Enhanced Multifactorial Evolutionary Algorithm With Meme Helper-Tasks

Xiaoliang Ma, *Member, IEEE*, Jian Yin, Anmin Zhu, Xiaodong Li, *Fellow, IEEE*, Yanan Yu, Lei Wang, Yutao Qi, and Zexuan Zhu, *Senior Member, IEEE*

***Abstract*—Evolutionary multitasking (EMT) is an emerging research direction in the field of evolutionary computation. EMT solves multiple optimization tasks simultaneously using evolutionary algorithms with the aim to improve the solution for each task via intertask knowledge transfer. The effectiveness of intertask knowledge transfer is the key to the success of EMT. The multifactorial evolutionary algorithm (MFEA) represents one of the most widely used implementation paradigms of EMT. However, it tends to suffer from noneffective or even negative knowledge transfer. To address this issue and improve the performance of MFEA, we incorporate a prior-knowledge-based multiobjectivization via decomposition (MVD) into MFEA to construct strongly related meme helper-tasks. In the proposed method, MVD creates a related multiobjective optimization problem for each component task based on the corresponding problem structure or decision variable grouping to enhance positive intertask knowledge transfer. MVD can reduce the number of local optima and increase population diversity. Comparative experiments on the widely used test problems demonstrate that the constructed meme helper-tasks can utilize the prior knowledge of the target problems to improve the performance of MFEA.**

***Index Terms*—Evolutionary multitasking (EMT), helper-task, knowledge transfer, multifactorial evolutionary algorithm (MFEA), multiobjectivization, multiobjectivization via decomposition (MVD).**

Manuscript received June 15, 2020; revised October 16, 2020; accepted January 3, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61976143, Grant 61871272, Grant 61603259, Grant 61303119, and Grant 61772392; in part by the Natural Science Foundation of Guangdong Province under Grant 2019A1515010869 and Grant 2020A151501946; in part by the Shenzhen Fundamental Research Program under Grant JCYJ20190808173617147; in part by the Science Basic Research Plan in Shaanxi Province of China under Grant 2018JM6009; in part by the Scientific Research Foundation of Shenzhen University for Newly-recruited Teachers under Grant 85304/00000247; and in part by the Zhejiang Labs International Talent Fund for Young Professionals. This article was recommended by Associate Editor C. F. Juang. *(Corresponding author: Zexuan Zhu.)*

Xiaoliang Ma, Jian Yin, Anmin Zhu, and Yanan Yu are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: maxiaoliang@szu.edu.cn; yinjian0316@163.com; azhu@szu.edu.cn; 1810272005@email.szu.edu.cn).

Xiaodong Li is with the School of Science (Computer Science and Software Engineering), RMIT University, Melbourne, VIC 3001 Australia (e-mail: xiaodong.li@rmit.edu.au).

Lei Wang is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: lei.wang1@siat.ac.cn).

Yutao Qi is with the School of Computer Science and Technology, Xidian University, Xi'an 710071, China (e-mail: ytqi@xidian.edu.cn).

Zexuan Zhu is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, also with Shenzhen Pengcheng Laboratory, Shenzhen 518055, China, and also with the SZU Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen University, Shenzhen 518060, China (e-mail: zhuzx@szu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3050516.

Digital Object Identifier 10.1109/TCYB.2021.3050516

NOMENCLATURE

η_i	Scalar fitness assigned to the ith solution.
$\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T$	Multiobjectivization of $f(\mathbf{x})$.
$\mathbf{G}(\mathbf{x}) = [g_1(\mathbf{x}), \ldots, g_l(\mathbf{x})]^T$	Multiobjectivization of $g(\mathbf{x})$.
$\mathbf{H}(\mathbf{x}) = [f(\mathbf{x}), g(\mathbf{x})]^T$	Biobjective optimization problem.
$\mathbf{x}$	Decision vector.
τ_i	Skill factor of the ith solution.
$f(\mathbf{x}), g(\mathbf{x})$	Two single-objective functions.
$f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})$	m Subfunctions of $f(\mathbf{x})$, s.t. $f(\mathbf{x}) = \sum_{i=1}^{m} f_i(\mathbf{x})$.
$g_1(\mathbf{x}), \ldots, g_l(\mathbf{x})$	l Subfunctions of $g(\mathbf{x})$, s.t. $g(\mathbf{x}) = \sum_{i=1}^{l} g_i(\mathbf{x})$.
N	Population size.
$N(\mu, \sigma^2)$	Normal probability density function.
P	Evolutionary population.
r_{ij}	Factorial rank of the ith solution in the jth task.

I. INTRODUCTION

EVOLUTIONARY algorithms (EAs) are optimization metaheuristics inspired by Darwinian principles of survival of the fittest [1]. They begin with a population of individual solutions and undergo reproduction and/or mutation to generate offspring. In each iteration, the fitter offspring individuals are selected for the next iteration, which ultimately leads to a population of the best possible solutions to a given task. EAs have been successfully applied to a variety of optimization problems, thanks to their efficiency and ease of implementation [1]–[3].

Traditionally, EAs are developed for single-tasking optimization [3]–[5]. Evolutionary multitasking (EMT) [6], also known as multitasking optimization (MTO), is a new paradigm aiming at solving multiple optimization tasks simultaneously by transferring data or knowledge acquired from the solving process of one task to facilitate solving another task [7]. The multifactorial evolutionary algorithm (MFEA) [6] is one of the most widely used EMT algorithms, where each component task contributes a unique factor influencing the evolution of a population of individuals. MFEA allows intertask knowledge transfer via assortative mating and vertical cultural transmission. Many studies [8]–[17] have demonstrated that with the help of positive intertask knowledge transfer, MFEAs can achieve better performance than the corresponding counterpart single-task EAs. However, they tend to suffer from the issue of noneffective or even negative knowledge transfer, especially in the case of uncorrelated or weakly correlated tasks [7].

Nevertheless, if the prior knowledge of the target problems is available, it is desirable to leverage this prior knowledge to enhance knowledge transfer. A feasible way to achieve this is to construct helper-tasks based on prior knowledge to enhance the effectiveness of knowledge transfer. Currently, it is still an open issue about how to construct the strongly related helper-tasks to help solve the original target task(s). Multiobjectivization via decomposition (MVD) is a candidate solution. MVD was initially designed to transform a single-objective optimization problem (SOP) into a multiobjective optimization problem (MOP) such that the original problem can be better solved with a multiobjective solver [18]. This transformation is usually performed by adding helper-objective(s) into the original objective or decomposing the original objective into multiple subobjectives [19], [20].

The method proposed in this article (hereinafter referred to as MFEA/MVD) integrates prior-knowledge-based MVD with MFEA to construct strongly related meme helper-tasks and increases the intertask correlation, thus improving the performance of MFEA. The prior knowledge in memetic computing is referred to as meme [21]. A meme helper-task in the proposed method is created for each original task based on the problem structure or decision variable grouping. This created meme helper-task based on MVD is strongly related to the original task and can increase the intertask correlation when the original and helper tasks are optimized simultaneously. Moreover, a helper-task is a multiobjectivization of the original task, that is, it naturally has the advantages of multiobjectivization, such as the reduction of local optima and increase of incomparable solution plateaus [18]–[20], [22]–[25]. Experimental studies on benchmark problems demonstrate that MVD is a promising way of constructing strongly related tasks, and integrating the prior knowledge into MFEA can substantially improve its performance. The main contributions of this study can be summarized as follows.

1) Prior knowledge-based MVD is introduced in this article as a simple yet effective strategy for constructing strongly related helper-tasks.
2) Six different solving paradigms are compared and investigated, where two of them are proposed for the first time in this article. These paradigms offer different perspectives to study the same problem.
3) Existing prior-knowledge-based MVD methods are studied mainly on combinatorial optimization problems [18]–[20], [22]–[25], but rarely on continuous optimization problems. This study fills this gap by investigating the effectiveness and efficiency of MVD in continuous optimization.

The remainder of this article is organized as follows. Section II introduces the related background. Section III describes the proposed algorithm. Section IV presents the test problems, the prior-knowledge-based helper-task, and the experimental results. Finally, Section V concludes this article.

II. Background

To facilitate the understanding of the proposed method, this section provides some background on MOP, EMT, multiobjectivization, and memetic algorithms, followed by the motivation of combining MVD with MFEA.

A. MOP

A MOP [5], [26]–[30] can be represented as follows:

$$\begin{cases} \min \mathbf{O}(\mathbf{x}) = [o_1(\mathbf{x}), \ldots, o_m(\mathbf{x})]^T \\ \text{subject to} : \mathbf{x} \in \Omega \end{cases} \tag{1}$$

where $\mathbf{x}$ denotes a decision vector, Ω denotes the feasible decision space, $\mathbf{O}(\mathbf{x}) : \mathbf{R}^n \to \mathbf{R}^m$ consists of m objective functions, and n denotes the number of decision variables. Let $\mathbf{x}_a$ and $\mathbf{x}_b$ be two decision vectors. Solution $\mathbf{x}_a$ is said to dominate solution $\mathbf{x}_b$ (denoted as $\mathbf{x}_a \prec \mathbf{x}_b$), if and only if $\mathbf{O}(\mathbf{x_a}) \neq \mathbf{O}(\mathbf{x_b})$ and $\forall i \in \{1, \ldots, m\}, o_i(\mathbf{x}_a) \leq o_i(\mathbf{x}_b)$. A solution $\mathbf{x}^* \in \Omega$ is called a Pareto-optimal solution if it is not dominated by any other feasible solution. All Pareto-optimal solutions are defined as the Pareto-optimal set (PS), that is, $\text{PS} = \{\mathbf{x}^* | \nexists \mathbf{x} \in \Omega, \mathbf{x} \prec \mathbf{x}^*\}$. The objective vectors of PS are known as the PF, that is, $\text{PF} = \{\mathbf{O}(\mathbf{x}) | \mathbf{x} \in \text{PS}\}$.

B. MTO

MTO solves multiple optimization tasks simultaneously in the EA framework to improve the performance of solving each task independently via intertask knowledge transfer. A MTO problem can be defined as

$$\{\mathbf{x}_1^*, \ldots, \mathbf{x}_K^*\} = \underset{(\mathbf{x}_1, \ldots, \mathbf{x}_K)}{\arg\min} \{I_1(\mathbf{x}_1), \ldots, I_K(\mathbf{x}_K)\} \tag{2}$$

where $I_j(\mathbf{x}) : \Omega_j \to \mathbf{R}$ $(\Omega_j \subseteq \mathbf{R}^{n_j})$ denotes the jth optimization task T_j, K denotes the number of optimization tasks, and $\mathbf{x}_j^* \in \Omega_j \subseteq \mathbf{R}^{n_j}$ denotes the optimal solution of $I_j(\mathbf{x}_j)$. Unlike single-tasking optimization, MTO calls for new individual fitness assignment and comparison criteria. Accordingly, the following set of new properties for MTO was proposed in [6].

1) *Factorial Rank:* The factorial rank r_{ij} of an individual p_i on task T_j is the index of p_i in the individual population that is sorted in ascending order in terms of the performance on task T_j.
2) *Skill Factor:* The skill factor τ_i of p_i is one of the component tasks, where p_i achieves the best rank among all tasks, that is

$$\tau_i = \underset{1 \leq j \leq K}{\arg\min} \{r_{ij}\}. \tag{3}$$

3) *Scalar Fitness:* The scalar fitness η_i of p_i is defined as $\eta_i = \min_{1 \leq j \leq K} \{1/r_{ij}\}$.

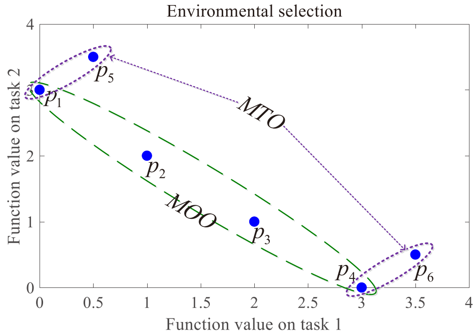


Fig. 1. Sample points in the combined objective space of two optimization tasks.

Both multiobjective optimization (MOO) and MTO handle multiple optimization functions/tasks simultaneously, yet they have some differences that mainly lie in the following.

1) The functions/tasks in MOO are defined in the same decision space, whereas the functions/tasks in MTO can have different decision spaces. Typically, MOO attempts to find a set of representative solutions over PF [4], whereas MTO aims at finding the global optimum of at least one constitutive optimization task utilizing intertask knowledge transfer.
2) The preferences of individuals in MOO and MTO are also different. Fig. 1 takes an example to show their differences in selecting four elite solutions from six alternatives. If nondominated sorting [4] is used for MOO, solutions p_1, p_2, p_3, and p_4 belonging to the first nondominated front are selected as the elite solutions. For MTO, solutions $\{p_1, p_5\}$ and solutions $\{p_4, p_6\}$ are chosen as the elite solutions of task 1 and task 2, respectively.

C. Multifactorial Evolutionary Algorithm and Its Variants

MFEA [6] is a representative EMTO algorithm. Its basic procedure is similar to that of the majority of EAs except for the following five core parts: 1) unified representation; 2) assortative mating; 3) vertical cultural transmission; 4) selective evaluation; and 5) rank-based individual fitness assignment introduced in (3).

Quite a few studies have been proposed to improve the performance and scalability of MFEA. Many proposed MFEAs assume that the source task and the target task are related to each other in some sense. However, if the assumption does not hold, negative transfer among tasks may happen, which may cause MFEAs perform worse than the corresponding counterpart single-task EAs with no transferring at all. To address the issue of negative knowledge transfer, the study [9] introduced a denoising autoencoder to transfer knowledge by learning a linear transformation from the subpopulation generated for one task to the subpopulation generated for another task. Similarly, LDA-MFEA [31] incorporated a linearized domain adaptation (LDA) strategy into MFEA via transforming the search space from one task to another task. Furthermore, they proposed MFEA-II [32], which updates the random mating probability *rmp* adaptively based on online learning and intertask similarities. Unlike MFEA-II, the study [33] reset the parameter *rmp* by calculating the accumulated survival rate of divergents, which are the offspring generated by two parents with different skill factors. In the method presented in [34], decision variable translation and shuffling strategies were used to solve MTO problems whose optima are of different locations and dimensions, respectively.

To improve the scalability of MFEA, a multiobjective version of MFEA (MO-MFEA) [8] was proposed for multiobjective MTO, where the fitness of each individual is estimated based on nondominated rank and crowding distance. The method presented in [35] employed a permutation-based unified individual representation and a level-based survivor selection to handle permutation-based combinatorial optimization. Other methods integrated different metaheuristic algorithms, including differential evolution (DE) [36], memetic algorithm [37], opposition-based learning [15], one-dimension search [38], and particle swarm optimization (PSO) [39].

MFEAs have achieved good results in many real-world applications, including cloud computing service composition [13], vehicle routing problem [14], annealing production process [40], and the composites manufacturing industry [41].

TABLE I
MVD AND MULTIOBJECTIVIZATION VIA HELPER-OBJECTIVE ON $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x}) + f_3(\mathbf{x})$

Method	Alternative multi-objectivization problems
Multi-objectivization via helper-objective	$[f(\mathbf{x}), f_1(\mathbf{x})]^T$, $[f(\mathbf{x}), f_2(\mathbf{x})]^T$, $[f(\mathbf{x}), f_3(\mathbf{x})]^T$, $[f(\mathbf{x}), f_1(\mathbf{x})+f_2(\mathbf{x})]^T$, $[f(\mathbf{x}), f_1(\mathbf{x})+f_3(\mathbf{x})]^T$ $[f(\mathbf{x}), f_2(\mathbf{x})+f_3(\mathbf{x})]^T$, and $[f(\mathbf{x}), f_1(\mathbf{x}), f_2(\mathbf{x})]^T$
Multi-objectivization via decomposition	$[f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})]^T$ (complete decomposition), $[f_1(\mathbf{x}), f_2(\mathbf{x})+f_3(\mathbf{x})]^T$, $[f_2(\mathbf{x}), f_1(\mathbf{x})+f_3(\mathbf{x})]^T$, $[f_3(\mathbf{x}), f_1(\mathbf{x})+f_2(\mathbf{x})]^T$, and $[f_1(\mathbf{x}), f_2(\mathbf{x}) * f_3(\mathbf{x})]$

D. Multiobjectivization

Multiobjectivization was initially designed to reformulate a SOP into a MOP to find a solution to the SOP [18]. Multiobjectivization can be divided into two main types [20]: 1) adding helper-objectives to the original objective [19], [22] and 2) decomposing the original objective into multiple subobjectives optimized simultaneously [18], [22]. A sum-of-parts function $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x}) + f_3(\mathbf{x})$ is taken to illustrate the principle of multiobjectivization in Table I.

Multiobjectivization via helper-objectives is the most commonly used multiobjectivization method, which optimizes the original objective and helper-objectives simultaneously [42]. Introducing helper-objectives can reduce the selection pressure toward the original objective [43] and help the search move through an inferior region of the original objective to avoid local optima. Moreover, helper-objectives allow incorporating domain knowledge and preferences into the search. Let $f(\mathbf{x})$ be the original objective. Common helper-objectives include $1/f(\mathbf{x})$ [44], $f(\mathbf{x})+\varepsilon$ [45], $f(\mathbf{x}+\varepsilon)$ [45], $[(\|\nabla f(\mathbf{x})\|_1)/(n)]$ [46], $\|\nabla f(\mathbf{x})\|_2$ [46], subobjective [47], [48], a variation of the original objective [49]–[51], solution age [44], distance to the closest neighbor [52], and average distance to all individuals [53]. Adding a small number of helper-objectives was suggested by Jensen [19] to help optimize the original objective.

The term "multiobjectivization" via decomposition was first coined by Knowles *et al.* [18], where a problem is divided into smaller subproblems solved simultaneously to find optimal solutions of the original problem. These subproblems are required to cover all parts of the original problem, so that optimizing subproblems simultaneously could solve the original problem [52]. The most used MVD strategies include transforming $f_1(\mathbf{x})+\cdots+f_m(\mathbf{x})$ into $[f_1(\mathbf{x}),\ldots,f_m(\mathbf{x})]^T$ [22], dividing $f_1(\mathbf{x})+\lambda * f_2(\mathbf{x})$ into $[f_1(\mathbf{x}),f_2(\mathbf{x})]^T$ [54], and converting $f_1(\mathbf{x})/f_2(\mathbf{x})$ into $[f_1(\mathbf{x}),-f_2(\mathbf{x})]^T$ [55]. The general MVD strategy divides a SOP $g(f_1(\mathbf{x}),\ldots,f_m(\mathbf{x}))$ into $[f_1(\mathbf{x}),\ldots,f_m(\mathbf{x})]^T$. Early MVD methods are based on static decomposition, which may bias the search. If there are multiple alternative decompositions, a smart switch of these decompositions may contribute to the search [19], [20], [23]. Furthermore, several theoretical studies have been reported for MVD [20], [56]. For example, Lochtefeld and Ciarallo [22] proved the relationship between MVD and helper-objective formalization. Handl *et al.* [56] explained how MVD reduces the number of local optima. They also proved that MVD increases the size of incomparable solutions, which is one of the reasons why MVD can make the problem easier or harder [18], [22], [56], [57].

Existing MVD methods are designed mainly for combinatorial optimization and relatively scarce for continuous optimization. The study [58] presented the idea of multiobjectivization for generating helper-tasks in multitasking, but in the limited context of combinatorial optimization. The first paper applying MVD for continuous optimization is [59], where a deceptive problem could be solved more easily by using the MVD method. Another study along this direction is [60], which multiobjectivized the learning problem $f_1(\mathbf{x})+\lambda * f_2(\mathbf{x})$ of sparse features into $[f_1(\mathbf{x}),f_2(\mathbf{x})]^T$ to handle the difficulty in specifying the value of parameter λ. The localization problem of the underwater sensors was decomposed into a MOP based on triaxial magnetometers in [61] to minimize each error function.

E. Memetic Algorithms

Similar to a gene in genetics, a meme in memetics can be described as a building block of replicable and transmissible cultural knowledge. Typically, a meme has been perceived as a form of individual learning or local search to enhance the ability of population-based heuristics. Memetic algorithms represent a form of synergistic combination between local refinement and population-based search [21]. The performance of a population-based heuristic can be improved by incorporating domain knowledge-based refinement techniques. For example, Louis and McDonnell [62] proposed a case-injected genetic algorithm that periodically injects high-quality solutions obtained from solving other similar problems to accelerate the search. Furthermore, structured knowledge, captured from previous experiences of problem solving, can be used to generate good solutions for unseen problems in the same domain [63]. The parallel memetic algorithms presented in [64] combined one global and two local search methods. Feng *et al.* [65] proposed a multiagent system based on memetic automation toward human-like social agents. The probabilistic memetic framework presented in [66] estimates the theoretical upper bound on the computational budget for the local search intensity and frequency along with determining solutions to be included into local search.

In this article, we construct strongly related helper-tasks to assist the optimization of the original tasks. Each helper-task is constructed using prior knowledge-based MVD. In contrast to existing memetic algorithms, a meme in the proposed method is defined as a useful information unit acquired from the optimization of a helper-task. The primary goal of this study is to investigate whether prior knowledge-based MVD is a good way of generating helper-tasks and whether knowledge learned from MVD-based helper-tasks can enhance the performance of MFEA.

F. Motivation of Combining MFEA With MVD

This section discusses the advantages of MVD over single-objective optimization (SOO) and the motivation behind combining MFEA with MVD.

1) Advantages of MVD Over SOO: We first analyze what MVD can bring to SOO. Previous studies demonstrated that MVD can help avoid local optima and maintain good population diversity. In the following, we exemplify these advantages and make deeper analysis based on a simple continuous function:

$$\begin{aligned} f(\mathbf{x}) &= f_1(\mathbf{x}) + f_2(\mathbf{x}) \\ f_1(\mathbf{x}) &= -N(\mu = 1, \sigma = 0.8) = -1/\sqrt{2\pi * 0.8} * e^{-\frac{(x-1)^2}{2*0.8^2}} \\ f_2(\mathbf{x}) &= -N(\mu = 5, \sigma = 2) = -1/\sqrt{2\pi * 2} * e^{-\frac{(x-5)^2}{2*2^2}} \end{aligned} \tag{4}$$

where $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$ is an opposite Gaussian mixture density function. Equation (4) is taken here as an example, thanks to its good fitting ability, statistical properties, and wide applications.

In an EA framework, let the combined population be $P_t \cup O_t = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ as shown in Fig. 2 and $N = 4$ elite solutions be selected into the next generation.

1) If the optimization problem is the SOP $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$, the selected solutions into the next generation could be $\{x_3, x_4, x_5, x_6\}$, which tend to converge to the local optimum x_4 as shown in Fig. 2(a).
2) When the optimization problem is the MOP $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]^T$, the solutions selected into the next generation could be $\{x_1, x_2, x_3, x_4\}$, whose members do not dominate each other as shown in Fig. 2(b) and are closer to the global optimum x_b of $f(x)$ as shown in Fig. 2(a).

Compared to the original SOO, MVD impacts the comparability relation among solutions and increases the number of incomparable solutions [18], [22], [57]. Based on the above example, the advantages of MVD can be summarized as follows.

1) MVD can improve population diversity and obtain a better solution set than SOO in approximating the global optimum of $f(\mathbf{x})$ [19], [22], [57].

TABLE II
ENVIRONMENTAL SELECTION FOR SIX SOLVING PARADIGMS ON THE EXAMPLE SHOWN IN FIGS. 3 AND 5, WHERE FOUR ELITE SOLUTIONS ARE SELECTED FROM TEN CANDIDATE SOLUTIONS

	Problem	Algorithm	Selected solutions in environment choice	Properties of the environment choice	Involved figure(s)
Optimize $f(\mathbf{x})$	1.$f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$	SOEA [6]	$\{x_3, x_4, x_5, x_6\}$ showed in Fig. (3)	Converge to a local optimal solution	Fig. 3 (a)
	2.$\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]^T$	NSGA-II [4]	$\{x_1, x_2, x_3, x_4\}$ showed in Fig. (3)	Closer to the global optimal solution	Fig. 3 (b)
	3.Problem (5)	MFEA/MVD	$\{x_1, x_3, x_4, x_5\}$ showed in Fig. (3)	Better population diversity and closer to the global optimum	Fig. 3 (a)-(b)
Optimize $f(\mathbf{x})$ and $g(\mathbf{x})$ simultaneously	4.$\mathbf{H}(\mathbf{x}) = [f(\mathbf{x}), g(\mathbf{x})]^T$	NSGA-II [4]	$\{x_4, x_5, x_8, x_9\}$ showed in Fig. (6)	Trade-off between $f(\mathbf{x})$ and $g(\mathbf{x})$	Fig. 5 (e) and Fig. (6)
	5.Problem (6)	MFEA [6]	$\{x_4, x_5, x_8, x_9\}$ showed in Fig. (6)	Converge to the local optimal solution of each task	Fig. 5 (a) and (c) and Fig. (6)
	6.Problem (7)	MFEA/MVD	$\{x_1, x_4, x_9, x_{10}\}$ showed in Fig. (6)	Better diversity and closer to the global optimum of each task	Fig. 5 (a)-(d) and Fig. (6)

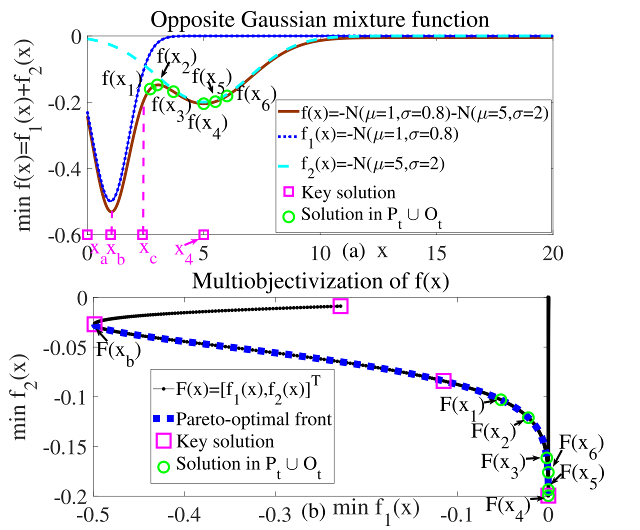


Fig. 2. MVD instance of (4), where $f_1(\mathbf{x}) = -N(\mu = 1, \sigma = 0.8)$ and $f_2(\mathbf{x}) = -N(\mu = 5, \sigma = 2)$. (a) Landscape of $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$. (b) Landscape of $\mathbf{F}(x) = [f_1(\mathbf{x}), f_2(\mathbf{x})]^T$.

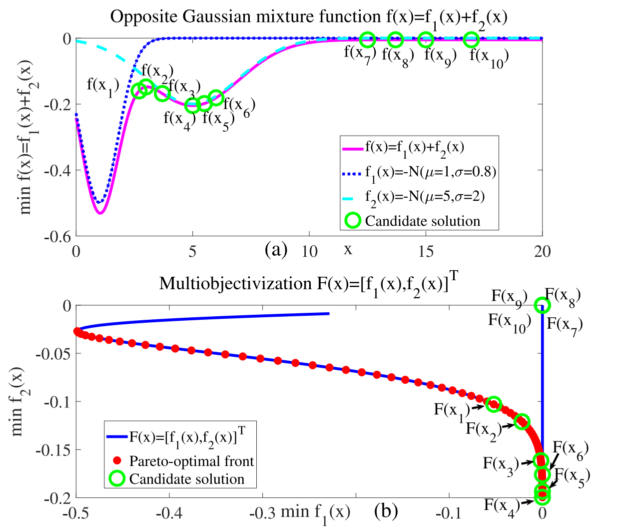


Fig. 3. Comparison of environmental selection in three solving paradigms based on (4). (a) SOP $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$ and (b) its MVD problem $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]^T$.

2) MVD can reduce the number of local optima [56], that is, it ensures that the number of local PSs in $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]^T$ is less than or equal to the number of local optima in $f(\mathbf{x})$.
3) Using the relaxed notion of nondomination, MVD introduces new neutral paths from local optima to the global optima [e.g., a path from x_4 to x_b in Fig. 2(b)] that are not available by solving the original SOP [18].

2) Motivation Behind Combining MFEA With MVD: The motivation can be analyzed from two parts: 1) single-tasking optimization and 2) MTO. In single-tasking optimization, we compare three solving paradigms: 1) single-objective evolutionary algorithm (SOEA) for optimizing $f(\mathbf{x}) = f_1(\mathbf{x}) + \cdots + f_m(\mathbf{x})$; 2) NSGA-II [4] for optimizing $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T$; and 3) the proposed MFEA/MVD for optimizing two tasks $f(\mathbf{x})$ and $\mathbf{F}(\mathbf{x})$ simultaneously, that is

$$\text{MTO}\begin{cases}\text{Task 1: SOO } f(\mathbf{x}) = f_1(\mathbf{x}) + \cdots + f_m(\mathbf{x}) \\ \text{Task 2: MOO } \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T\end{cases} \quad (5)$$

where $\mathbf{F}(\mathbf{x})$ is a prior-knowledge-based helper-task of $f(\mathbf{x})$. Fig. 3 takes the example in (4) to show the differences in the environmental selection. Table II summarizes the selected solutions and properties of the compared solving paradigms.

1) On optimizing $f(\mathbf{x})$, SOEA selects four elite solutions $\{x_3, x_4, x_5, x_6\}$, which tend to converge on a locally optimal solution x_4 as shown in Fig. 3(a).
2) When optimizing $\mathbf{F}(\mathbf{x})$, NSGA-II chooses four elite solutions $\{x_1, x_2, x_3, x_4\}$, which are of a lower nondominated rank with respect to $\mathbf{F}(\mathbf{x})$ as shown in Fig. 3(b). Solutions $\{x_1, x_2, x_3, x_4\}$ selected by NSGA-II achieve a better diversity than those selected by SOEA.
3) In comparison, MFEA/MVD (optimizing $f(\mathbf{x})$ and $\mathbf{F}(\mathbf{x})$ simultaneously in (5)) selects four elite solutions $\{x_1, x_3, x_4, x_5\}$ since they are the best solutions of task $f(\mathbf{x})$ or $\mathbf{F}(\mathbf{x})$ as illustrated in Fig. 3. According to the above results, MFEA/MVD selects better solutions $\{x_1, x_3, x_4, x_5\}$ than NSGA-II and SOEA in keeping population diversity and approximating the global optimal solution of $f(\mathbf{x})$, respectively.

In MTO, we compare MFEA with and without MVD to demonstrate the motivation of employing MVD to enhance the performance of MFEA. In particular, MFEA without MVD is used to optimize two tasks $f(\mathbf{x})$ and $g(\mathbf{x})$ simultaneously in (6), while the proposed MFEA/MVD is used to optimize four tasks $f(\mathbf{x})$, $\mathbf{F}(\mathbf{x})$, $g(\mathbf{x})$, and $\mathbf{G}(\mathbf{x})$ simultaneously in (7)

$$\text{MTO}\begin{cases}\text{Task 1: SOO } f(\mathbf{x}) \\ \text{Task 2: SOO } g(\mathbf{x})\end{cases} \quad (6)$$

Optimization problem	Category	Solved algorithm	Population size	Feature
1.SOO $f(\mathbf{x}) = f_1(\mathbf{x}) + \cdots + f_m(\mathbf{x})$	Optimize $f(\mathbf{x})$ only	SOEA	N for $f(\mathbf{x})$	Single-task optimization
2.MOO $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T$		NSGA-II	N for $\mathbf{F}(\mathbf{x})$	Single-task optimization
3. $MTO \begin{cases} Task\ 1: SOO\ f(\mathbf{x}) = f_1(\mathbf{x}) + \cdots + f_m(\mathbf{x}) \\ Task\ 2: MOO\ \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T \end{cases}$ (5)		MFEA/MVD	$N/2$ *for* $f(\mathbf{x})$, $N/2$ *for* $\mathbf{F}(\mathbf{x})$	Multi-task optimization
4.MOO $\mathbf{H}(\mathbf{x}) = [f(\mathbf{x}), g(\mathbf{x})]^T$	Optimize $f(\mathbf{x})$ and $g(\mathbf{x})$ simultaneously	NSGA-II	N for $\mathbf{H}(\mathbf{x})$	Single-task optimization
5. $MTO \begin{cases} Task\ 1: SOO\ f(\mathbf{x}) \\ Task\ 2: SOO\ g(\mathbf{x}) \end{cases}$ (6)		MFEA	$N/2$ *for* $f(\mathbf{x})$, $N/2$ *for* $g(\mathbf{x})$	Multi-task optimization
6. $MTO \begin{cases} Task\ 1: SOO\ f(\mathbf{x}) = f_1(\mathbf{x}) + \cdots + f_m(\mathbf{x}) \\ Task\ 2: SOO\ g(\mathbf{x}) = g_1(\mathbf{x}) + \cdots + g_l(\mathbf{x}) \\ Task\ 3: MOO\ \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T \\ Task\ 4: MOO\ \mathbf{G}(\mathbf{x}) = [g_1(\mathbf{x}), \ldots, g_l(\mathbf{x})]^T \end{cases}$ (7)		MFEA/MVD	$N/4$ *for* $f(\mathbf{x})$, $N/4$ *for* $g(\mathbf{x})$, $N/4$ *for* $\mathbf{F}(\mathbf{x})$, $N/4$ *for* $\mathbf{G}(\mathbf{x})$	Multi-task optimization

Fig. 4. Six solving paradigms compared and investigated in this work for $f(\mathbf{x})$, where $g(\mathbf{x})$ is an assistant optimization task/problem with unknown correlation to $f(\mathbf{x})$.

Algorithm 1 Framework of the Proposed MFEA/MVD

Require: N: the population size.
Ensure: P: the evolved population.
1: $P \leftarrow$ RandomInitialize(N).
2: Evaluate each individual of P on all optimization tasks.
3: Compute the skill factor of each individual in P.
4: **While** the termination criterion not fulfilled **do**
5: $O \leftarrow$ Crossover & mutation (P, N) - Algorithm 2.
6: Evaluate each individual of O on a selected task only.
7: $P \leftarrow$ Assignment & selection($P \cup O$, N) - Algorithm 3.
8: **End while**
9: Return P.

Algorithm 2 Crossover and Mutation Operators (Assortative Mating and Vertical Cultural Transmission)

Require: p_1, p_2: Two random selected population members. rmp: Random mating probability.
Ensure: c_1, c_2: Two generated offspring.
1: Record skill factors of p_1 and p_2 as τ_{p_1} and τ_{p_2}, respectively.
2: **If** ($\tau_{p_1} == \tau_{p_2}$ or $rand(0, 1) < rmp$)
3: (c_1,c_2)=crossover(p_1, p_2).
4: **If** $rand < 0.5$
5: c_1 and c_2 inherit the skill factors of p_1 and p_2, respectively.
6: **Else**
7: c_1 and c_2 inherit the skill factors of p_2 and p_1, respectively.
8: **End if**
9: **Else**
10: c_1=mutation(p_1), c_2=mutation(p_2).
11: c_1 and c_2 inherit the skill factors of p_1 and p_2, respectively.
12: **End If**

where $g(\mathbf{x})$ has an unknown correlation with $f(\mathbf{x})$

$$\text{MTO} \begin{cases} \text{Task 1: SOO } f(\mathbf{x}) = f_1(\mathbf{x}) + \cdots + f_m(\mathbf{x}) \\ \text{Task 2: SOO } g(\mathbf{x}) = g_1(\mathbf{x}) + \cdots + g_l(\mathbf{x}) \\ \text{Task 3: MOO } \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T \\ \text{Task 4: MOO } \mathbf{G}(\mathbf{x}) = [g_1(\mathbf{x}), \ldots, g_l(\mathbf{x})]^T \end{cases} \tag{7}$$

where $\mathbf{F}(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$ are prior knowledge-based helper-tasks of $f(\mathbf{x})$ and $g(\mathbf{x})$, respectively.

Whether optimization tasks are correlated or not plays an important role in the effectiveness of intertask knowledge transfer. If tasks are uncorrelated or weakly correlated, noneffective or even negative knowledge transfers may occur [7], preventing MFEAs from optimizing each task. However, when the prior knowledge of the target problems is available, it is desirable to use this prior knowledge to facilitate the optimization of the problems. A feasible way of utilizing the prior knowledge is to use MVD to construct helper-tasks for improving the effectiveness of knowledge transfer. Without loss of generality, we consider bi task optimization as an example to illustrate the constructed MVD task. When (6) optimizes two tasks $f(\mathbf{x})$ and $g(\mathbf{x})$ simultaneously by MFEAs, (7) adds their prior-knowledge-based MVD tasks $\mathbf{F}(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$ as helper-tasks, respectively. Since $\mathbf{F}(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$ in (7) are, respectively, related to tasks $f(\mathbf{x})$ and $g(\mathbf{x})$, the intertask correlation in (7) is stronger than that in (6), which is expected to lead to the proposed MFEA/MVD outperforming MFEA. A summary of the aforementioned six solving paradigms is presented in Fig. 4.

III. Proposed Method

In the proposed MFEA/MVD method, prior knowledge-based MVD is incorporated into MFEA (MFEA/MVD) to construct meme helper-tasks for improving the intertask correlation and performance of MFEA. First, MFEA/MVD creates a prior knowledge-based MOP $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T$ as a meme helper-task for each optimization task $f(\mathbf{x}) = f_1(\mathbf{x}) + \cdots + f_m(\mathbf{x})$ based on the problem structure or decision variable grouping. As such, the resulting helper-task $\mathbf{F}(\mathbf{x})$ is strongly related to the task $f(\mathbf{x})$ and hence, can increase the intertask correlation when the original and helper tasks are optimized simultaneously. Moreover, the helper-task $\mathbf{F}(\mathbf{x})$ is a multiobjectivization of the original task $f(\mathbf{x})$. It also can exploit the advantages of multiobjectivization, that is, reduction of the number of local optima and the increase of population diversity. Finally, the proposed MFEA/MVD method optimizes the original tasks of MFEA and their helper-tasks simultaneously with the expectation of obtaining better performance compared to MFEA. The following sections introduce the framework of the proposed method, MVD strategies, fitness assignment, and environmental selection.

A. Framework of the Proposed Method

The pseudocode of MFEA/MVD is presented in Algorithm 1. The main procedure is similar to that of MFEA. First, a population P comprising N individuals is initialized randomly and evaluated on all component tasks at the beginning. The individual encoding is unified with random-key representation, that is, all individuals are encoded into a unified search space. The skill factor of each individual in P is computed in line 3 based on (3).

During the evolution, MFEA/MVD repeatedly performs crossover, mutation, evaluation, fitness assignment, and selection in lines 4–8. The processes of crossover and mutation are presented in Algorithm 2. If two randomly selected parent individuals have the same skill factor, they perform crossover to reproduce offspring. If the two parent individuals are with different skill factors, they still can mate at a random mating probability (*rmp*) to enhance the exploration ability, otherwise, only mutation is performed. Each offspring inherits the skill factor from one of the parent randomly to imitate vertical cultural transmission [10]. In line 6, selective evaluation is performed to reduce the resource consumption, that is, each newly generated offspring is evaluated on a single task associated with its skill factor.

In MFEA/MVD, two key modifications are applied to MFEA: 1) the prior knowledge-based MVD is incorporated into MFEA to generate strongly related helper-tasks to increase intertask correlation and improve the performance of MFEA and 2) a new fitness assignment and environmental selection is used. These two modifications are detailed in the following two sections, respectively.

B. Meme Helper-Tasks Generated by the Prior Knowledge-Based MVD Strategy

Memes are units of domain-specific information, which is useful for problem solving. Here, the problem structure or decision variable grouping is applied to construct a meme MVD helper-task based on prior knowledge for each component task of MFEA. To facilitate understanding, the following simple example in (8) is provided to demonstrate two MVD helper-task generation strategies:

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} e^{x_i} \tag{8}$$

where n is the number of decision variables.

The majority of existing MVD strategies are based on the problem structure [18], [20], [22], [57]. If an optimization task is defined as $g(f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, its meme helper-task based on the problem structure can be represented as a MOP $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T$, e.g., the MOP $\left[\sum_{i=1}^{n} x_i^2, \sum_{i=1}^{n} e^{x_i}\right]^T$ can be used as a helper-task of (8).

If the problem structure is unclear, a decision-variable grouping-based problem decomposition can be used as an alternative. In particular, if an optimization task is defined as $h(f_1(\mathbf{x}_1), \ldots, f_m(\mathbf{x}_m))$, the meme helper-task based on decision variable grouping can be represented as a MOP $[f_1(\mathbf{x}_1), \ldots, f_m(\mathbf{x}_m)]^T$, where $\mathbf{x}_1, \ldots, \mathbf{x}_m$ are m subsets of decision variables $\{x_1, \ldots, x_n\}$. In (8), the meme helper-task based on decision variable grouping can be defined as $\left[\sum_{i=1}^{\lfloor n/2 \rfloor}(x_i^2 + e^{x_i}), \sum_{i=\lfloor n/2 \rfloor+1}^{n}(x_i^2 + e^{x_i})\right]^T$.

Note that as a pioneer work, two simple yet effective MVD strategies are considered in this study. The focus is on the verification of the effects of introducing MVD to MFEA. For more complex problems, basis function decomposition and elementary landscape decomposition [67] can be used as alternative strategies. A more general MVD strategy is desirable for black-box problems.

Algorithm 3 Fitness Assignment and Environmental Selection

Require: N: the population size.
P: the current population.
O: the offspring population.
Ensure: P: the current population.
1: $S \leftarrow P \cup O$.
2: Sort individuals in S based on each task.
3: Update the skill factors of all individuals in S based on (3).
4: Update the scalar fitness of all individuals in S.
5: $P \leftarrow$ Select N fittest individuals from S based on their scalar
fitness on the original tasks and their helper-tasks.

C. Fitness Assignment and Environmental Selection

The pseudocode of fitness assignment and environmental selection is shown in Algorithm 3. After the evaluation of offspring individuals, all parent and offspring individuals are sorted for each task (line 2), where nondominated ranking (NR) and crowding distances (CD) [4] are used to evaluate the individual fitness. The skill factor τ_i of an individual p_i is the task that p_i performs best among all tasks (line 3). Finally, the elite strategy is used to select the top N individuals into the next generation (line 5). In particular, MFEA/MVD selects the best individuals from the rest of candidate individuals based on original tasks and their helper-tasks sequentially.

The following simple example defined in (9) is considered to study the effects of introducing MVD into MFEA in the environmental selection, that is:

$$\begin{aligned}
f_1(x) &= -N(\mu = 1, \sigma = 0.8)\\
f_2(x) &= -N(\mu = 5, \sigma = 2)\\
f(x) &= f_1(x) + f_2(x) = -N(\mu = 1, \sigma = 0.8)\\
&\quad - N(\mu = 5, \sigma = 2)\\
&= -\frac{1}{\sqrt{2\pi * 0.8}} * e^{-\frac{(x-1)^2}{2*0.8^2}} - \frac{1}{\sqrt{2\pi * 2}} * e^{-\frac{(x-5)^2}{2*2^2}}\\
g_1(x) &= -N(\mu = 15, \sigma = 1.5)\\
g_2(x) &= -N(\mu = 19, \sigma = 1)\\
g(x) &= g_1(x) + g_2(x) = -N(\mu = 15, \sigma = 1.5)\\
&\quad - - N(\mu = 19, \sigma = 1)\\
&= -\frac{1}{\sqrt{2\pi * 1.5}} * e^{-\frac{(x-15)^2}{2*1.5^2}} - \frac{1}{\sqrt{2\pi}} * e^{-\frac{(x-19)^2}{2}}
\end{aligned} \tag{9}$$

where $f(\mathbf{x})$ and $g(\mathbf{x})$ are the opposite Gaussian mixture density functions of two local optima as shown in Fig. 5. The detailed processes of environmental selection are shown in Fig. 6. Table II further illustrates the selected solutions and summarizes the properties of the solving paradigms.

To demonstrate the advantages of incorporating MVD into MFEA, the performance in environmental selection of the following methods is compared: NSGA-II for MOP $[f(\mathbf{x}), g(\mathbf{x})]^T$, MFEA for (6), and MFEA/MVD for (7). Four elite solutions from ten alternative solutions are selected in each case.

1) On solving MOP $[f(\mathbf{x}), g(\mathbf{x})]^T$, NSGA-II selects solutions $\{x_4, x_5, x_8, x_9\}$ because they have lower nondominated ranks as shown in Fig. 5(e).

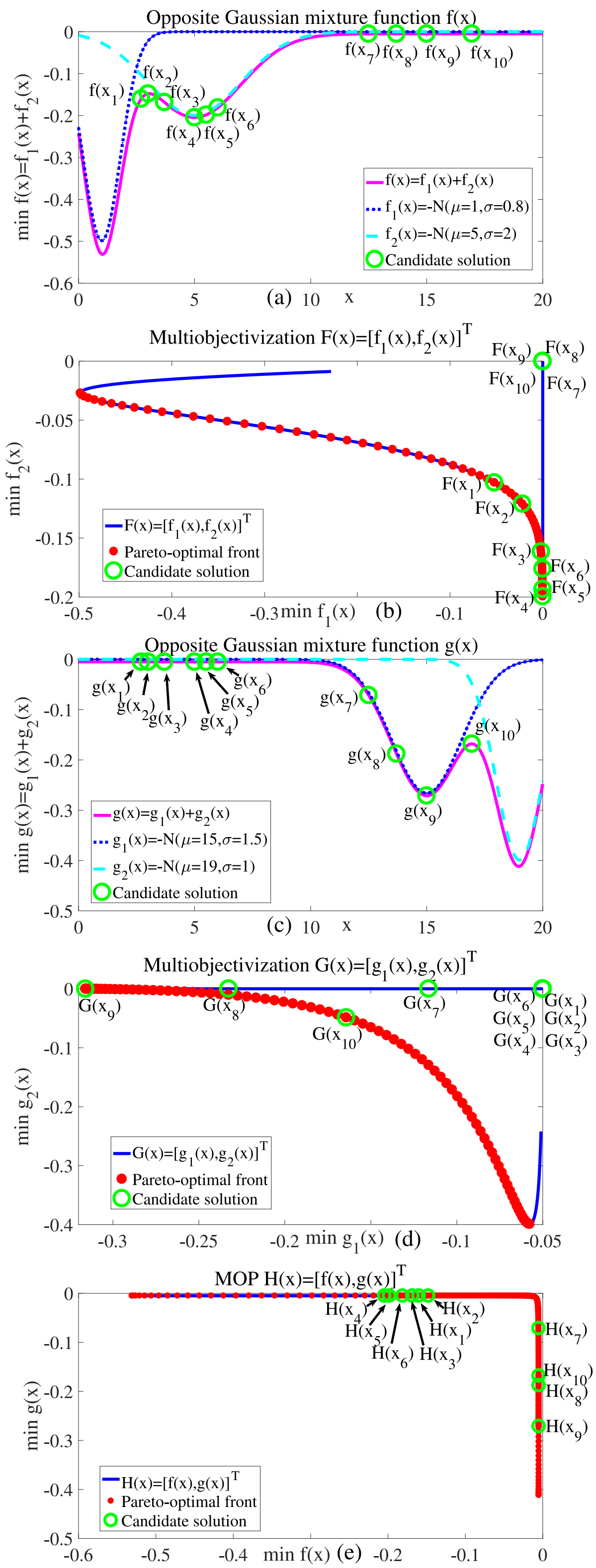


Fig. 5. Plot of the environmental selection for six solving paradigms on the example (9). (a) SOP $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$. (b) Helper-MOP $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]^T$. (c) SOP $g(\mathbf{x}) = g_1(\mathbf{x}) + g_2(\mathbf{x})$. (d) Helper-MOP $\mathbf{G}(\mathbf{x}) = [g_1(\mathbf{x}), g_2(\mathbf{x})]^T$. (e) MOP $\mathbf{H}(\mathbf{x}) = [f(\mathbf{x}), g(\mathbf{x})]^T$.

2) On solving (6), MFEA also selects solutions $\{x_4, x_5, x_8, x_9\}$ since they are the elite solutions of the original tasks $f(\mathbf{x})$ and $g(\mathbf{x})$ as illustrated in Fig. 5(a) and (c). However, the elite solutions $\{x_4, x_5, x_8, x_9\}$

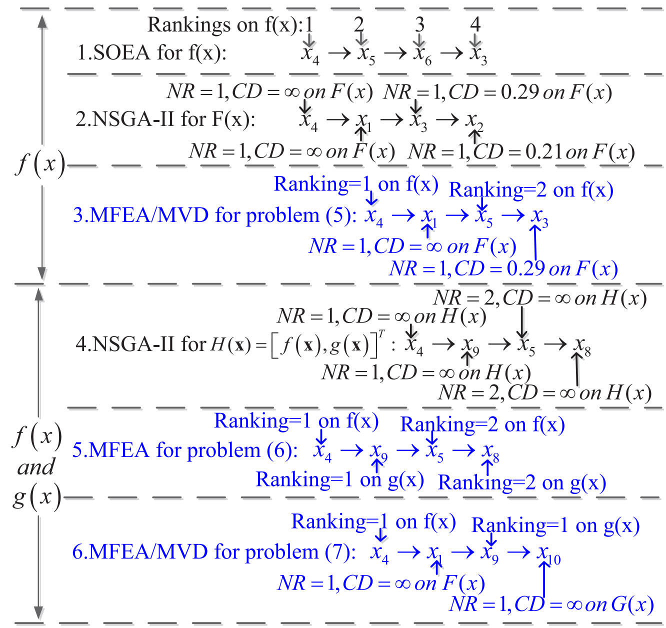


Fig. 6. Environmental selection processes of six solving paradigms on the example shown in Fig. 5, where four elite solutions are selected from ten candidate solutions.

selected by NSGA-II and MFEA tend to converge to the local optimal solution of each objective/task, for example, x_4 for $f(\mathbf{x})$ as shown in Fig. 5(a) and x_9 for $g(\mathbf{x})$ as shown in Fig. Fig. 5(c).

3) In contrast, MFEA/MVD selects solutions $\{x_1, x_4, x_9, x_{10}\}$ when solving (7) as shown in Fig. 5(a)–(d) since x_4, x_9, x_1, and x_{10} are the best solutions of tasks $f(\mathbf{x})$, $g(\mathbf{x})$, $\mathbf{F}(\mathbf{x})$, and $\mathbf{G}(\mathbf{x})$, respectively. Most importantly, the solutions $\{x_1, x_4, x_9, x_{10}\}$ selected by MFEA/MVD are better than that selected by MFEA and NSGA-II in terms of population diversity and the distance to the global optimal solution of each original task.

Next, we consider the scenario, where $f(\mathbf{x})$ is the main task and $g(\mathbf{x})$ is an assistant task.

1) Using the multiobjectivization problem $[f(\mathbf{x}), g(\mathbf{x})]^T$ to solve $f(\mathbf{x})$ reduces the selection pressure toward the main task $f(\mathbf{x})$ [68].
2) If the assistant task $g(\mathbf{x})$ in (6) is randomly selected or weakly related to the main task $f(\mathbf{x})$, optimizing tasks $f(\mathbf{x})$ and $g(\mathbf{x})$ simultaneously would result in noneffective or negative knowledge transfers [6], [7], [69].
3) On solving (7), MVD is employed to construct related helper-tasks and improve the intertask correlation and population diversity. This strategy allows obtaining more rich information based on both original and helper tasks compared to employing only the original task.

IV. Experimental Results

The performance of the proposed MFEA/MVD is validated in comparison with SOEA [6], NSGA-II [4], MFEA [6], and two variations of MFEA, namely, LDA-MFEA [31] and MFEARR [33]. The analysis allows us to answer the following questions:

TABLE III
BENCHMARK TEST SET 1. TWO SUBOBJECTIVES $f_1(\mathbf{x})$ AND $f_2(\mathbf{x})$ AND THE HELPER-MOP $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]^T$ FOR EACH SOP $f(\mathbf{x})$ ARE PROPOSED IN THIS ARTICLE

Problem	Objective function $f(\mathbf{x})$	1st sub-objective $f_1(\mathbf{x})$	2nd sub-objective $f_2(\mathbf{x})$	Helper-MOP $\mathbf{F}(\mathbf{x})$
Sphere	$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2, \mathbf{x} \in [-100, 100]^n$	$\sum_{i=1}^{n/2} x_i^2$	$\sum_{i=n/2+1}^{n} x_i^2$	$[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$
Rosenbrock	$f(\mathbf{x}) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$, $\mathbf{x} \in [-50, 50]^n$	$\sum_{i=1}^{n-1} 100(x_i^2 - x_{i+1})^2$	$\sum_{i=1}^{n-1} (x_i - 1)^2$	$[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$
Griewank	$f(\mathbf{x}) = 1 + \sum_{i=1}^{n} x_i^2/4000 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i})$, $\mathbf{x} \in [-100, 100]^n$	$\sum_{i=1}^{n} x_i^2/4000$	$1 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i})$	$[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$
Rastrgin	$f(\mathbf{x}) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$, $\mathbf{x} \in [-50, 50]^n$	$\sum_{i=1}^{n/2} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$\sum_{i=n/2+1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$
Ackley	$f(\mathbf{x}) = 20 - 20\exp(-0.2\sqrt{\sum_{i=1}^{n} x_i^2/n}) + e - \exp(\sum_{i=1}^{n} \cos(2\pi x_i)/n), \mathbf{x} \in [-50, 50]^n$	$20 - 20\exp(-0.2\sqrt{\sum_{i=1}^{n} x_i^2/n})$	$e - \exp(\sum_{i=1}^{n} \cos(2\pi x_i)/n)$	$[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$
Schwefel	$f(\mathbf{x}) = 418.9829n - \sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i \rvert})$, $\mathbf{x} \in [-500, 500]^n$	$418.9829n/2 - \sum_{i=1}^{n/2} x_i \sin(\sqrt{\lvert x_i \rvert})$	$418.9829n/2 - \sum_{i=n/2+1}^{n} x_i \sin(\sqrt{\lvert x_i \rvert})$	$[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$
Weierstrass	$f(\mathbf{x}) = \sum_{i=1}^{n} \sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (x_i + 0.5))] - n\sum_{k=0}^{k_{max}} [a^k \cos(\pi b^k)]$, $a = 0.5, b = 3, k_{max} = 20, \mathbf{x} \in [-0.5, 0.5]^n$	$\sum_{i=1}^{n/2} \sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (x_i + 0.5))] - n/2 \sum_{k=0}^{k_{max}} [a^k \cos(\pi b^k)]$	$\sum_{i=n/2+1}^{n} \sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (x_i + 0.5))] - n/2 \sum_{k=0}^{k_{max}} [a^k \cos(\pi b^k)]$	$[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$

TABLE IV
BENCHMARK MTO PROBLEM SET PROPOSED FOR THE CEC 2017 EVOLUTIONARY MULTITASK OPTIMIZATION COMPETITION [70]

Category	Task	n	Landscape	Degree of intersection	R_s
CIHS	Griewank(T_1) Rastrigin(T_2)	50 50	multimodal+nonseparable multimodal+separable	Complete intersection	1.0000
CIMS	Ackley(T_1) Rastrigin(T_2)	50 50	multimodal+nonseparable multimodal+separable	Complete intersection	0.2261
CILS	Ackley(T_1) Schwefel(T_2)	50 50	multimodal+nonseparable multimodal+separable	Complete intersection	0.0002
PIHS	Rastrigin(T_1) Sphere(T_2)	50 50	multimodal+separable unimodal+separable	Partial intersection	0.8670
PIMS	Ackley(T_1) Rosenbrock(T_2)	50 50	multimodal+nonseparable multimodal+nonseparable	Partial intersection	0.2154
PILS	Ackley(T_1) Weierstrass(T_2)	50 25	multimodal+nonseparable multimodal+separable	Partial intersection	0.0725
NIHS	Rosenbrock(T_1) Rastrigin(T_2)	50 50	multimodal+nonseparable multimodal+separable	No intersection	0.9434
NIMS	Griewank(T_1) Weierstrass(T_2)	50 50	multimodal+nonseparable multimodal+separable	No intersection	0.3669
NILS	Rastrigin(T_1) Schwefel(T_2)	50 50	multimodal+separable multimodal+separable	No intersection	0.0016

1) Is MVD a good way to construct helper-task(s)?
2) Whether the simultaneous optimization of $f(\mathbf{x})$ and $\mathbf{F}(\mathbf{x})$ in MFEA/MVD can significantly outperform the single-task optimization, that is, SOEA for $f(\mathbf{x})$ and NSGA-II for $\mathbf{F}(\mathbf{x})$?
3) Can MFEA/MVD outperform MFEA and its variants?

A. Benchmark Test Sets and the Proposed Helper-Task Based on MVD

Two benchmark test sets are used in the following experimental study. The first set is a SOO test set containing seven SOPs, that is, Ackley, Griewank, Rastrigin, Rosenbrock, Schwefel, Sphere, and Weierstrass. Table III details the SOPs, along with their two subobjectives and helper-MOPs.

The second set is an MTO test set taken from the CEC 2017 Evolutionary Multitask Optimization Competition [70]. Each MTO problem includes two SOPs. Table IV summarizes the MTO problems that are divided into nine categories based on the intertask landscape similarity and intersection degree of their global optima. Table V defines the concepts of complete

TABLE V
INTERTASK LANDSCAPE SIMILARITY AND THE INTERSECTION DEGREE OF THE GLOBAL OPTIMA FOR OPTIMIZATION TASKS, WHERE R_s IS SPEARMAN'S RANK CORRELATION COEFFICIENT [71]

Property	Description
High similarity (HS)	$R_s \geq 0.8$
medium similarity (MS)	$0.2 \leq R_s < 0.8$
Low similarity (LS)	$R_s < 0.2$
Complete intersection (CI)	The global optima of optimization tasks are same in the unified search space on all variables
Partial intersection (PI)	The global optima of optimization tasks are same in the unified search space on a subset of variables only, and different on the remaining variables
No intersection (NI)	The global optima of optimization tasks are different in the unified search space on all variables

intersection (CI), partial intersection (PI), no intersection (NI), low similarity (LS), medium similarity (MS), and high similarity (HS). Spearman's rank (R_s) correlation coefficient [71] is used to estimate the intertask landscape similarity as shown in Tables IV and V.

Based on the two benchmark test sets, two experiments are conducted as follows.

1) The performance of MFEA/MVD in solving (5) is compared with SOEA for $f(\mathbf{x})$ and NSGA-II for $\mathbf{F}(\mathbf{x})$ based on the single-task SOO test set.
2) MFEA/MVD is compared with NSGA-II, MFEA, LDA-MFEA, and MFEARR using the MTO test set.

B. Parameter Settings

The parameters of the algorithms are set as follows.

1) All the compared algorithms use the same evolutionary operators, that is, simulated binary crossover (SBX) and polynomial mutation (PM) [4], with distribution indexes set to 15. The crossover probability is set to 1, while the mutation probability is set to $1/n$.
2) The random mating probability *rmp* in MFEA, LDA-MFEA, MFEARR, and MFEA/MVD is set to 0.3.
3) The population size $N = 50$ and the maximum number of function evaluations 50 000 are used for SOEA in solving $f(\mathbf{x})$, NSGA-II in solving $\mathbf{F}(\mathbf{x})$, and MFEA/MVD in solving (5). For fair comparison, $N = 100$ and the maximum number of function evaluations of

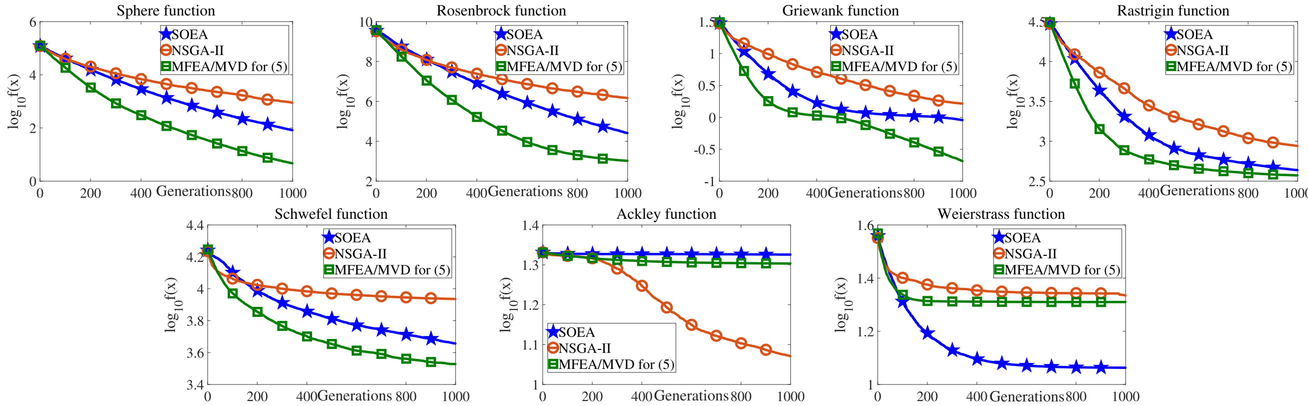


Fig. 7. Plot of the average convergence trend of the compared algorithms on seven SOPs.

TABLE VI
MEAN AND STANDARD DEVIATION OF THE APPROXIMATION SOLUTION OBTAINED BY THE COMPARED ALGORITHMS ON TEST SUITE 1. BASED ON THE t-TEST AT A 5% SIGNIFICANCE LEVEL, "+" IS THE PERFORMANCE OF BEST ALGORITHM ARE STATISTICALLY BETTER THAN OTHER COMPARED ALGORITHM

Function	SOEA	NSGA-II for $\mathbf{F}(\mathbf{x})$	MFEA/MVD for (5)
Sphere	8.26E+01$^+$ (1.47E+01)	8.79E+02$^+$ (3.17E+02)	**4.67E+00** (1.38E+00)
Rosenbrock	2.30E+04$^+$ (9.34E+03)	1.86E+06$^+$ (7.38E+05)	**8.49E+02** (2.78E+02)
Griewank	9.07E-01$^+$ (6.87E-02)	1.64E+00$^+$ (2.59E-01)	**2.06E-01** (4.32E-02)
Rastrigin	4.24E+02$^+$ (5.35E+01)	9.95E+02$^+$ (6.42E+02)	**3.39E+02** (6.12E+01)
Schwefel	4.54E+03$^+$ (5.58E+02)	8.63E+03$^+$ (7.30E+02)	**3.37E+03** (4.35E+02)
Ackley	2.12E+01$^+$ (1.19E-01)	**1.18E+01** (1.35E+00)	2.01E+01$^+$ (6.69E-02)
Weierstrass	**3.75E+01** (4.03E+00)	5.45E+01$^+$ (1.81E+00)	4.79E+01$^+$ (4.80E+00)
Average rank sum	2	2.71	**1.29**

100 000 are used for NSGA-II in solving $[f(\mathbf{x}), g(\mathbf{x})]^T$, MFEA/MVD in solving problem (7), and MFEA, LDA-MFEA, and MFEARR in solving (6).

To enable a statistical significance comparison, all the compared algorithms are independently run for 20 times on each test problem. The statistical testing results are provided based on the t-test at a 5% significance level.

C. Results on Test Suite 1: Single-Task Optimization

Test suite 1 is a single-task SOO problem set including seven SOPs. For this test suit, Table VI reports the mean and standard deviation of the best solutions obtained by MFEA/MVD for (5), SOEA for $f(\mathbf{x})$, and NSGA-II for $\mathbf{F}(\mathbf{x})$. The best average value on each SOP is highlighted in bold. Note that the best solution found by NSGA-II in solving $\mathbf{F}(\mathbf{x})$ is the smallest value of $f_1(\mathbf{x}) + f_2(\mathbf{x})$ among the obtained nondominated solutions. The convergence curves of the three compared algorithms are shown Fig. 7.

1) According to Table VI and Fig. 7, MFEA/MVD achieves faster convergence, better solutions, and better stability than SOEA. The results imply the effectiveness of incorporating the prior knowledge-based helper-task $\mathbf{F}(\mathbf{x})$. The introduction of the MVD problem $\mathbf{F}(\mathbf{x})$ can improve the population diversity and help to solve problem $f(\mathbf{x})$.
2) The performance of NSGA-II in solving the helper-task $\mathbf{F}(\mathbf{x})$ is worse than that of SOEA in solving $f(\mathbf{x})$, which implies that merely using MVD may not be effective. However, MFEA/MVD performs better in solving $f(\mathbf{x})$ and $\mathbf{F}(\mathbf{x})$ simultaneously than SOEA in solving $f(\mathbf{x})$ and NSGA-II in solving $\mathbf{F}(\mathbf{x})$. This result demonstrates the benefits of combining MVD with multitask optimization. The success of MFEA/MVD comes from two aspects. Compared with SOEA applied to $f(\mathbf{x})$ [NSGA-II applied to $\mathbf{F}(\mathbf{x})$], MFEA/MVD solves the task $f(\mathbf{x})$ and its helper-task $\mathbf{F}(\mathbf{x})$ simultaneously via intertask knowledge transfer. The added helper-task $\mathbf{F}(\mathbf{x})$ is a multiobjectivization of the original task $f(\mathbf{x})$. $\mathbf{F}(\mathbf{x})$ is strongly related to $f(\mathbf{x})$ and hence solving $\mathbf{F}(\mathbf{x})$ can help solve the original task $f(\mathbf{x})$, especially by reducing the number of local optima and improving the population diversity.
3) The overall performance of the proposed MFEA/MVD method is better than that of SOEA and NSGA-II in solving the SOP $f(\mathbf{x})$. This result indicates that multitask optimization can achieve better solutions and faster convergence than single-task optimization when the intertask correlation is strong. MVD indeed is a good way of generating strong related helper-task(s).
4) The helper-task $\mathbf{F}(\mathbf{x})$ is unhelpful in solving Ackley and Weierstrass functions. The reason may be that: a) introducing plateaus of nondominated solutions may mislead the MVD method to explore regions away from the global optimum [72], and b) it is problem dependent on the optimal choice for how many and which decomposed subfunctions should be used in the MVD [19]. An added task could contribute negatively, however, since the added task is generated from the original task, the likelihood of positive transfer is much higher.

D. Results on Test Suite 2: MTO

Test suite 2 consists of nine MTO problems, each of which is made up of two tasks $f(\mathbf{x})$ and $g(\mathbf{x})$. To study whether incorporating MVD can enhance MFEA, the proposed MFEA/MVD is applied to solve (7) and compared with

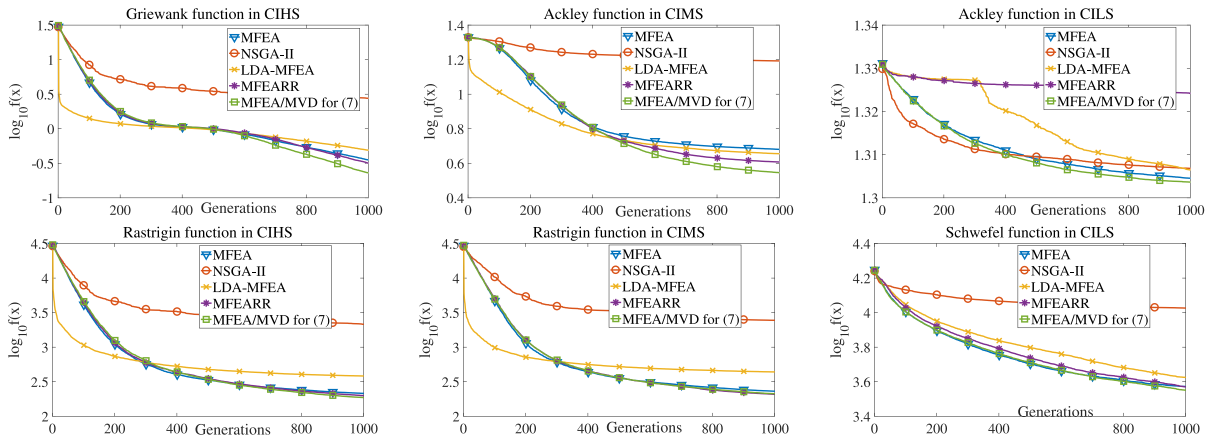


Fig. 8. Plot of the average convergence trend of each optimization task on CIHS, CIMS, and CILS.

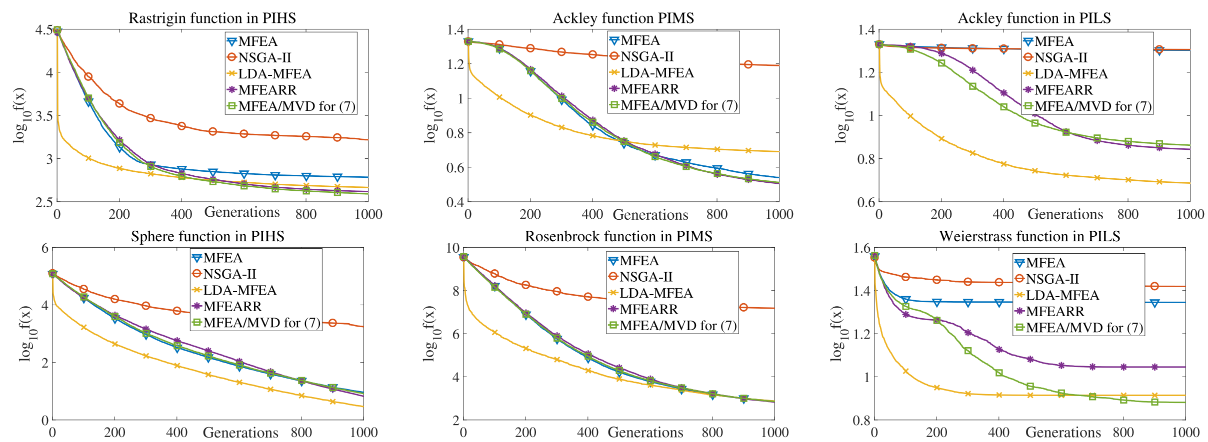


Fig. 9. Plot of the average convergence trend of each optimization task on PIHS, PIMS, and PILS.

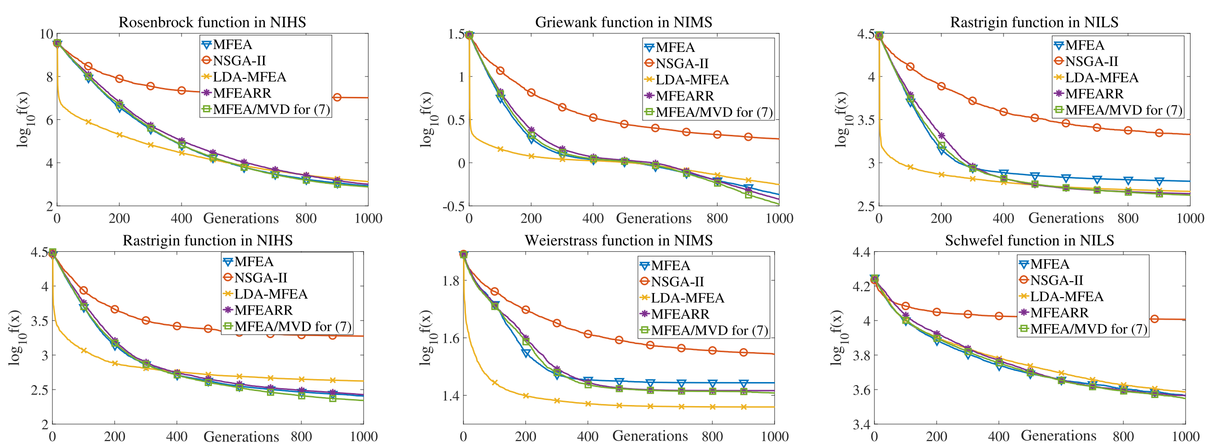


Fig. 10. Plot of the average convergence trend of each optimization task on NIHS, NIMS, and NILS.

MFEAs for (6) and NSGA-II for $[f(\mathbf{x}), g(\mathbf{x})]^T$. Table VII reports the mean and standard deviation of the best solutions obtained by the above three compared algorithms. The best average value on each task is highlighted in bold. Figs. 8–10 show the convergence curves of the three compared algorithms.

1) According to Table VII and Figs. 8–10, the proposed MFEA/MVD performs better than the other MFEAs because it employs strongly related helper-tasks $\mathbf{F}(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$. This result consistently implies that the proposed MVD is a good way of generating related helper-task(s). On the contrary, other MFEAs use the task with the unknown intertask correlation to help solve the target task. They tends to suffer from noneffective or negative knowledge transfer.
2) In the comparison of multitask optimization and MOO, MTO algorithms (i.e., MFEA/MVD, MFEA,

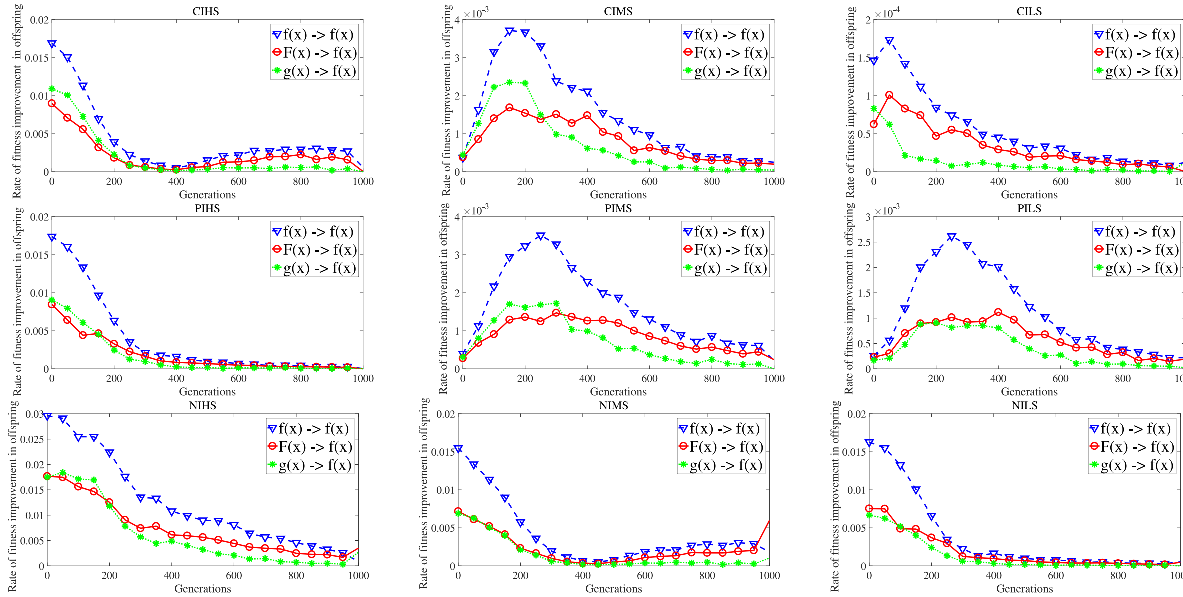


Fig. 11. Flow of intertask genetic transfers and the rate of fitness improvement of its offspring to the current best solution.

TABLE VII
MEAN AND STANDARD DEVIATION OF THE APPROXIMATION SOLUTION OBTAINED BY THE COMPARED ALGORITHMS ON TEST SUITE 2

MTO problem	Task	Performance				
		MFEA	LDA-MFEA	MFEARR	NSGA-II	MFEA/MVD for (7)
CIHS	Griewank	3.54E-1$^+$ (7.00E-2)	4.90E-1$^+$ (1.20E-1)	3.18E-1$^+$ (5.05E-2)	2.76E+0$^+$ (3.87E-1)	**2.30E-1** (7.19E-2)
	Rastrigin	2.14E+2$^+$ (4.16E+1)	3.83E+2$^+$ (9.03E+1)	1.99E+2$^+$ (4.37E+1)	2.14E+3$^+$ (3.72E+2)	**1.87E+2** (4.05E+1)
CIMS	Ackley	4.79E+0$^+$ (6.98E-1)	4.52E+0$^+$ (7.06E-1)	4.05E+0$^+$ (7.52E-1)	1.56E+1$^+$ (1.95E-1)	**3.51E+0** (5.05E-1)
	Rastrigin	2.30E+2$^+$ (4.19E+1)	4.38E+2$^+$ (9.03E+1)	**2.08E+2** (5.71E+1)	2.45E+3$^+$ (1.30E+2)	2.12E+2$^+$ (5.31E+1)
CILS	Ackley	2.02E+1$^+$ (4.48E-2)	2.03E+1$^{\approx}$ (2.91E+0)	2.11E+1$^+$ (2.91E-1)	2.03E+1$^+$ (7.07E-2)	**2.01E+1** (7.73E-2)
	Schwefel	3.73E+3$^{\approx}$ (5.44E+2)	4.21E+3$^+$ (5.73E+2)	3.72E+3$^+$ (4.28E+2)	1.06E+4$^+$ (8.95E+2)	**3.56E+3** (4.03E+2)
PIHS	Rastrigin	6.09E+2$^+$ (1.17E+2)	4.63E+2$^+$ (8.20E+1)	4.15E+2$^+$ (5.13E+1)	1.65E+3$^+$ (3.21E+2)	**3.90E+2** (8.97E+1)
	Sphere	9.11E+0$^+$ (1.48E+0)	**2.88E+0** (1.39E+0)	6.61E+0$^+$ (2.58E+0)	1.75E+3$^+$ (1.24E+3)	8.22E+0$^+$ (2.03E+0)
PIMS	Ackley	3.47E+0$^+$ (6.29E-1)	4.91E+0$^+$ (1.07E+0)	**3.20E+0** (4.15E-01)	1.55E+1$^+$ (1.67E+0)	3.25E+0$^{\approx}$ (4.75E-1)
	Rosenbrock	7.35E+2$^+$ (3.22E+2)	7.46E+2$^+$ (7.11E+2)	**6.74E+2** (2.01E+2)	1.51E+7$^+$ (8.50E+6)	7.14E+2$^+$ (2.92E+2)
PILS	Ackley	2.01E+1$^+$ (6.97E-2)	**4.85E+0** (1.14E+0)	6.97E+0$^+$ (5.81E+0)	2.02E+1$^+$ (5.36E-2)	7.28E+0$^+$ (6.51E+0)
	Weierstrass	2.21E+1$^+$ (2.23E+0)	8.21E+0$^+$ (1.80E+0)	1.11E+1$^+$ (5.97E+0)	2.63E+1$^+$ (2.40E+0)	**7.61E+0** (6.51E+0)
NIHS	Rosenbrock	8.54E+2$^+$ (3.61E+2)	1.33E+3$^+$ (4.93E+2)	1.00E+3$^+$ (6.88E+2)	4.34E+2$^{\approx}$ (6.64E+6)	**3.65E+2** (4.01E+2)
	Rastrigin	2.55E+2$^{\approx}$ (6.69E+1)	4.20E+2$^+$ (6.97E+1)	2.67E+2$^+$ (6.31E+1)	1.88E+3$^+$ (5.14E+2)	**2.20E+2** (5.51E+1)
NIMS	Griewank	4.28E-1$^+$ (7.38E-2)	5.59E-1$^+$ (6.53E-2)	3.78E-1$^+$ (6.53E-2)	1.89E+0$^+$ (5.11E-1)	**3.31E-1** (6.69E-2)
	Weierstrass	2.78E+1$^+$ (3.07E+0)	**2.29E+01** (3.77E+0)	2.61E+1$^+$ (5.17E+0)	3.49E+1$^+$ (4.09E+0)	2.56E+1$^{\approx}$ (3.61E+0)
NILS	Rastrigin	6.10E+2$^+$ (1.36E+2)	4.65E+2$^+$ (6.76E+1)	4.37E+2$^{\approx}$ (9.32E+1)	2.13E+3$^+$ (5.10E+2)	**4.21E+2** (8.10E+1)
	Schwefel	3.68E+3$^+$ (4.48E+2)	3.86E+3$^+$ (4.56E+2)	3.67E+3$^+$ (3.82E+2)	1.02E+4$^+$ (1.00E+3)	**3.54E+3** (4.71E+2)
	Average rank sum	3.22	3.17	2.22	4.94	**1.44**

LDA-MFEA, and MFEARR) perform better than the MOO algorithm NSGA-II, which indicates the effectiveness of MTO.

E. Flow of Intertask Knowledge Transfer and Effects of Single Helper-Task Against Multiple Helper-Tasks

This section takes MFEA solving three tasks $f(\mathbf{x})$, $\mathbf{F}(\mathbf{x})$, and $g(\mathbf{x})$ simultaneously for example to visualize the flow of intertask genetic transfers that leads to better performance. Fig. 11 demonstrates the flow of intertask genetic transfer and the fitness improvement rate of the offspring to the current best solution. If one parent solution of task i and another parent solution of task j perform crossover and generate an offspring whose skill factor is j and the fitness is better than the current best solution, the flow of genetic transfer is denoted as "task $i \rightarrow$ task j" in Fig. 11.

1) According to Fig. 11, the flow of "$\mathbf{F}(\mathbf{x}) \rightarrow f(\mathbf{x})$" has higher fitness improvements than the flow of "$g(\mathbf{x}) \rightarrow f(\mathbf{x})$" in most cases. This implies that the helper-task $\mathbf{F}(\mathbf{x})$ can better help solve the task $f(\mathbf{x})$ than the task $g(\mathbf{x})$.
2) The flow of "$f(\mathbf{x}) \rightarrow f(\mathbf{x})$" outperforms the flow of "$\mathbf{F}(\mathbf{x}) \rightarrow f(\mathbf{x})$" in most cases. This means that in performing the crossover operator, two parent solutions from the same task do better than one from the task $f(\mathbf{x})$ and another from the task $\mathbf{F}(\mathbf{x})$. The observation is reasonable as $f(\mathbf{x})$ can learn more information from itself than that from $\mathbf{F}(\mathbf{x})$.

To study the effects of using single helper-task against multiple helper-tasks, we compare three solving models: 1) MFEA for two tasks [i.e., $f(\mathbf{x})$ and $\mathbf{F}(\mathbf{x})$]; 2) MFEA for three tasks [i.e., $f(\mathbf{x})$, $\mathbf{F}(\mathbf{x})$, and $g(\mathbf{x})$]; and 3) MFEA/MVD for four tasks [i.e., $f(\mathbf{x})$, $\mathbf{F}(\mathbf{x})$, $g(\mathbf{x})$, and $\mathbf{G}(\mathbf{x})$]. Table VIII presents the mean and standard deviation of the best solutions obtained by the compared algorithms.

1) When we drop out $\mathbf{G}(\mathbf{x})$, the MFEA for optimizing $f(\mathbf{x})$, $\mathbf{F}(\mathbf{x})$, and $g(\mathbf{x})$ can perform better than MFEA for optimizing two tasks $f(\mathbf{x})$ and $g(\mathbf{x})$, but worse than MFEA/MVD for optimizing four tasks $f(\mathbf{x})$, $\mathbf{F}(\mathbf{x})$, $g(\mathbf{x})$, and $\mathbf{G}(\mathbf{x})$. The results imply that introducing a

TABLE VIII
MEAN AND STANDARD DEVIATION OF THE APPROXIMATION SOLUTION OBTAINED BY THE COMPARED ALGORITHMS ON TEST SUITE 2, WHERE THE BEST AVERAGE VALUE IS HIGHLIGHTED IN BOLD

MTO problem	Task	Performance: NSGA-II for $[f(\mathbf{x}), g(\mathbf{x})]$	MFEA for $f(\mathbf{x})$ and $g(\mathbf{x})$	MFEA for $f(\mathbf{x})$, $\mathbf{F}(\mathbf{x})$, and $g(\mathbf{x})$	MFEA/MVD for (7)
CIHS	Griewank	2.76E+0$^+$ (3.87E-1)	3.54E-1$^+$ (7.00E-2)	2.40E-1$^+$ (6.15E-2)	**2.30E-1** (7.19E-2)
	Rastrigin	2.14E+3$^+$ (3.72E+2)	2.14E+2$^+$ (4.16E+1)	2.08E+2$^+$ (5.02E+1)	**1.87E+2** (4.05E+1)
CIMS	Ackley	1.56E+1$^+$ (1.95E-1)	4.79E+0$^+$ (6.98E-1)	3.45E+0$^+$ (5.23E-1)	**3.51E+0** (5.05E-1)
	Rastrigin	2.45E+3$^+$ (1.30E+2)	2.30E+2$^+$ (4.19E+1)	2.21E+2$^{\approx}$ (4.01E+1)	2.12E+2$^+$ (5.31E+1)
CILS	Ackley	2.03E+1$^+$ (7.07E-2)	2.02E+1$^+$ (4.48E-2)	2.01E+1$^{\approx}$ (6.70E-2)	**2.01E+1** (7.73E-2)
	Schwefel	1.06E+4$^+$ (8.95E+2)	3.73E+3$^{\approx}$ (5.44E+2)	3.59E+3$^+$ (5.18E+2)	**3.56E+3** (4.03E+2)
PIHS	Rastrigin	1.65E+3$^+$ (3.21E+2)	6.09E+2$^+$ (1.17E+2)	4.17E+2$^+$ (8.01E+1)	**3.90E+2** (8.97E+1)
	Sphere	1.75E+3$^+$ (1.24E+3)	9.11E+0$^+$ (1.48E+0)	8.29E+0$^+$ (1.97E+0)	**8.22E+0** (2.03E+0)
PIMS	Ackley	1.55E+1$^+$ (1.67E+0)	3.47E+0$^+$ (6.29E-1)	3.26E+0$^{\approx}$ (3.51E-1)	**3.25E+0** (4.75E-1)
	Rosenbrock	1.51E+7$^+$ (8.50E+6)	7.35E+2$^+$ (3.22E+2)	**6.35E+2** (2.53E+2)	7.14E+2$^+$ (2.92E+2)
PILS	Ackley	2.02E+1$^+$ (5.36E-2)	2.01E+1$^+$ (6.97E-2)	8.70E+0$^+$ (4.93E+0)	**7.28E+0** (6.51E+0)
	Weierstrass	2.63E+1$^+$ (2.40E+0)	2.21E+1$^+$ (2.23E+0)	9.28E+0$^+$ (5.56E+0)	**7.61E+0** (6.51E+0)
NIHS	Rosenbrock	4.34E+2$^{\approx}$ (6.64E+6)	8.54E+2$^+$ (3.61E+2)	7.32E+2$^+$ (3.71E+2)	**3.65E+2** (4.01E+2)
	Rastrigin	1.88E+3$^+$ (5.14E+2)	2.55E+2$^{\approx}$ (6.69E+1)	2.61E+2$^+$ (6.33E+1)	**2.20E+2** (5.51E+1)
NIMS	Griewank	1.89E+0$^+$ (5.11E-1)	4.28E-1$^+$ (7.38E-2)	**3.04E-1** (6.66E-2)	3.31E-01$^+$ (6.69E-2)
	Weierstrass	3.49E+1$^+$ (4.09E+0)	2.78E+1$^+$ (3.07E+0)	2.73E+1$^+$ (2.60E+0)	**2.56E+1** (3.61E+0)
NILS	Rastrigin	2.13E+3$^+$ (5.10E+2)	6.10E+2$^+$ (1.36E+2)	4.33E+2$^+$ (8.89E+1)	**4.21E+2** (8.10E+1)
	Schwefel	1.02E+4$^+$ (1.00E+3)	3.68E+3$^+$ (4.48E+2)	3.70E+3$^+$ (3.56E+2)	**3.54E+3** (4.71E+2)

helper-task $\mathbf{F}(\mathbf{x})$ can also improve the performance of MFEA.

2) Introducing helper-tasks $\mathbf{F}(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$ simultaneously into MFEA performs the best, as more positive knowledge transfer is achieved.

V. CONCLUSION

In MFEA, noneffective or even negative knowledge transfer may occur on uncorrelated or weakly correlated tasks, preventing the algorithm from solving the tasks. To address this issue, this article incorporates the prior knowledge-based MVD into MFEA to construct helper-tasks and improve the performance of MFEA. According to the proposed method, MVD creates a prior knowledge-based helper-task for each original task based on the problem structure or decision variable grouping. The resulting helper-task is strongly correlated with original task and, hence, can facilitate the positive intertask knowledge transfer. The resultant algorithm MFEA/MVD optimizes the original tasks and their helper-tasks simultaneously. It exploits the advantages of MVD by reducing the number of local optima and improving the population diversity.

MFEA/MVD is tested on the most widely used benchmark problems and the following results are observed.

1) An assistant optimization task highly similar to the target task can help the optimization of the target task.
2) MVD is a simple yet effective way of constructing strong related helper-task. Combining MVD with MTO can obtain better performance than solving the MVD problem or MTO problem individually.
3) When multiple optimization tasks are considered simultaneously, MTO may perform better than MOO, especially when the intertask correlation is high.
4) The proposed MVD strategies are effective for separable functions but not for nonseparable functions.

In the future work, it would be interesting to design more general MVD strategies for black-box and complex problems, along with more ways of constructing other helper-tasks. Introducing multiobjectivization via helper-objectives to construct related tasks is another potential direction. The idea of meme helper tasks has also been briefly introduced in [73] and [74], which presents a data-driven view of optimization through the framework of memetic computation. The work can also be extended to multitask multiobjective or many-objective optimization [75], [76]. The MATLAB source code of MFEA/MVD is available from the authors upon request.

REFERENCES

[1] T. Back, U. Hammel, and H.-P. Schwefel, “Evolutionary computation: Comments on the history and current state,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 3–17, Apr. 1997.

[2] C. M. Fonseca and P. J. Fleming, “An overview of evolutionary algorithms in multiobjective optimization,” *Evol. Comput.*, vol. 3, no. 1, pp. 1–16, Mar. 1995.

[3] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.

[4] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[5] 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, Apr. 2016.

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

[7] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

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

[9] L. Feng *et al.*, “Evolutionary multitasking via explicit autoencoding,” *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3457–3470, Sep. 2019.

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

[11] A. T. W. Min, A. Gupta, Y.-S. Ong, and C.-K. Goh, “Multiproblem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 15–28, Feb. 2019.

[12] R. Sagarna and Y.-S. Ong, “Concurrently searching branches in software tests generation through multitask evolution,” in *Proc. IEEE Symp. Comput. Intell.*, Athens, Greece, 2017, pp. 1–8.

[13] B. Liang *et al.*, “An evolutionary multitasking algorithm for cloud computing service composition,” in *Proc. World Congr. Serv.*, 2018, pp. 130–144.

[14] L. Zhou, L. Feng, J. Zhong, Y.-S. Ong, Z. Zhu, and E. Sha, “Evolutionary multitasking in combinatorial search spaces: A case study in capacitated vehicle routing problem,” in *Proc. IEEE Symp. Comput. Intell. (SSCI)*, Athens, Greece, 2016, pp. 1–8.

[15] Y. Yu, A. Zhu, Z. Zhu, Q. Lin, J. Yin, and X. Ma, “Multifactorial differential evolution with opposition-based learning for multi-tasking optimization,” in *Proc. IEEE Congr. Evol. Comput.*, Wellington, New Zealand, 2019, pp. 1898–1905.

[16] J. Yin, A. Zhu, Z. Zhu, Y. Yu, and X. Ma, "Multifactorial evolutionary algorithm enhanced with cross-task search direction," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Wellington, New Zealand, 2019, pp. 2244–2251.

[17] Y. Zheng, Z. Zhu, Y. Qi, L. Wang, and X. Ma, "Multi-objective multifactorial evolutionary algorithm enhanced with the weighting helper-task," in *Proc. 2nd Int. Conf. Ind. Artif. Intell.*, Shenyang, China, 2020, pp. 1–7.

[18] J. D. Knowles, R. A. Watson, and D. W. Corne, "Reducing local optima in single-objective problems by multi-objectivization," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, 2001, pp. 269–283.

[19] M. T. Jensen, "Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation," *J. Math. Model. Algorithms*, vol. 3, no. 4, pp. 323–347, 2004.

[20] D. F. Lochtefeld and F. W. Ciarallo, "An analysis of decomposition approaches in multi-objectivization via segmentation," *Appl. Soft Comput.*, vol. 18, pp. 209–222, May 2014.

[21] Y.-S. Ong, M. H. Lim, and X. Chen, "Memetic computation—Past, present & future [research frontier]," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 24–31, May 2010.

[22] D. F. Lochtefeld and F. W. Ciarallo, "Multi-objectivization via decomposition: An analysis of helper-objectives and complete decomposition," *Eur. J. Oper. Res.*, vol. 243, no. 2, pp. 395–404, 2015.

[23] M. Jähne, X. Li, and J. Branke, "Evolutionary algorithms and multi-objectivization for the travelling salesman problem," in *Proc. Conf. Genet. Evol. Comput.*, 2009, pp. 595–602.

[24] A. Syberfeldt and J. Rogström, "A two-step multi-objectivization method for improved evolutionary optimization of industrial problems," *Appl. Soft Comput.*, vol. 64, pp. 331–340, Mar. 2018.

[25] D. F. Lochtefeld and F. W. Ciarallo, "Helper-objective optimization strategies for the job-shop scheduling problem," *Appl. Soft Comput.*, vol. 11, no. 6, pp. 4161–4174, 2011.

[26] C. A. C. Coello, G. Lamont, and A. V. D. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Heidelberg, Germany: Springer, 2007.

[27] X. Ma, Q. Zhang, G. Tian, J. Yang, and Z. Zhu, "On Tchebycheff decomposition approaches for multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 226–244, Apr. 2018.

[28] X. Ma *et al.*, "MOEA/D with baldwinian learning inspired by the regularity property of continuous multiobjective problem," *Neurocomputing*, vol. 145, pp. 336–352, Dec. 2014.

[29] X. Ma *et al.*, "MOEA/D with opposition-based learning for multiobjective optimization problem," *Neurocomputing*, vol. 146, pp. 48–64, Dec. 2014.

[30] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.

[31] K. K. Bali, A. Gupta, L. Feng, Y. S. Ong, and T. P. Siew, "Linearized domain adaptation in evolutionary multitasking," in *Proc. IEEE Congr. Evol. Comput.*, San Sebastian, Spain, 2017, pp. 1295–1302.

[32] K. K. Bali, Y.-S. Ong, A. Gupta, and P. S. Tan, "Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 69–83, Feb. 2020.

[33] Y.-W. Wen and C.-K. Ting, "Parting ways and reallocating resources in evolutionary multitasking," in *Proc. IEEE Congr. Evol. Comput.*, San Sebastian, Spain, 2017, pp. 2404–2411.

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

[35] Y. Yuan, Y.-S. Ong, A. Gupta, P. S. Tan, and H. Xu, "Evolutionary multitasking in permutation-based combinatorial optimization problems: Realization with TSP, QAP, LOP, and JSP," in *Proc. Region 10 Conf.*, Singapore, 2017, pp. 3157–3164.

[36] L. Feng *et al.*, "An empirical study of multifactorial PSO and multifactorial DE," in *Proc. IEEE Congr. Evol. Comput.*, San Sebastian, Spain, 2017, pp. 921–928.

[37] Q. Chen, X. Ma, Y. Sun, and Z. Zhu, "Adaptive memetic algorithm based evolutionary multi-tasking single-objective optimization," in *Proc. Int. Conf. Simul. Evol. Learn.*, 2017, pp. 462–472.

[38] Q. Chen, X. Ma, Z. Zhu, and Y. Sun, "Evolutionary multi-tasking single-objective optimization based on cooperative co-evolutionary memetic algorithm," in *Proc. 13th Int. Conf. Comput. Intell. Security*, Hong Kong, China, 2017, pp. 197–201.

[39] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, WA, Australia, 1995, pp. 1942–1948.

[40] Z. Wang and X. Wang, "Multiobjective multifactorial operation optimization for continuous annealing production process," *Ind. Eng. Chem. Res.*, vol. 58, no. 41, pp. 19166–19178, 2019.

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

[42] M. T. Jensen, "Guiding single-objective optimization using multi-objective methods," in *Proc. Workshops Appl. Evol. Conf.*, 2003, pp. 268–279.

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

[44] H. A. Abbass and K. Deb, "Searching under multi-evolutionary pressures," in *Proc. 2nd Int. Conf. Evol. Multi-Criterion Optim.*, 2003, pp. 391–404.

[45] S. Watanabe and K. Sakakibara, "Multi-objective approaches in a single-objective optimization environment," in *Proc. IEEE Congr. Evol. Comput.*, Edinburgh, U.K., 2005, pp. 1–8.

[46] J. Yao, N. Kharma, and P. Grogono, "Bi-objective multipopulation genetic algorithm for multimodal function optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 80–102, Feb. 2010.

[47] M. Wright, "Subcost-guided search—Experiments with timetabling problems," *J. Heuristics*, vol. 7, no. 3, pp. 251–260, 2001.

[48] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler, "On the effects of adding objectives to plateau functions," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 591–603, Jun. 2009.

[49] A. Alsheddy, "A penalty-based multi-objectivization approach for single objective optimization," *Inf. Sci.*, vols. 442–443, pp. 1–17, May 2018.

[50] T. Brys, A. Harutyunyan, P. Vrancx, A. Nowé, and M. E. Taylor, "Multi-objectivization and ensembles of shapings in reinforcement learning," *Neurocomputing*, vol. 26, pp. 48–59, Nov. 2017.

[51] H. Ishibuchi, H. Masuda, and Y. Nojima, "Meta-level multi-objective formulations of set optimization for multi-objective optimization problems: Multi-reference point approach to hypervolume maximization," in *Proc. Genet. Evol. Comput. Conf.*, 2014, pp. 89–90.

[52] E. Segredo, C. Segura, and C. León, "Memetic algorithms and hyperheuristics applied to a multiobjectivised two-dimensional packing problem," *J. Global Optim.*, vol. 58, no. 4, pp. 769–794, 2014.

[53] C. Segura, E. Segredo, and C. León, "Analysing the robustness of multi-objectivisation approaches applied to large scale optimisation problems," in *EVOLVE—A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation*. Heidelberg, Germany: Springer, 2013, pp. 365–391.

[54] M. Gong, J. Liu, H. Li, Q. Cai, and L. Su, "A multiobjective sparse feature learning model for deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3263–3277, Dec. 2015.

[55] A. J. Nebro, E. Alba, G. Molina, F. Chicano, and J. J. Durillo, "Optimal antenna placement using a new multi-objective chc algorithm," in *Proc. 9th Annu. Genet. Evol. Comput. Conf. (GECCO)*, 2007, pp. 876–883.

[56] J. Handl, S. C. Lovell, and J. Knowles, "Multiobjectivization by decomposition of scalar cost functions," in *Proc. 10th Int. Conf. Parallel Problem Solving Nat.*, 2008, pp. 349–366.

[57] M. Garza-Fabre, G. Toscano-Pulido, and E. Rodriguez-Tello, "Multi-objectivization, fitness landscape transformation and search performance: A case of study on the hp model for protein structure prediction," *Eur. J. Oper. Res.*, vol. 243, no. 2, pp. 405–422, 2015.

[58] B. Da, A. Gupta, Y.-S. Ong, and L. Feng, "Evolutionary multitasking across single and multi-objective formulations for improved problem solving," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2016, no. 1695–1701.

[59] S. J. Louis and G. J. E. Rawlins, "Pareto optimality, GA-easiness and deception," in *Proc. 5th Int. Conf. Genet. Algorithms*, 1993, pp. 1–6.

[60] H. Li, J. Sun, M. Wang, and Q. Zhang, "MOEA/D with chain-based random local search for sparse optimization," *Soft Comput.*, vol. 22, pp. 7087–7102, Sep. 2018.

[61] Y. Zhou, C. Xiao, and G. Zhou, "Multi-objectivization-based localization of underwater sensors using magnetometers," *IEEE Sensors J.*, vol. 14, no. 4, pp. 1099–1106, Apr. 2014.

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

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

[64] F. Caraffini, F. Neri, G. Iacca, and A. Mol, "Parallel memetic structures," *Inf. Sci.*, vol. 227, pp. 60–82, Apr. 2013.

[65] L. Feng, Y.-S. Ong, A.-H. Tan, and X.-S. Chen, "Towards human-like social multi-agents with memetic automaton," in *Proc. IEEE Congr. Evol. Comput.*, New Orleans, LA, USA, 2011, pp. 1092–1099.

[66] Q. H. Nguyen, Y.-S. Ong, and M. H. Lim, "A probabilistic memetic framework," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 604–623, Jun. 2009.

[67] J. Ceberio, B. Calvo, A. Mendiburu, and J. A. Lozano, "Multi-objectivising combinatorial optimisation problems by means of elementary landscape decompositions," *Evol. Comput.*, vol. 27, no. 2, pp. 291–311, 2019.

[68] C. Segura, C. A. C. Coello, G. Miranda, and C. León, "Using multi-objective evolutionary algorithms for single-objective constrained and unconstrained optimization," *Ann. Oper. Res.*, vol. 240, no. 1, pp. 217–250, 2016.

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

[70] B. Da *et al.*, "Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metric, and baseline results," School Comput. Sci. and Eng., Nanyang Technol. Univ., Singapore, Rep. 201701, 2017.

[71] J. H. Zar, "Significance testing of the spearman rank correlation coefficient," *Publ. Amer. Stat. Assoc.*, vol. 67, no. 339, pp. 578–580, 1972.

[72] J. Handl, S. C. Lovell, and J. Knowles, "Investigations into the effect of multiobjectivization in protein structure prediction," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 2008, pp. 1–11.

[73] A. Gupta and Y.-S. Ong, *Memetic Computation: The Mainspring of Knowledge Transfer in a Data-Driven Optimization Era*, Heidelberg, Germany: Springer, 2018.

[74] X. Ma, Q. Chen, Y. Yu, Y. Sun, L. Ma, and Z. Zhu, "A two-level transfer learning algorithm for evolutionary multitasking," *Front. Neurosci.*, vol. 13, p. 1408, Jan. 2020.

[75] X. Ma, Y. Yu, X. Li, Y. Qi, and Z. Zhu, "A survey of weight vector adjustment methods for decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 634–649, Aug. 2020.

[76] X. Ma *et al.*, "A survey on cooperative co-evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 421–441, Jun. 2019.

Xiaoliang Ma (Member, IEEE) received the B.S. degree in computing computer science and technology from Zhejiang Normal University, Jinhua, China, in 2006, and the Ph.D. degree from the School of Computing, Xidian University, Xi'an, China, in 2014.

He is currently an Assistant Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include evolutionary computation, multiobjective optimization, and cooperative coevolution.

Jian Yin received the B.S. degree in civil construction engineering from Tianjin Chengjian University, Tianjin, China, in 2017. He is currently pursuing the M.S. degree with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

His current research interests include evolutionary computation, multitask optimization, and various heuristic algorithms.

Anmin Zhu received the B.Sc. and M.Sc. degrees in computer science from Tongji University, Shanghai, China, in 1987 and 1990, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Guelph, Guelph, ON, Canada, in 2005.

He is currently a Faculty Member with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His current research interests include fuzzy systems, neural networks, robotics, self-organizing arrangement, machine intelligence, and multiagent systems.

Xiaodong Li (Fellow, IEEE) received the B.Sc. degree from Xidian University, Xi'an, China, in 1988 and Ph.D. degree in information science from the University of Otago, Dunedin, New Zealand, in 1998.

He is a Professor with the School of Science (Computer Science and Software Engineering), RMIT University, Melbourne, VIC, Australia. His research interests include machine learning, evolutionary computation, neural networks, data analytics, multiobjective optimization, multimodal optimization, and swarm intelligence.

Prof. Li is a recipient of the 2013 ACM SIGEVO Impact Award and the 2017 IEEE CIS "IEEE Transactions on Evolutionary Computation Outstanding Paper Award." He serves as an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, *Swarm Intelligence* (Springer), and *International Journal of Swarm Intelligence Research*. He is the Founding Member of IEEE CIS Task Force on Swarm Intelligence, the Vice-Chair of IEEE Task Force on Multimodal Optimization, and the Former Chair of the IEEE CIS Task Force on Large Scale Global Optimization.

Yanan Yu received the B.S. degree in computing computer science and technology from the Chengdu University of Information Technology, Chengdu, China, in 2018. She is currently pursuing the M.S. degree with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

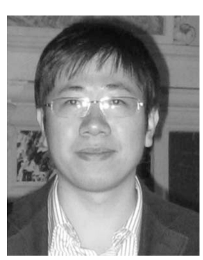
Her current research interests include evolutionary computation, multiobjective optimization, and multitask optimization.

Lei Wang received the Ph.D. degree from Xidian University, Xi'an, China, in 2010.

From 2011 to 2012, he worked with Huawei Technologies Company Ltd., Shenzhen, China. From 2014 to 2015, he was with the Department of Embedded Systems Engineering, Incheon National University, Incheon, South Korea, as a Postdoctoral Fellow. He is currently an Associate Professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen. He has authored or coauthored around 30 papers in conferences and journals, such as IEEE SIGNAL PROCESSING LETTERS and IEEE GEOSCIENCE REMOTE SENSING LETTERS. His research interests include image transforms, machine learning, computer vision, visual semantic understanding, video analysis, and deep learning.

Yutao Qi received the B.Sc. degree in software engineering, the M.Sc. degree in computer science and technology, and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2003, 2006, and 2008, respectively.

He is a Professor with the School of Computer Science and Technology, Xidian University. His current research interests include evolutionary computation, machine learning, and computer vision. His main research interests include evolutionary computation, artificial immune systems, and engineering optimization methods.



Zexuan Zhu (Senior Member, IEEE) received the B.S. degree in computer science and technology from Fudan University, Shanghai, China, in 2003, and the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore, in 2008.

He is currently a Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include computational intelligence, machine learning, and bioinformatics.

Prof. Zhu is an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE. He is also the Chair of the IEEE CIS Emergent Technologies Task Force on Memetic Computing.