenzhen University, China, in 2007 and 2010, respectively. He received the Ph.D. degree from Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, in 2014. He is currently an assistant professor in College of Computer Science and Software Engineering, Shenzhen University. He has published over sixty research papers since 2008. His current research interests include artificial immune system, multi-objective optimization, and dynamic system.

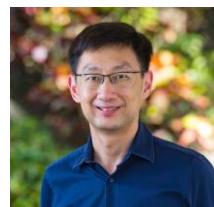


**Ka-Chun Wong** received his B.Eng. in Computer Engineering from The Chinese University of Hong Kong in 2008. He has also received his M.Phil. degree at the same university in 2010. He received his PhD degree from the Department of Computer Science, University of Toronto in 2015. After that, he assumed his duty as assistant professor at City University of Hong Kong. His research interests include Bioinformatics, Computational Biology, Evolutionary Computation, Data Mining, Machine Learning, and Interdisciplinary Research.



**Qing Li** received the B.Eng. degree in computer science from Hunan University, Changsha, China, and the M.Sc. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles, CA, USA.

Prof. Li has been a chair professor of data science, the Hong Kong Polytechnic University since December 2018. Prior to that, he taught at the City University of Hong Kong, the Hong Kong University of Science and Technology, and Australian National University. He is currently a fellow of IET (formerly IEE), a senior member of the IEEE, and a member of ACM-SIGMOD and the IEEE Technical Committee on Data Engineering. He is the chairperson of the Hong Kong Web Society, and also served/is serving as an executive committee (EXCO) member of the IEEE-Hong Kong Computer Chapter and ACM Hong Kong Chapter



**Kay Chen Tan** (Fellow, IEEE) received the B.Eng. degree (First Class Hons.) in electronics and electrical engineering and the Ph.D. degree from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively. He is currently a Chair Professor of the Department of Computing at the Hong Kong Polytechnic University, Hong Kong. He has published over 200 refereed articles and six books, and holds one U.S. patent on surface defect detection.

Prof. Tan is currently the Vice-President (Publications) of IEEE Computational Intelligence Society, USA. He has served as the Editor-in-Chief of IEEE Transactions on Evolutionary Computation from 2015–2020 and IEEE Computational Intelligence Magazine from 2010–2013, and currently serves as the Editorial Board Member of over 10 journals. He is currently an IEEE Distinguished Lecturer Program (DLP) speaker and Chief Co-Editor of Springer Book Series on Machine Learning: Foundations, Methodologies, and Applications.