Multiobjective Evolutionary Multitasking With Two-Stage Adaptive Knowledge Transfer Based on Population Distribution

Zhengping Liang, Weiqi Liang, Zhiqiang Wang, Xiaoliang Ma, Ling Liu, and Zexuan Zhu, *Senior Member, IEEE*

***Abstract*—Multitasking optimization can achieve better performance than traditional single-tasking optimization by leveraging knowledge transfer between tasks. However, the current multitasking optimization algorithms suffer from some deficiencies. Particularly, on high similar problems, the existing algorithms might fail to take full advantage of knowledge transfer to accelerate the convergence of the search, or easily get trapped in the local optima. Whereas, on low similar problems, they tend to suffer from negative transfer, resulting in performance degradation. To solve these issues, this article proposes an evolutionary multitasking optimization algorithm for multiobjective/many-objective optimization with two-stage adaptive knowledge transfer based on population distribution. The resultant algorithm named EMT-PD can improve the convergence performance of the target optimization tasks based on the knowledge extracted from the probability model that reflects the search trend of the whole population. At the first stage of knowledge transfer, an adaptive weight is used to adjust the search step size of each individual, which can reduce the impact of negative transfer. At the second stage of knowledge transfer, the search range of each individual is further adjusted dynamically, which can improve the population diversity and be beneficial for jumping out of the local optima. Experimental results on multitasking multiobjective optimization test suites show that EMT-PD is superior to other state-of-the-art evolutionary multitasking/single-tasking algorithms. To further investigate the effectiveness of EMT-PD on many-objective optimization problems, a multitasking many-objective optimization test suite is also designed in this article. The experimental results on the new test suite also demonstrate the competitiveness of EMT-PD.**

***Index Terms*—Evolutionary multitasking (EMT), knowledge transfer, many-objective optimization, multiobjective optimization, population distribution.**

Manuscript received April 14, 2021; accepted June 30, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61871272 and Grant 62001300; in part by the National Natural Science Foundation of Guangdong, China, under Grant 2020A1515010479, Grant 2021A1515011911, and Grant 2021A1515011679; in part by the Guangdong Provincial Key Laboratory under Grant 2020B121201001; in part by the Shenzhen Fundamental Research Program under Grant 20200811181752003 and Grant JCYJ20190808173617147; and in part by the BGI-Research Shenzhen Open Funds under Grant BGIRSZ20200002. This article was recommended by Associate Editor N. T. Nguyen. *(Corresponding authors: Ling Liu; Zexuan Zhu.)*

Zhengping Liang, Weiqi Liang, Zhiqiang Wang, Xiaoliang Ma, and Ling Liu are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: liangzp@szu.edu.cn; liangwq0131@foxmail.com; wangzq@szu.edu.cn; maxiaoliang@yeah.net; liulingcs@szu.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 Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: zhuzx@szu.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TSMC.2021.3096220.

Digital Object Identifier 10.1109/TSMC.2021.3096220

I. Introduction

MULTIOBJECTIVE optimization problems (MOPs) widely exist in the real world [1]–[3]. Generally, a MOP can be described as follows:

$$\min_{\boldsymbol{x}} \quad (F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})))$$
$$\text{subject to:} \quad \boldsymbol{x} \in \Omega^D \tag{1}$$

where $\boldsymbol{x} = (x_1, \ldots, x_D)$ represents a D-dimensional decision vector in search space Ω^D. $F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x}))$ denotes the objective function vector of n objectives. A MOP is also referred to as a many-objective optimization problem (MaOP) if $n > 3$ [4]–[6]. A solution $\boldsymbol{x}$ is said to dominate another solution $\boldsymbol{y}$, if and only if $x_i \leq y_i$ for all $i \in [1, n]$ and there exists one objective $j \in [1, n]$ such that $x_j < y_j$. A solution not dominated by any other solution is called a Pareto-optimal solution. All Pareto-optimal solutions form the Pareto-optimal set (PS) of which the corresponding projection in the objective space is called the Pareto front (PF).

MOPs and MaOPs can be effectively solved by multiobjective/many-objective evolutionary algorithms [7]–[20]. The majority of the existing multi/many-objective evolutionary algorithms are designed to handle one problem at a time. To solve a new problem, the algorithms must start from scratch. However, many optimization problems encountered in the real world are correlated to each other. The experience of solving one problem can benefit the solving of another related one. Inspired by the capability of human brain, which processes transactions in parallel, Gupta *et al.* [21] proposed a new optimization paradigm called evolutionary multitasking (EMT) to deal with multiple optimization tasks simultaneously. Compared to the single-tasking counterpart algorithms, EMT has been shown to be able to considerably improve the overall performance in solving multiple related optimization tasks via knowledge transfer [22].

Following [21], a number of multiobjective EMT algorithms (MOMFEAs) have been proposed in the literature to solve various MOPs. For example, Gupta *et al.* extended the work of [21] and proposed an MOMFEA [23] to deal with multiple MOPs simultaneously through assortative mating and vertical cultural transmission. Yang *et al.* [24] presented a two-stage assortative mating method for multiobjective EMT with decision variables classification, i.e., assortative mating is carried out on different variable groups with different parameters to balance the diversity and convergence. Feng *et al.* [25] proposed an EMT algorithm with explicit genetic transfer (EMT-EGT) to enhance the search ability of the evolution population. Chen *et al.* [26] introduced a memetic EMT framework for knowledge transfer between subpopulations. Tuan *et al.* [27] put forward an EMT algorithm employing the local search strategy to accelerate the convergence of population in multiobjective continuous optimization. EMT has also achieved successes in real-world applications, e.g., permutation-based combinatorial optimization problems [28], branch testing problems in software engineering [29], modular knowledge representation in neural networks [30], symbolic regression problems [31], multiobjective pollution-routing problems [32], and hyperspectral image unmixing [33].

The research on EMT algorithms has made remarkable progress, yet there remains room for further improvement and some open question still needed to be addressed. Particularly, on high similar problems, the existing algorithms might fail to fully use the knowledge of high-quality solutions to improve the convergence of the population, or do not take good care of the situation, where the population falls into the local optima. On low similar problems, the population distributions of the tasks are generally different, which tends to result in a negative transfer between tasks [34].

To address the aforementioned issues, this article proposes a new MOMFEA abbreviated as EMT-PD with two-stage adaptive knowledge transfer based on population distribution. EMT-PD first builds probability models for each task and then obtains knowledge from the product of different probability models. The knowledge can help to accelerate the convergence rate of population. At the first stage of knowledge transfer, the search step size of each individual is adjusted by adaptive weight, which can reduce the probability of generating negative transfer. At the second stage of knowledge transfer, the search range of each individual is dynamically adjusted to promote the population diversity and avoid getting trapped in the local optima. EMT-PD is then tested on multitasking multi/many-objective optimization test suites. The comparison studies with other state-of-the-art evolutionary multi/single-tasking algorithms demonstrate the competitiveness of EMT-PD. The main contributions of this article are highlighted as follows.

1) A multitasking multiobjective/many-objective evolutionary optimization algorithm with a novel method of extracting and transferring knowledge is proposed to improve the efficiency and performance of optimization.
2) A multitasking many-objective optimization test suite is designed based on a representative many-objective test suite MaF [35].
3) Based on three test suites, the strengths and weaknesses of EMT-PD are fully analyzed by comparing with other state-of-the-art algorithms.

The remainder of this article is organized as follows. Section II introduces the related work and the motivation of EMT-PD. Section III describes the details of EMT-PD. Section IV presents the experimental design and results. Finally, Section V concludes this work, with additional discussions on some potential future directions.

II. Related Work and Motivation

This section briefly reviews the related knowledge extracting and transferring methods used in EMT and explains the motivation of the proposed method.

A. Extracting and Transferring Knowledge in EMT

In EMT algorithms, knowledge can be extracted from a single or multiple individuals and transferred to other individuals to facilitate their search. Particularly, knowledge transfer methods based on single individual (KTS) refer to extracting knowledge from a single individual of one task, and transferring knowledge to other tasks. EMT algorithms with KTS include MFEA [21], M-BLEA [36], LDA-MFEA [37], S&M-MFEA [38], MO-MFEA [23], GMFEA [39], TMO-MFEA [24], MTO-DRA [40], MFEA-II [41], MFEA-GHS [42], and MFGP [31]. All the above algorithms transfer knowledge through assortative mating and vertical cultural transmission [43]. In assortative mating, two individuals are selected from the population randomly and then generate offspring by simulated binary crossover (SBX) and polynomial mutation. In vertical cultural transmission, each offspring is randomly assigned to a task. In KTS, each individual provides different knowledge for the tasks, and the diversity of population is maintained effectively. However, EMT algorithms with KTS cannot fully utilize the knowledge of high-quality solutions to accelerate the convergence rate of the population, due to the randomness of knowledge transfer.

Knowledge transfer methods based on multiple individuals (KTM) refer to extracting knowledge from multiple individuals of one task, and transferring knowledge to other tasks. EMT algorithms with KTM can be implemented based on particle swarm optimization (PSO) [44], [45]. For example, Feng *et al.* [46] proposed a multitasking PSO algorithm. The convergence of the population is accelerated by the guidance of optimal solutions of multiple tasks. Tang and Gong [47] proposed an adaptive multitasking PSO algorithm by introducing a self-adaption strategy to adjust the intertask knowledge transfer probability, which reduces the probability of negative transfer effectively. Song *et al.* [48] proposed a multitasking multiswarm optimization algorithm. The quality of the solutions is improved by a crossover between optimal individuals of all tasks.

Differential evolution (DE) [49], [50] is another popular platform to construct EMT with KTM. For instance, Liu *et al.* [51] proposed SaM-MA, with three different mechanisms employed, i.e., DE algorithm, predicting optimal solution via surrogate model, and local searching strategy. The

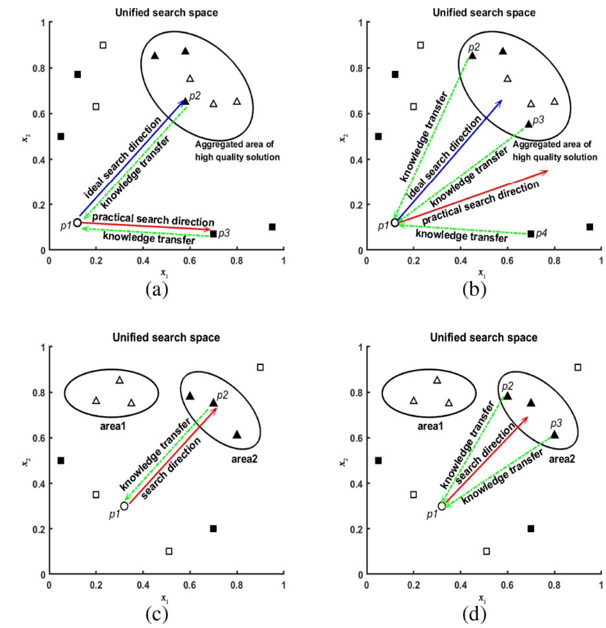


Fig. 1. Illustration of knowledge transfer and search direction, where x_1 and x_2 are the two dimensions of the decision variables. Triangles and squares represent the high-quality solutions and inferior solutions, respectively. The solutions of task1 and task2 are distinguished with hollow and solid faces, respectively. The global optima are supposed to be located in the ellipse areas. (a) and (b) KTS and KTM at the HS scenario, respectively. (c) and (d) depict KTS and KTM at the LS scenario, respectively.

mixed use of these three mechanisms can balance the diversity and convergence of population. Zhou *et al.* [52] proposed a new mutation strategy called DE/best/1+ρ. With gradually increased weight of knowledge transfer in the process of evolution, it improves the diversity of population. Liang *et al.* [53] proposed an MOMFEA based on subspace alignment and self-adaptive DE, which can improves the quality of knowledge transfer among the tasks.

Explicit knowledge transfer represents another form of KTM in EMT algorithms. Feng *et al.* [25] first proposed an EMT algorithm called EMT-EGT with explicit knowledge transfer, which allows the incorporation of multiple search mechanisms with different biases. It is able to improve the search ability of the population. Shang *et al.* [54] developed a credit assignment approach, which selects proper individuals for explicit knowledge transfer. The efficiency of knowledge transfer of this method was shown to be improved.

In a word, KTM is beneficial to improve the convergence of population, but it also has the probability of extracting knowledge from inferior individuals. Besides, the search of the entire population at one generation is guided by the same solutions in some KTMs, which increases the probability of falling into local optima.

B. Motivation of Two-Stage Adaptive Knowledge Transfer Based on Population Distribution

In this section, we take an example of two-tasking problem as shown in Fig. 1 to explain the motivation of the proposed algorithm with two-stage adaptive knowledge transfer based on population distribution.

Algorithm 1 Framework of EMT-PD

```
Input: R, type of probability model.
Output: a series of non-dominated solutions.
Initialize the population P;
Split P into two subpopulations pop_1 and pop_2;
while stopping conditions are not satisfied do
    Build probability model of pop_1 and pop_2 by Algorithm 2;
    Calculate the maximum points of the product of two probability models mp according to Eq. (6);
    Conduct two-stage adaptive knowledge transfer and generate the offspring population C by Algorithm 3;
    Evaluate C;
    Perform environmental selection;
end while
```

For high similar problems, knowledge extracted from the high-quality solutions of one task can usually accelerate the convergence of another task effectively [39]. However, in most of the existing KTSs and KTMs [21], [23], [41], knowledge may be extracted from inferior individuals. Fig. 1(a) shows an example of KTS at a high similarity (HS) scenario. In the search space, the high-quality solutions of task1 and task2 converge in the same area, where $p1$ is a solution of task1, $p2$ is a high-quality solution of task2, and $p3$ is an inferior solution of task2. If knowledge is transferred from $p2$ to $p1$, then $p1$ can be guided to search the centralized area of high-quality solutions. However, KTS may transfer knowledge from inferior solution $p3$ to $p1$, which would slow down the convergence of task1. Fig. 1(b) shows an example of KTM at a HS scenario, where a solution is guided by two other solutions. Here, $p1$ is a solution of task1. $p2$ and $p3$ are two high-quality solutions of task2, respectively. $p4$ is an inferior solution of task2. If knowledge is transferred from $p2$ and $p3$ to $p1$, $p1$ can be leaded to the centralized area of high-quality solutions. However, KTM may also transfer the knowledge of the inferior solution $p4$–$p1$. If $p1$ learns from $p3$ and $p4$, it would deviate from the ideal search direction. A representative instance showing this kind of phenomenon is also provided in Fig. S1(c) of the supplementary material. On the contrary, in some KTS and KTM methods [46], [53], the search of the entire population at each generation is only guided by one or several best solutions of the tasks at each time. If the selected solutions used for transfer are not the globally optimal solutions, it easily leads the population to the local optima as shown in Fig. S1(f) of the supplementary material.

For low similar problems, the population distributions of task1 and task2 are very different [55]. For both KTS and KTM, the populations of different tasks are inappropriate to guide each other to search with an ideal direction. There would be a high probability of negative transfer between those tasks. Fig. 1(c) shows an example of KTS at low similarity (LS) scenario, where high-quality solutions of different tasks are distributed in different areas. Here, $p1$ represents an individual of task1 and $p2$ is a high-quality solution of task2. When knowledge is transferred from $p2$ to $p1$, $p1$ deviates severely from the convergence area of high-quality solutions for task1.

Algorithm 2 Build Probability Model

Input: *pop*, the subpopulation of task1 or task2; R, the type of probability model.

Output: $\boldsymbol{M}$, the probabilistic model; $\boldsymbol{m}$, maximum point of $\boldsymbol{M}$.

```
for each dimension of decision variable do
    Generate the log-likelihood LL(θ_j) by Eq. (2);
    Build the probability model M_j for pop by Eq. (3);
    Calculate the maximum point m_j of M_j by Eq. (5);
end for
```

Fig. 1(d) shows an example of KTM at an LS scenario, which is similar to the situation of KTS. Fig. S1(l) in the supplementary material gives a representative instance, which has this kind of phenomenon.

To address the above issues, this article presents a novel EMT algorithm with two-stage adaptive knowledge transfer based on population distribution. Specifically, one probability model is first built for the population of each task, and the knowledge used for transfer is extracted from the maximum point of the product of the two probability models. Note that the population distribution rather than individual(s) is used as the knowledge source to relieve the impact of inferior individuals. The effectiveness of the motivation is demonstrated in the comparison with other algorithms using various KTS and KTM strategies in Sections IV and II of the supplementary material.

III. Proposed Algorithm

Multiobjective optimization algorithms have been implemented in many practical applications [1]–[3]. However, the potential synergies between distinct optimization problems have not been fully explored in traditional multiobjective optimization algorithms. In this section, we propose a multitasking algorithm EMT-PD with two-stage knowledge transfer based on population distribution to exploit the potential synergies between problems. We first introduce the main framework of EMT-PD. Then, the process of building the probability model and extracting knowledge is described. Afterward, the details of the two-stage adaptive knowledge transfer are explained. Finally, the computational complexity of EMT-PD is analyzed.

A. Main Framework of EMT-PD

The main framework of EMT-PD is outlined in Algorithm 1. First, the population $\boldsymbol{P}$ is initialized and divided into two subpopulations $\boldsymbol{pop}_1$ and $\boldsymbol{pop}_2$ for different tasks. In each evolutionary generation, a probability model is built for each task by Algorithm 2 in line 4. In line 5, the maximum point $\boldsymbol{mp}$ of the product of probability models is obtained by (6), which is used to guide the search of population. The two-stage adaptive knowledge transfer and offspring generation are carried out by Algorithm 3 in line 6. It is worth noting that EMT-PD is a general algorithm, which supports various types of probability models. In the case where the decision space dimensions of two tasks are different, EMT-PD pads the corresponding solutions with random variables to ensure the solutions of both tasks have equal decision space dimensionality, which is similar to other EMT algorithms [25].

Algorithm 3 Two-Stage Adaptive Knowledge Transfer and Offspring Generation

Input: $\boldsymbol{pop}$, the subpopulation of task; $\boldsymbol{mp}$, the maximum point of product of two probability models; $\boldsymbol{m}$, the maximum point of the probability model of $\boldsymbol{pop}$.

Output: $\boldsymbol{C}$, offspring population.

```
Calculate the Euclidean distance d_1 of m and mp according to Eq. (7);
for each individual p ∈ pop do
    Calculate the Euclidean distance d_2 of m and p according to Eq. (8);
    Generate an intermediate individual p' according to Eq. (9);
    Generate an offspring c according to p' by Eq. (11);
    c' = Polynomial mutation(c);
    C = C ∪ c';
end for
```

B. Building Probability Model

In this article, the maximum likelihood estimation (MLE) is used to estimate the parameters of the probability model according to population distribution, and the maximum point of the probability model is thus obtained.

Let $\boldsymbol{M}$ denote the probability models of the population distribution. θ_j is the parameter of $\boldsymbol{M}$ in the jth dimension of decision variable. The MLE is used to obtain the best estimation $\hat{\theta}_j$ of θ_j on which the likelihood $\mathrm{LL}(\theta_j)$ is maximized. First, $\mathrm{LL}(\theta_j)$ is calculated as follows:

$$\mathrm{LL}(\theta_j) = \sum_{i=1}^{N} \ln f\left(p_{i,j}|\theta_j\right) \tag{2}$$

where $p_{i,j}$ represents the jth variable of the ith individual for $i = 1, 2, \ldots, N$ with N denoting the population size. The type of probability model $f(p_{i,j}|\theta_j)$ is R, and ln denotes the natural logarithm.

Then, $\hat{\theta}_j$ is calculated as follows:

$$\hat{\theta}_j = \arg\max_{\theta_j} \mathrm{LL}\left(\theta_j\right). \tag{3}$$

Based on the above calculations, we can obtain the probability model of the jth decision variable M_j, i.e., $f(p_{i,j}|\hat{\theta}_j)$, which also can be labeled as follows:

$$M_j(x) = f\left(x \mid \hat{\theta}_j\right). \tag{4}$$

The maximum point $\boldsymbol{m}$ of probability model $\boldsymbol{M}$ learned from population reflects the centralization of population in each generation of the evolution [56]. The formula of calculating m_j is given by

$$m_j = \arg\max_{x} M_j(x). \tag{5}$$

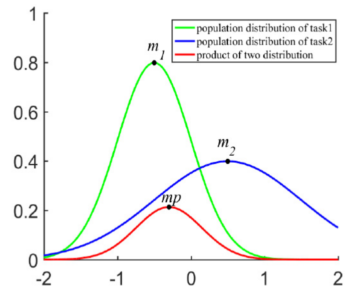


Fig. 2. Maximum point of the product of probability models.

Algorithm 2 shows the process of building the probability models by MLE. For each dimension of the decision variable, the log-likelihood $\mathrm{LL}(\theta_j)$ is calculated by (2) in line 2. Then, the parameters of the probability model M_j are calculated via (3) in line 3. After that, the maximum point m_j of M_j is obtained according to (5) in line 4.

Finally, the probability model $\boldsymbol{M}$ as well as the maximum point $\boldsymbol{m}$ of $\boldsymbol{M}$ is obtained, where $\boldsymbol{M} = \{M_1, M_2, \ldots, M_j, \ldots, M_D\}$, $\boldsymbol{m} = (m_1, m_2, \ldots, m_j, \ldots, m_D)$, and D is the dimension of the decision variable.

C. Extracting Knowledge From Population Distribution

The maximum point of the product of probability models reflects the common centralization of two populations, which gives expression to the common search trend of two populations in the evolutionary process [56]. In our algorithm, the maximum point is used to guide the search of each task. It is helpful to accelerate the convergence rate and decrease the probability of falling into the local optimum.

According to Algorithm 2, the two probability models $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ can be learned from the populations of task1 and task2, respectively, where $\boldsymbol{M}_1 = \{M_{1,1}, M_{1,2}, \ldots, M_{1,j}, \ldots, M_{1,D}\}$ and $\boldsymbol{M}_2 = \{M_{2,1}, M_{2,2}, \ldots, M_{2,j}, \ldots, M_{2,D}\}$. The maximum point $\boldsymbol{mp} = (mp_1, mp_2, \ldots, mp_j, \ldots, mp_D)$ of the product of $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ can be calculated as follows:

$$mp_j = \arg\max_x \big(M_{1,j}(x) * M_{2,j}(x)\big). \tag{6}$$

As shown in Fig. 2, the green curve denotes the population distribution of task1. The blue curve indicates the population distribution of task2. $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ are the maximum points of the two distributions, respectively. The red curve represents the product of the population distributions of task1 and task2, which reflects the common search trend of task1 and task2. The maximum point $\boldsymbol{mp}$ of red curve reflects the common centralization of the populations of task1 and task2, and $\boldsymbol{mp}$ is used to guide the search of each population.

D. Two-Stage Knowledge Transfer and Offspring Generation

For high similar problems, the populations of task1 and task2 tend to converge to an analogous area, and the convergence rate of task1 and task2 can be accelerated with the guide of $\boldsymbol{mp}$ directly. But for low similar problems, the populations of task1 and task2 would converge to different areas. If $\boldsymbol{mp}$ is directly used to guide the search of the populations, it very likely results in negative transfers. In addition, it is necessary to balance the population convergence and the diversity on both high and low similar problems. To solve these issues, we propose a two-stage adaptive knowledge transfer based on the population distribution. At the first stage, the search step size of each individual is adjusted adaptively to reduce the impact of negative transfer. At the second stage, the search range of each individual is further adjusted based on an intermediate individual, which increases the population diversity of the population and avoids getting trapped in local optimum.

Algorithm 3 presents the pseudocode of the two-stage adaptive knowledge transfer and the offspring generation for one task. First, the Euclidean distance d_1 between $\boldsymbol{mp}$ and the maximum point $\boldsymbol{m}$ of the task's probability model is calculated in line 1. Then, for each individual p, the Euclidean distance d_2 between p and $\boldsymbol{m}$ is calculated in line 3. d_1 and d_2 are calculated as follows:

$$d_1 = \sqrt{(\boldsymbol{m} - \boldsymbol{mp})(\boldsymbol{m} - \boldsymbol{mp})^T} \tag{7}$$

$$d_2 = \sqrt{(\boldsymbol{m} - p)(\boldsymbol{m} - p)^T}. \tag{8}$$

In line 4, the intermediate individual p' is generated via the first stage of knowledge transfer. p' is calculated is as follows:

$$p' = p + w(\boldsymbol{mp} - p) \tag{9}$$

where w is the adaptive weight of knowledge transfer, defined as follows:

$$w = d_2/(d_1 + d_2). \tag{10}$$

As $\boldsymbol{m}$ reflects the centralization of the population which p belongs to, if $\boldsymbol{mp}$ is close to $\boldsymbol{m}$, the population distributions of the two tasks are similar, and the knowledge transfer from $\boldsymbol{mp}$ can effectively guide the search of p. Therefore, it is suitable to increase the weight of knowledge transfer w when $\boldsymbol{mp}$ is close to $\boldsymbol{m}$. In (10), d_1 indicates the distance between $\boldsymbol{mp}$ and $\boldsymbol{m}$. The smaller d_1 is, the larger w is. On the contrary, if $\boldsymbol{m}$ is far away from $\boldsymbol{mp}$, the difference between the population distributions of two tasks becomes large, the knowledge transfer from $\boldsymbol{mp}$ could be negative transfer. It is necessary to decrease the weight w of knowledge transfer in such situation. As can be seen from (10), the larger d_1 is, the smaller w is. In addition, d_2 is used to measure the distance between an individual p and the centralization of the population. Smaller d_2 means that p is closer to the centralization of the population. The weight w should be decreased to reduce the transfer distance of p. Consequently, w becomes smaller in (10). On the contrary, if d_2 is large, it suggests that p is far away from the centralization of the population. The weight w should be increased to enlarge the transfer distance of p.

In line 5 of Algorithm 3, an offspring c is generated via the second stage of knowledge transfer as follows:

$$c = p' + v \tag{11}$$

where p' is an intermediate individual generated in the first stage of knowledge transfer, and v is a search vector defined by

$$v = \frac{1}{D} * F * Q * (d_1 + d_2) \tag{12}$$

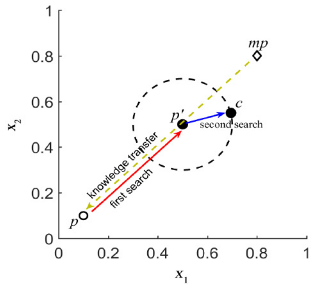


Fig. 3. Example of two-stage knowledge transfer in the 2-D unified searching space.

where D is the dimension of decision variable, Q is the D-dimensional Gaussian white noise, and F is the scaling factor. The Gaussian white noise Q adopted in this article follows a standard Gaussian distribution, and its mean and variance are set to 0 and 1, respectively. The scaling factor F is used to adjust the order of the magnitude of the white noise, which is introduced to increase population diversity.

The reason for introducing the second-stage knowledge transfer is to better balance the population convergence and diversity as all individuals search in the same direction guided by $\boldsymbol{mp}$ at the first stage. If d_1 is small, the knowledge transfer at the first stage is effective; thanks to the HS between the two tasks. It is unnecessary to substantially adjust the search range at the second stage. On the contrary, if d_1 is large, i.e., the similarity between two tasks is low, the adjustment of search range at the second stage should be reasonably increased. As for d_2, if it is small, the individual p is close to the convergence area. It also needs no substantial adjustment at the second stage. If d_2 is large, the adjustment of search range at the second stage should also be reasonably increased. At the end of Algorithm 3, polynomial mutation is performed in line 6, and the offspring c' is combined into the offspring population $\boldsymbol{C}$ in line 7.

Fig. 3 shows an example of the two-stage knowledge transfer, where p is an individual selected from the population of a certain task, and $\boldsymbol{mp}$ is the maximum point of the product of the probability models. At the first stage of knowledge transfer, the knowledge is transferred from $\boldsymbol{mp}$ to p. The search of p is guided by $\boldsymbol{mp}$, and an intermediate individual p' is generated at this stage. At the second stage, the search range of the intermediate individual p' is dynamically adjusted again to generate an offspring c.

E. Computational Complexity Analysis

In this section, the computational complexity of EMT-PD within a generation is analyzed. EMT-PD mainly consists of three parts: 1) building probability model; 2) two-stage knowledge transfer and offspring generation; and 3) environmental selection. The MLE is used to build the probability models, which calls for a computational complexity of $O(DN)$. In the two-stage knowledge transfer and offspring generation, a computational complexity of $O(DN)$ is needed. In the environmental selection, the computational complexities of nondominated sorting and crowding distance are $O(nN^2)$ and $O(nN\log(N))$, respectively. Overall, the computational complexity of each generation of EMT-PD is $O(nN^2)$, where D and N represent the dimension of the decision variable and population size, respectively.

IV. Experiment and Analysis

The performance of EMT-PD is first evaluated and compared to several state-of-the-art optimization algorithms on multitasking multiobjective test suites. Then, a new multitasking many-objective test suite is designed to challenge EMT-PD (the key experimental results are presented in this section; whereas, due to the page limit, some empirical studies and discussions are provided in the supplementary material).

A. Experiments on Multiobjective Problems

1) Test Problems and Compared Algorithms: To assess the performance of EMT-PD, two multiobjective test suites are applied.

1) The first test suite is the classical multitasking multiobjective test suite MTMOPs [57], of which the test problems can be split into three groups according to the degree of intersection, i.e., complete intersection (CI), partial intersection (PI), and no intersection (NI). The groups can be further partitioned into problems with HS, medium similarity (MS), and LS. Therefore, there are nine types of problems, namely, CIHS, CIMS, CILS, PIHS, PIMS, PILS, NIHS, NIMS, and NILS.
2) The second test suite is the complex multitasking multiobjective optimization test suite CEC2019-CMO proposed in the IEEE CEC2019 competition on EMT optimization, which contains ten multitasking multiobjective problems. The details of CEC2019-CMO are summarized in [58].

Six state-of-the-art algorithms are involved in the comparison with EMT-PD, including three MOMFEAs, i.e., MO-MFEA [23], TMO-MFEA [24], and EMT-EGT [25], and three representative multiobjective evolutionary algorithms, namely, NSGA-II [17], AR-MOEA [9], and CMOPSO [59]. NSGA-II is a basic multiobjective evolutionary algorithm serving as the baseline here. AR-MOEA is an indicator-based multiobjective optimization algorithm with an enhanced inverted generational distance (IGD) indicator. CMOPSO is a multiobjective PSO algorithm with the competitive mechanism.

2) Performance Metric: The IGD [60] is widely used to evaluate the performance of multiobjective evolutionary algorithms. It is calculated as follows:

$$\mathrm{IGD}\big(P^{*}, \boldsymbol{A}\big) = \frac{1}{|P^{*}|}\sum_{z\in P^{*}} \min d(z, \boldsymbol{A}) \tag{13}$$

where $d(z, \boldsymbol{A})$ refers to the Euclidean distance between a reference point z and a solution set $\boldsymbol{A}$ in the objective space, and P^* represents a predefined set of reference points on the PF. The smaller IGD is, the better the convergence and diversity of $\boldsymbol{A}$ is.

3) Parameter Settings: In EMT-PD, the type of probability model R is set to the Gaussian probability model, and the scale factor F is set to 0.01. The parameters of TMO-MFEA

TABLE I
PARAMETER SETTINGS OF THE ALGORITHMS

Parameter	Value
Size of population (N)	200
Maximal iteration (G)	1000
Maximal function evaluations (EFs)	200000
Crossover probability (p_c)	0.3
Mutation probability (p_m)	$1/N$
Distribution index for crossover (η_c)	20
Distribution index for mutation (η_m)	20

are set according to [24], i.e., *rmp* is set to 1 for the diversity variable and 0.3 for the convergence variable. According to the description of EMT-EGT in [25], SPEA2 is used for one task and NSGA-II is used for the other task. The interval of explicit transmission is set to 10. The size of the elite population λ of CMOPSO is set to 10 according to [59]. The common parameter settings of the algorithms are summarized in Table I.

It is worth noting that F is set to a lower weight for two main reasons. The first is that the scale of decision variable space is between 0 and 1 after normalization. If a relatively high noise is chosen, it may cause the individual to exceed the search range of decision variable. The second is that an individual experienced the first knowledge transfer has moved toward the center area of the high-quality solution. If a relatively large disturbance is introduced, it easily drives the individual to jump out of the center area of the high-quality solutions, leading to performance deterioration. The sensitivity analysis of F in Section V of the supplementary material also justify the setting of F.

4) Results on MTMOPs: Table II shows the experimental results on MTMOPs where the best result is highlighted. The Wilcoxon rank sum test is performed at a significance level of 5%. The symbols "+," "−," and "≈" denote that the result is significantly better, significantly worse, and comparable with that of EMT-PD, respectively.

EMT algorithms work well on high similar problems CI+HS, PI+HS, and NI+HS. The overall performance of EMT algorithms is better than single-tasking algorithms. The results confirm that knowledge transfer across tasks in EMT is capable of accelerating convergence and finding better solutions.

On medium similar problems, such as CI+MS, PI+MS, and NI+MS, EMT algorithms except for EMT-PD are not very competitive. We note that CI+MS, PI+MS, and NI+MS are comprised of one unimodal function and one multimodal function. The unimodal function has only one single global optimal solution. On the unimodal function, the solution population can gradually converge to a small area of search space. Whereas, the multimodal function has multiple global optimal solutions or local optima, where the solution population tend to converge to multiple different areas of the search space. Since the convergence areas of the populations on the unimodal and multimodal functions are quite different, the probability of negative transfers is high. MO-MFEA, TMO-MFEA, and

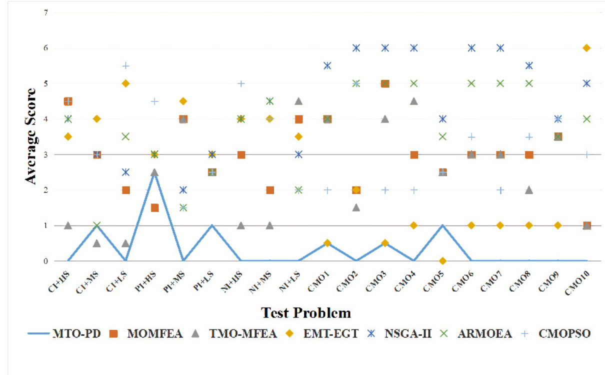


Fig. 4. Average performance score of all algorithms on MTMOPs and CEC2019-CMO. The values of EMT-PD are connected by a solid line to assess the score more easily.

EMT-EGT cannot effectively handle these kinds of negative transfer. Nevertheless, EMT-PD can reduce the probability of negative transfer and increase the population diversity through the two-stage adaptive knowledge transfer. It thus achieves a good performance on the medium similar problems.

EMT algorithms are competitive on CI+LS and PI+LS, although they belong to low similar problems. Because the global optima of the two tasks in CI+LS and PI+LS problems have identical variables in the unified search space, the two tasks can still provide useful knowledge to each other on identical variables. On the contrary, on the NI+LS problem, all global optima variables of the two tasks are different, which poses great challenges for EMT algorithms. EMT-PD manages to solve NI+LS thanks to the adaptively adjusted search step size at the first stage of knowledge transfer.

5) Results on CEC2019-CMO: Table III presents the experimental results on CEC2019-CMO. The overall performance of EMT-PD is better than that of the other algorithms. CMO2, CMO4, and CMO10 are composed of functions with the same PS. Therefore, the population diversity and convergence can be maintained well via knowledge transfer between tasks. Compared to single-tasking algorithms, the performance of EMT algorithms is superior. For CMO1, CMO3, and CMO6, the objective functions of the two tasks are very similar, i.e., the two tasks have similar properties. EMT algorithms can effectively optimize such problems. CMO7, CMO8, and CMO9 have different PSs and objective functions. The performance of the other EMT algorithms on these problems is unsatisfactory. However, based on the adaptive search step size at the first stage of knowledge transfer, EMT-PD can effectively reduce the probability of negative transfer and achieve good results. The PFs of the two tasks in CMO5 are very complex. To achieve better performance, the population diversity must be carefully maintained in the optimization process. EMT-EGT applies multiple searching mechanisms for the population, which is conducive to maintain the diversity of population and, hence, obtains the best performance on CMO5.

Fig. 4 shows the average performance score of all algorithms on MTMOPs and CEC2019-CMO. The average performance score is defined in [61] with the lower value indicating the

TABLE II
AVERAGED IGD VALUE OBTAINED BY EMT-PD, MO-MFEA, TMO-MFEA, EMT-EGT, NSGA-II, ARMOEA, AND CMOPSO ON OVER 30 INDEPENDENT RUNS ON MTMOPs

Problem	Task	EMT-PD	MO-MFEA	TMO-MFEA	EMT-EGT	NSGA-II	ARMOEA	CMOPSO
CI+HS	T1	**9.10E-04**	1.43E-02(-)	9.14E-03(-)	1.24E-03(-)	4.42E-01(-)	3.11E-01(-)	2.59E-01(-)
	T2	**9.66E-03**	3.22E-01(-)	6.58E-02(-)	6.62E+01(-)	3.53E-01(-)	3.83E-01(-)	2.34E-01(-)
CI+MS	T1	6.84E+00	7.34E+00(-)	1.64E+01(-)	5.34E+00(+)	8.47E+00(-)	**4.92E+00**(+)	7.17E+00(-)
	T2	**2.76E-03**	4.84E-03(-)	4.72E-03(-)	4.35E-03(-)	8.09E-03(-)	4.72E-03(-)	4.62E-03(-)
CI+LS	T1	**2.07E-03**	1.58E-02(-)	2.04E-02(-)	2.02E-02(-)	2.92E+01(-)	9.96E+00(-)	8.95E+00(-)
	T2	**9.54E-04**	6.59E-02(-)	2.04E-03(-)	2.21E+00(-)	2.23E-01(-)	2.26E-01(-)	2.23E-01(-)
PI+HS	T1	**1.46E-01**	1.49E-01(-)	1.29E-01(-)	2.43E-01(-)	2.01E-01(-)	1.94E-01(-)	2.25E-01(-)
	T2	1.30E+01	2.28E+01(-)	2.11E+01(-)	2.06E+01(-)	2.48E+01(-)	1.18E+01(+)	**1.18E+01**(+)
PI+MS	T1	**1.33E-02**	3.39E-02(-)	2.27E-01(-)	2.98E-02(-)	4.41E-02(-)	1.39E-02(-)	1.39E-02(-)
	T2	**2.90E+01**	7.92E+02(-)	5.21E+02(-)	5.80E+02(-)	5.80E+02(-)	4.47E+02(-)	2.92E+02(-)
PI+LS	T1	1.57E-02	3.23E-02(-)	2.26E-02(-)	**5.31E-03**(+)	7.70E-02(-)	6.70E-03(+)	5.92E-03(+)
	T2	**1.73E+01**	3.81E+01(-)	2.59E+01(-)	2.09E+01(-)	5.77E+01(-)	1.73E+01(≈)	1.73E+01(≈)
NI+HS	T1	**5.09E-02**	4.94E+01(-)	5.00E+01(-)	4.73E+01(-)	3.56E+02(-)	8.80E+01(-)	1.48E+01(-)
	T2	**1.60E-03**	4.35E-02(-)	1.71E-02(-)	1.66E-02(-)	6.74E-01(-)	2.05E-01(-)	2.25E-01(-)
NI+MS	T1	**7.28E+00**	2.40E+01(-)	1.72E+01(-)	1.62E+01(-)	6.82E+01(-)	2.85E+01(-)	2.37E+01(-)
	T2	**9.89E-04**	1.17E+00(-)	8.16E-03(-)	7.75E-03(-)	4.05E+00(-)	3.28E+00(-)	2.70E+00(-)
NI+LS	T1	**6.27E-03**	6.53E-02(-)	2.07E-02(-)	2.87E-02(-)	6.66E-02(-)	1.10E-02(-)	1.09E-02(-)
	T2	**6.08E-02**	2.06E+01(-)	2.10E+01(-)	2.15E+01(-)	3.40E+01(-)	1.93E+01(-)	2.06E+01(-)
$+/-/\approx$			0/18/0	0/18/0	2/16/0	0/18/0	3/14/1	2/15/1

TABLE III
AVERAGED IGD VALUE OBTAINED BY EMT-PD, MO-MFEA, TMO-MFEA, EMT-EGT, NSGA-II, ARMOEA, AND CMOPSO OVER 30 INDEPENDENT RUNS ON CEC2019-CMO

Problem	Task	EMT-PD	MO-MFEA	TMO-MFEA	EMT-EGT	NSGA-II	ARMOEA	CMOPSO
CMO1	T1	3.78E-03	7.06E-03(-)	7.08E-03(-)	**3.72E-03**(+)	7.98E-03(-)	6.44E-03(-)	5.75E-03(-)
	T2	**3.12E-03**	3.49E-02(-)	3.34E-02(-)	3.51E-02(-)	4.01E-02(-)	5.61E-02(-)	1.72E-02(-)
CMO2	T1	**3.27E-03**	5.34E-03(-)	5.24E-03(-)	3.81E-03(-)	8.00E-03(-)	6.29E-03(-)	5.80E-03(-)
	T2	**5.51E-03**	1.40E-02(-)	1.36E-02(-)	2.93E-02(-)	1.91E-01(-)	2.52E-01(-)	6.49E-02(-)
CMO3	T1	1.25E-02	8.55E-02(-)	7.58E-02(-)	**1.02E-02**(+)	1.14E-01(-)	1.21E-01(-)	5.85E-02(-)
	T2	**7.77E-03**	2.90E-02(-)	2.71E-02(-)	1.82E-02(-)	4.05E-02(-)	4.50E-02(-)	2.30E-02(-)
CMO4	T1	**9.87E-03**	6.46E-02(-)	4.31E-02(-)	1.37E-02(-)	1.13E-01(-)	1.30E-01(-)	6.09E-02(-)
	T2	**1.04E-02**	6.60E-02(-)	4.64E-02(-)	1.79E-02(-)	1.44E-01(-)	1.48E-01(-)	5.53E-02(-)
CMO5	T1	8.78E-03	3.97E-02(-)	3.97E-02(-)	**8.16E-03**(≈)	5.23E-02(-)	6.26E-02(-)	3.91E-02(-)
	T2	1.87E-02	4.14E-01(-)	3.70E-01(-)	**1.37E-02**(+)	1.66E-01(-)	1.33E-01(-)	4.08E-01(-)
CMO6	T1	**7.73E-03**	2.95E-02(-)	2.97E-02(-)	1.36E-02(-)	4.91E-02(-)	6.71E-02(-)	4.45E-02(-)
	T2	**1.05E-02**	8.71E-02(-)	6.91E-02(-)	3.61E-02(-)	1.30E-01(-)	1.26E-01(-)	5.33E-02(-)
CMO7	T1	**8.06E-03**	3.02E-02(-)	3.00E-02(-)	8.27E-03(-)	3.94E-02(-)	4.52E-02(-)	2.36E-02(-)
	T2	**7.92E-03**	3.02E-02(-)	2.96E-02(-)	1.08E-02(-)	3.98E-02(-)	4.40E-02(-)	2.89E-02(-)
CMO8	T1	**8.95E-03**	3.69E-02(-)	3.39E-02(-)	1.22E-02(-)	4.00E-02(-)	4.85E-02(-)	2.89E-02(-)
	T2	**1.64E-02**	1.55E-01(-)	1.54E-01(-)	2.62E-02(-)	3.82E-01(-)	4.46E-01(-)	2.62E-01(-)
CMO9	T1	**1.84E-02**	3.59E-01(-)	3.58E-01(-)	3.16E-02(-)	2.93E-01(-)	2.57E-01(-)	7.01E-01(-)
	T2	**1.01E-02**	9.71E-02(-)	9.42E-02(-)	4.60E-02(-)	1.55E-01(-)	1.37E-01(-)	5.31E-02(-)
CMO10	T1	**1.72E-02**	1.51E-01(-)	1.32E-01(-)	5.95E-02(-)	3.91E-01(-)	4.46E-01(-)	2.42E-01(-)
	T2	**1.72E-02**	1.53E-01(-)	1.37E-01(-)	9.67E-02(-)	3.08E-01(-)	3.79E-01(-)	1.66E-01(-)
$+/-/\approx$			0/20/0	0/20/0	3/16/1	0/20/0	0/20/0	0/20/0

better performance an algorithm achieved. EMT-PD shows the best overall results on MTMOPs and CEC2019-CMO.

B. Experiments on Many-Objective Problems

1) Test Problems and Compared Algorithms: MaF [35] is the newly proposed many-objective optimization test suite considered more related to real-world problems than other many-objective optimization test suites. In order to further validate the performance of EMT-PD on MaOPs, we design a novel multitasking many-objective test suite based on MaF [35], abbreviated as MTMaOPs. The proposed MTMaOPs includes six problems, as shown in Table IV. MaF-HS1 is composed of MaF3 and MaF4. MaF-HS2 is composed of MaF4 and MaF6. MaF-MS1 is composed of MaF1 and MaF5*, where MaF5* is a shifted version of MaF5. The shifted individual is $z = (p - \boldsymbol{r})$, where $\boldsymbol{r} = (0.05)^D$ and the original individual $p \in (0, 1)^D$. MaF-MS2 consists of MaF5 and MaF6*, where MaF6* uses

TABLE IV
SUMMARY OF THE PROPOSED MTMaOPs

Problem	Task	Function	sim with $n=10$	sim with $n=20$	sim with $n=30$
MaF-HS1	T1 T2	MaF3 MaF4	1	1	1
MaF-HS2	T1 T2	MaF4 MaF6	1	1	1
MaF-MS1	T1 T2	MaF1 MaF5*	0.3703	0.3756	0.4236
MaF-MS2	T1 T2	MaF5 MaF6*	0.3865	0.3866	0.4396
MaF-LS1	T1 T2	MaF4 MaF5	0.0038	0.0051	0.0044
MaF-LS2	T1 T2	MaF3 MaF6	0.0038	0.0052	0.0051

TABLE V
PARAMETERS SETTING FOR DIFFERENT OBJECTIVE DIMENSION ON MTAMOPS

n	H	N	G	FEs
10	(3,1)	230	300	69000
20	(2,2)	420	300	126000
30	(2,1)	465	300	139500

TABLE VI
PARAMETER SETTINGS FOR CROSSOVER AND MUTATION ON MTAMOPS

Parameter	Value
Crossover probability (p_c)	0.3
Mutation probability (p_m)	$1/N$
Distribution index for crossover (η_c)	20
Distribution index for mutation (η_m)	20

the same shift operation as MaF5*. MaF-LS1 is composed of MaF4 and MaF5. MaF-LS2 is composed of MaF3 and MaF6. The similarity (denoted by sim) of the problem is calculated as follows [57]:

$$\text{sim} = \frac{\text{cov}(\boldsymbol{R}_1, \boldsymbol{R}_2)}{\text{std}(\boldsymbol{R}_1)\text{std}(\boldsymbol{R}_2)} \tag{14}$$

where $\boldsymbol{R}_1 = (R_{1,1}, R_{1,2}, \ldots, R_{1,k}, \ldots, R_{1,K})$ and $\boldsymbol{R}_2 = (R_{2,1}, R_{2,2}, \ldots, R_{2,k}, \ldots, R_{2,K})$ for $k = (1, 2, \ldots, K)$. The elements $R_{1,k}$ and $R_{2,k}$ are the ranks of the kth solution in the population with respect to the two tasks, respectively. To calculate the value of sim, 1 000 000 solutions are randomly generated, i.e., K =1 000 000.

The sim values lying in (0, 1/3], (1/3, 2/3], and (2/3, 1] indicate low, medium, and HS, respectively. According to the degree of similarity, MTMaOPs are divided into high similar problems, including MaF-HS1 and MaF-HS2, medium similar problems, including MaF-MS1 and MaF-MS2, and low similar problems, including MaF-LS1 and MaF-LS2. In addition, each problem of MTMaOPs is set to contain three instances of different objective dimensions, that is, $n = 10$, $n = 20$, and $n = 30$.

Six state-of-the-art algorithms are involved in the comparison with EMT-PD on MTMaOPs, including three EMT algorithms, i.e., MO-MFEA [23], TMO-MFEA [24], and EMT-EGT [25], and three many-objective evolutionary algorithms, i.e., NSGA-III [4], VaEA [62], and DDEANS [16]. NSGA-III is a basic many-objective evolutionary algorithms serving as the baseline here. VaEA is characterized by novel selection strategies. DDEANS is featured by a novel dynamical decomposition strategy.

2) Performance Metric: The modified inverted generational distance (IGD$_+$) [63] is adopted in this article as the performance evaluation measure on MTMaOPs. Since the IGD$_+$ takes the Pareto dominance relation between a reference vector and a solution into consideration, it can evaluate the performance of many-objective optimization algorithms more accurately. The IGD$_+$ value is calculated as follows:

$$\text{IGD}_+\big(P^*, \boldsymbol{A}\big) = \frac{1}{|P^*|}\sum_{z \in P^*} \min \sqrt{\sum_{k=1}^{n} (\max(\{z_k - A_k\}, 0))^2} \tag{15}$$

where $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)$ represents a reference vector, $\boldsymbol{A} = (A_1, A_2, \ldots, A_n)$ represents a solution set, n is the dimension of objectives, and P^* denotes a predefined set of reference points on the PF. In this experiment, the number of reference points for calculating IGD$_+$ is set to 100 000. The smaller value of the IGD$_+$, the better convergence and diversity of the solution set.

3) Parameter Settings: Because the decomposition-based algorithms NSGA-III and VaEA are subject to the reference vector, the population size of NSGA-III and VaEA cannot be set optionally. For a fair comparison, the same setting of the population size is employed for all algorithms following [64]. The details of the parameter settings for different objective dimension are summarized in Table V, where n indicates the number of objectives, H is the number of divisions considered along each objective coordinate, N represents the size of population, G is the maximal iteration, and FEs denote the maximal function evaluations. The common parameters of crossover and mutation are configured following the references [4], [7], [16], [19], [62], which are summarized in Table VI.

4) Results on MTMaOPs: Table VII reports the statistical results on MTMaOPs. Except for MaF-HS2 and MaF-MS1, EMT-PD performs better than other algorithms. The result show that the knowledge extraction and transfer of EMT-PD is efficient on MaOPs. The results on MaF-HS2 and MaF-MS1 are analyzed in detail as follows.

MaF-HS2 is a high similar problem and consists of MaF4 and MaF6. The property of HS can promote the cooperation between two tasks and improve the population diversity and convergence for EMT algorithms. It benefits all EMT algorithms to have obvious competitiveness on MaF-HS2 compared to single-tasking many-objective algorithms. MaF4 is

TABLE VII
AVERAGED IGD+ VALUE OBTAINED BY EMT-PD, MO-MFEA, TMO-MFEA, EMT-EGT, NSGA-III, VAEA, AND DDEANS OVER 30 INDEPENDENT RUNS ON MTMAOPS

Problem	Number of Objectives	Task	EMT-PD	MO-MFEA	TMO-MFEA	EMT-EGT	NSGA-III	VaEA	DDEANS
MaF-HS1	10	T1	**2.03E-01**	5.39E+05(-)	5.48E+05(-)	6.44E+05(-)	1.74E+06(-)	4.24E+04(-)	1.26E+03(-)
		T2	**5.57E-01**	2.08E+01(-)	2.81E+01(-)	8.83E+01(-)	7.52E+01(-)	2.16E+01(-)	3.08E+01(-)
	20	T1	**2.10E-01**	4.57E+02(-)	8.61E+02(-)	1.71E+02(-)	4.95E+03(-)	4.10E+04(-)	9.26E+02(-)
		T2	**8.13E-01**	1.98E+03(-)	8.86E+03(-)	1.23E+03(-)	3.65E+04(-)	7.98E+03(-)	1.31E+04(-)
	30	T1	**2.25E-01**	4.84E+03(-)	8.98E+03(-)	1.84E+03(-)	5.22E+03(-)	4.55E+04(-)	9.35E+02(-)
		T2	**9.14E-01**	1.98E+03(-)	8.86E+03(-)	1.23E+03(-)	3.65E+04(-)	7.98E+03(-)	1.31E+04(-)
MaF-HS2	10	T1	4.36E-01	2.00E+00(-)	**1.43E-01**(+)	2.06E-01(+)	4.04E+01(-)	2.74E+03(-)	9.53E+02(-)
		T2	**6.71E-01**	1.92E+00(-)	8.95E-01(-)	8.14E-01(-)	2.20E+00(-)	6.76E+00(-)	2.44E+00(-)
	20	T1	4.85E-01	1.14E+00(-)	4.05E-01(+)	**3.11E-01**(+)	3.65E+04(-)	7.98E+03(-)	1.31E+04(-)
		T2	**6.83E-01**	2.16E+01(-)	1.30E-01(-)	7.51E-01(-)	1.99E+00(-)	4.34E-01(-)	3.40E+00(-)
	30	T1	5.51E-01	1.20E+00(-)	3.87E-01(+)	**3.22E-01**(+)	2.93E+07(-)	7.30E+06(-)	1.15E+07(-)
		T2	**6.88E-01**	1.92E+01(-)	7.69E-01(-)	8.14E-01(-)	1.70E+00(-)	1.47E+00(-)	6.82E+00(-)
MaF-MS1	10	T1	6.12E-01	8.86E-01(-)	2.47E-01(+)	2.06E-01(+)	2.16E-01(+)	**1.67E-01**(+)	1.70E-01(+)
		T2	**6.24E-01**	4.12E+00(-)	2.12E+00(-)	4.38E+00(-)	1.06E+00(-)	1.43E+00(-)	1.08E+00(-)
	20	T1	4.90E-01	1.16E+00(-)	4.23E-01(+)	3.11E-01(+)	3.77E-01(+)	2.54E-01(+)	**2.51E-01**(+)
		T2	**6.02E-01**	3.41E+00(-)	3.31E+00(-)	4.85E+00(-)	1.55E+00(-)	1.80E+00(-)	1.51E+00(-)
	30	T1	6.41E-01	1.24E+00(-)	3.99E-01(+)	3.22E-01(+)	3.56E-01(+)	2.49E-01(+)	**2.48E-01**(+)
		T2	**7.88E-01**	3.53E+00(-)	3.84E+00(-)	4.95E+00(-)	1.69E+00(-)	1.89E+00(-)	1.69E+00(-)
MaF-MS2	10	T1	**6.09E-01**	1.07E+00(-)	4.12E+00(-)	4.11E+00(-)	1.08E+02(-)	1.06E+00(-)	1.43E+00(-)
		T2	**4.00E-03**	2.36E+01(-)	2.50E+00(-)	3.27E-01(-)	1.31E+00(-)	9.89E-01(-)	1.44E+00(-)
	20	T1	**7.62E-01**	3.53E+00(-)	4.90E+00(-)	4.12E+00(-)	1.55E+00(-)	1.79E+00(-)	1.51E+00(-)
		T2	**4.65E-03**	2.06E+01(-)	5.80E+00(-)	6.20E-01(-)	1.98E+00(-)	4.35E-01(-)	3.40E+00(-)
	30	T1	6.41E-01	3.65E+00(-)	4.96E+00(-)	4.96E+00(-)	1.69E+00(-)	1.89E+00(-)	1.68E+00(-)
		T2	**7.88E-01**	2.01E+01(-)	5.95E+00(-)	8.17E-01(-)	1.70E+00(-)	1.46E+00(-)	6.82E+00(-)
MaF-LS1	10	T1	**6.11E-01**	2.23E+05(-)	9.09E+03(-)	2.00E+04(-)	7.52E+01(-)	2.16E+01(-)	3.08E+01(-)
		T2	**6.22E-01**	4.14E+00(-)	2.22E+00(-)	8.64E+00(-)	1.06E+00(-)	1.43E+00(-)	1.08E+00(-)
	20	T1	**1.69E+00**	2.02E+05(-)	7.72E+04(-)	3.00E+04(-)	3.65E+04(-)	7.98E+03(-)	1.31E+04(-)
		T2	**5.04E-01**	3.77E+00(-)	3.10E+00(-)	1.03E+01(-)	1.55E+00(-)	1.80E+00(-)	1.51E+00(-)
	30	T1	**2.12E+00**	1.99E+06(-)	6.83E+04(-)	3.72E+05(-)	2.93E+06(-)	7.30E+05(+)	1.15E+06(-)
		T2	**6.13E-01**	3.74E+00(-)	3.70E+00(-)	1.11E+01(-)	1.69E+00(-)	1.89E+00(-)	1.68E+00(-)
MaF-LS2	10	T1	**1.95E-01**	4.92E+05(-)	8.90E+05(-)	7.28E+05(-)	1.74E+06(-)	4.24E+04(-)	1.26E+03(-)
		T2	**2.38E-03**	2.26E+01(-)	1.23E-01(-)	3.49E-01(-)	1.31E+00(-)	9.91E-01(-)	1.44E+00(-)
	20	T1	**2.56E-01**	4.13E+05(-)	1.04E+06(-)	1.70E+06(-)	4.95E+03(-)	4.10E+04(-)	9.26E+02(-)
		T2	**1.82E-03**	2.06E+01(-)	1.37E-01(-)	6.66E-01(-)	1.99E+00(-)	4.35E-01(-)	3.40E+00(-)
	30	T1	**3.38E-01**	4.14E+05(-)	1.13E+06(-)	2.38E+06(-)	1.74E+02(-)	1.01E+04(-)	6.59E+03(-)
		T2	**2.31E-03**	1.93E+01(-)	3.29E-01(-)	8.33E-01(-)	1.69E+00(-)	1.46E+00(-)	6.83E+00(-)
$+/-/\approx$				0/36/0	6/30/0	6/30/0	3/33/0	3/33/0	3/33/0

a badly scaled many-objective function. The traditional nondominated sorting employed by EMT-PD cannot normalize the value of the objective function. As a result, the population tends to converge to one side of PF and cannot guide the search of the population toward to convergence area effectively. It causes the poor performance of EMT-PD compared to EMT-EGT and TMO-MFEA.

MaF-MS1 is a medium similar problem and is composed of MaF1 and MaF5*. MaF1 is a linear problem, where the Pareto-optimal solutions mainly concentrate on a very small area of the search space. In the early stage of the evolution of EMT-PD, only a few individuals can accurately reflect the centralization of the population on MaF1. VaEA adopts the maximum-vector-angle-first principle in environmental selection to maintain the population diversity. DDEANS can dynamically balance the population diversity and convergence according to the Euclidean distance between the reference vector and the population. Owing to the effective diversity maintain mechanism, VaEA and DDEANS achieve competitive results on MaF-MS1. TMO-MFEA uses different crossover parameters for different variables and EMT-EGT applied a variety of search mechanisms to the population. The diversity mechanisms of these two algorithms are also superior to that of EMT-PD, which brings better performance of TMO-MFEA and EMT-EGT than EMT-PD on MaF-MS1.

Fig. 5 summarizes the average performance score of all algorithms on MTMaOPs. A smaller score indicates that an algorithm attains a better IGD_+ value. It can be seen that the single-tasking many-objective optimization algorithms achieves better performance than the EMT algorithms on most of the test problems, which suggests that the EMT algorithms proposed previously cannot effectively deal with the MaOPs. EMT-PD shows the competitiveness on MaOPs, because the

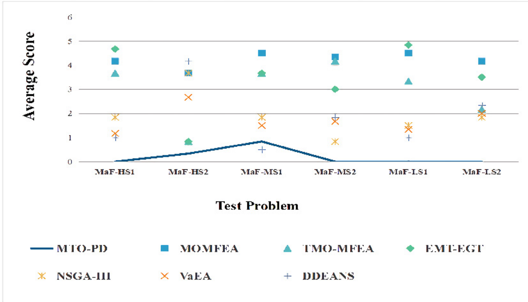


Fig. 5. Average performance score of all algorithms on MTMaOPs. The values of EMT-PD are connected by a solid line to assess the score more easily.

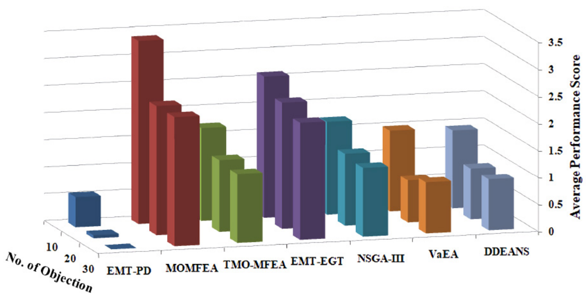


Fig. 6. Average performance score of all algorithms over 10, 20, and 30 objective dimensions of MTMaOPs.

search direction of each variable is guided by the population distribution, which leads to more accurate knowledge transfer.

The average performance scores of all algorithms over 10, 20, and 30 objective dimensions for MTMaOPs in term of the IGD$_+$ are shown Fig. 6. It can be seen that as the number of objectives increases, the average performance score of EMT-PD decreases, which indicates that EMT-PD has more advantages in handling optimization problems with larger objective dimensions.

V. Conclusion

This article proposed a new MOMFEA, namely, EMT-PD with two-stage adaptive knowledge transfer based on population distribution. The knowledge extracted from the probability model of the population distribution can effectively guide the search of the population. The first stage of knowledge transfer is characterized by a novel adaptive weight, which can effectively reduce the probability of negative transfer. At the second stage of knowledge transfer, the search range of each individual is adjusted dynamically to balance the population diversity and convergence and to help jumping across the local optimum. In order to challenge the EMT-PD, a new multitasking many-objective test suite MTMaOPs is proposed. EMT-PD is compared to other state-of-the-art algorithms on MTMOPs, CEC2019-CMO, and MTMaOPs. The experimental results showed the competitiveness of EMT-PD.

Although EMT-PD has shown superior performance on various test suites, there are still a few potential further works worth exploring. For example, the real-world problems could be very complex and the Gaussian probability model may not be the best model to fit a specific real-world problem. It is valuable to propose an adaptive selection of probability models according to the characteristics of the target problem. In addition, many-tasking optimization is also an interesting field worth further investigation. The sourcecode of EMT-PD is available publicly at https://github.com/CIA-SZU/LWQ.

References

[1] J. H. Wang, Y. Zhou, Y. Wang, J. Zhang, C. L. P. Chen, and Z. B. Zheng, "Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, instances, and algorithms," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 582–594, Mar. 2016.

[2] F. Sarro, F. Ferrucci, M. Harman, A. Manna, and J. Ren, "Adaptive multi-objective evolutionary algorithms for overtime planning in software projects," *IEEE Trans. Softw. Eng.*, vol. 43, no. 10, pp. 898–917, Oct. 2017.

[3] L. Z. Cui *et al.*, "Joint optimization of energy consumption and latency in mobile edge computing for Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4791–4803, Jun. 2019.

[4] 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.

[5] B. D. Li, J. L. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surveys*, vol. 48, no. 1, pp. 1–35, Sep. 2015.

[6] M. Li, S. Yang, and X. Liu, "Diversity comparison of Pareto front approximations in many-objective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2568–2584, Dec. 2014.

[7] H. Chen, R. Cheng, W. Pedrycz, and Y. Jin, "Solving many-objective optimization problems via multistage evolutionary search," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 6, pp. 3552–3564, Jun. 2021.

[8] Y. Tian, C. He, R. Cheng, and X. Zhang, "A multistage evolutionary algorithm for better diversity preservation in multiobjective optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Dec. 20, 2019, doi: 10.1109/TSMC.2019.2956288.

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

[10] W. J. Hong, K. Tang, A. M. Zhou, H. Ishibuchi, and X. Yao, "A scalable indicator-based evolutionary algorithm for large-scale multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 525–537, Jun. 2019.

[11] L. G. de la Fraga and E. Tlelo-Cuautle, "Optimizing an amplifier by a many-objective algorithm based on R2 indicator," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Lisbon, Portugal, May 2015, pp. 265–268.

[12] Y. N. Sun, G. G. Yen, and Z. Yi, "IGD indicator-based evolutionary algorithm for many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 173–187, Apr. 2019.

[13] Q. F. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[14] M. Y. Wu, K. Li, S. Kwong, Y. Zhou, and Q. F. Zhang, "Matching-based selection with incomplete lists for decomposition multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 554–568, Aug. 2017.

[15] R. Cheng, Y. C. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.

[16] X. Y. He, Y. R. Zhou, Z. F. Chen, and Q. F. Zhang, "Evolutionary many-objective optimization based on dynamical decomposition," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 361–375, Jun. 2019.

[17] 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, Apr. 2002.

[18] X. F. Zou, Y. Chen, M. Z. Liu, and L. S. Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 5, pp. 1402–1412, Oct. 2008.

[19] 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, Feb. 2016.

[20] V. Palakonda, S. Ghorbanpour, and R. Mallipeddi, "Pareto dominance-based MOEA with multiple ranking methods for many-objective optimization," in *Proc. IEEE Symp. Series Comput. Intell. (SSCI)*, Bangalore, India, Jul. 2018, pp. 958–964.

[21] 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.

[22] K. C. Tan, L. Feng, and M. Jiang, "Evolutionary transfer optimization—A new frontier in evolutionary computation research," *IEEE Comput. Intell. Mag.*, vol. 16, no. 1, pp. 22–33, Feb. 2021.

[23] 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.

[24] C. E. Yang, J. L. Ding, K. C. Tan, and Y. C. Jin, "Two-stage assortative mating for multi-objective multifactorial evolutionary optimization," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Melbourne, VIC, Australia, Dec. 2017, pp. 76–81.

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

[26] Y. L. Chen, J. H. Zhong, and M. K. Tan, "A fast memetic multi-objective differential evolution for multi-tasking optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8.

[27] N. Q. Tuan, T. D. Hoang, and H. T. T. Binh, "A guided differential evolutionary multi-tasking with powell search method for solving multi-objective continuous optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8.

[28] 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. IEEE Region 10 Annu. Int. Conf. (TENCON)*, Singapore, Nov. 2016, pp. 3157–3164.

[29] R. Sagarna and Y.-S. Ong, "Concurrently searching branches in software tests generation through multitask evolution," in *Proc. IEEE Symp. Series Comput. Intell. (SSCI)*, Athens, Greece, Dec. 2016, pp. 1–8.

[30] R. Chandra, A. Gupta, Y.-S. Ong, and C.-K. Goh, "Evolutionary multi-task learning for modular knowledge representation in neural networks," *Neural Process. Lett.*, vol. 47, no. 3, pp. 993–1009, Jun. 2018.

[31] J. Zhong, L. Feng, W. Cai, and Y.-S. Ong, "Multifactorial genetic programming for symbolic regression problems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 11, pp. 4492–4505, Nov. 2020.

[32] A. Rauniyar, R. Nath, and P. K. Muhuri, "Multi-factorial evolutionary algorithm based novel solution approach for multi-objective pollution-routing problem," *Comput. Ind. Eng.*, vol. 130, no. 5, pp. 757–771, Apr. 2019.

[33] H. Li, Y.-S. Ong, M. G. Gong, and Z. K. Wang, "Evolutionary multi-tasking sparse reconstruction: Framework and case study," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 733–747, Oct. 2019.

[34] Y.-S. Ong and A. Gupta, "Evolutionary multitasking: A computer science view of cognitive multitasking," *Cogn. Comput.*, vol. 8, no. 2, pp. 125–142, Apr. 2016.

[35] M. Q. Li *et al.*, "A benchmark test suite for evolutionary many-objective optimization," *Complex Intell. Syst.*, vol. 3, no. 1, pp. 67–81, Mar. 2017.

[36] A. Gupta, J. Mańdziuk, and Y.-S. Ong, "Evolutionary multitasking in bi-level optimization," *Complex Intell. Syst.*, vol. 1, no. 1, pp. 83–95, Dec. 2015.

[37] K. K. Bali, A. Gupta, L. Feng, Y. S. Ong, and P. S. Tan, "Linearized domain adaptation in evolutionary multitasking," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Donostia, Spain, Jun. 2017, pp. 1295–1302.

[38] B. S. 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. (CEC)*, Vancouver, BC, Canada, Jul. 2016, pp. 1695–1701.

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

[40] M. G. Gong, Z. D. Tang, H. Li, and J. Zhang, "Evolutionary multitasking with dynamic resource allocating strategy," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 858–869, Oct. 2019.

[41] 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.

[42] Z. P. Liang, J. Zhang, L. Feng, and Z. X. Zhu, "A hybrid of genetic transform and hyper-rectangle search strategies for evolutionary multi-tasking," *Expert Syst. Appl.*, vol. 138, no. 30, Dec. 2019, Art. no. 112798.

[43] B. S. Da, A. Gupta, Y. S. Ong, and L. Feng, *The Boon of Gene-Culture Interaction for Effective Evolutionary Multitasking* (Lecture Notes in Computer Science, 9592). Cham, Switzerland: Springer, Feb. 2016, pp. 54–65.

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

[45] X. Xia *et al.*, "Triple archives particle swarm optimization," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4862–4875, Dec. 2020.

[46] L. Feng *et al.*, "An empirical study of multifactorial PSO and multifactorial DE," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Donostia, Spain, Jun. 2017, pp. 921–928.

[47] Z. D. Tang and M. G. Gong, "Adaptive multifactorial particle swarm optimisation," *CAAI Trans. Intell. Technol.*, vol. 4, no. 1, pp. 37–46, Mar. 2019.

[48] H. Song, A. K. Qin, P.-W. Tsai, and J. J. Liang, "Multitasking multi-swarm optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Wellington, New Zealand, Jun. 2019, pp. 1937–1944.

[49] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous space," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[50] X. Xia *et al.*, "A fitness-based adaptive differential evolution algorithm," *Inf. Sci.*, vol. 549, pp. 116–141, Mar. 2021.

[51] D. N. Liu, S. J. Huang, and J. H. Zhong, "Surrogate-assisted multi-tasking memetic algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8.

[52] L. Zhou *et al.*, "Towards effective mutation for knowledge transfer in multifactorial differential evolution," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Wellington, New Zealand, Jun. 2019, pp. 1541–1547.

[53] Z. Liang, H. Dong, C. Liu, W. Liang, and Z. Zhu, "Evolutionary multitasking for multiobjective optimization with subspace alignment and adaptive differential evolution," *IEEE Trans. Cybern.*, early access, Jun. 24, 2020, doi: 10.1109/TCYB.2020.2980888.

[54] Q. Shang *et al.*, "A preliminary study of adaptive task selection in explicit evolutionary many-tasking," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Wellington, New Zealand, Jun. 2019, pp. 2153–2159.

[55] A. Gupta, Y. S. Ong, B. Da, L. Feng, and S. D. Handoko, "Landscape synergy in evolutionary multitasking," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, Jul. 2016, pp. 3076–3083.

[56] J. Zhang, W. E. Zhou, X. Q. Chen, W. Yao, and L. Cao, "Multisource selective transfer framework in multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 424–438, Jul. 2020.

[57] Y. Yuan *et al.*, "Evolutionary multitasking for multiobjective continuous optimization: Benchmark problems, performance metrics and baseline results," Jun. 2017. [Online]. Available: arXiv:1706.02766.

[58] L. Feng, K. Qin, A. Gupta, Y. Yuan, Y. S. Ong, and X. Chi. (2019). *IEEE CEC 2019 Competition on Evolutionary Multi-Task Optimization*. [Online]. Available: http://www.bdsc.site/websites/MTO_competiton_2019/MTO_Competition_CEC_2019.html

[59] X. Y. Zhang, X. T. Zheng, R. Cheng, J. F. Qiu, and Y. C. Jin, "A competitive mechanism based multi-objective particle swarm optimizer with fast convergence," *Inf. Sci.*, vol. 427, pp. 63–76, Feb. 2018.

[60] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[61] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Mar. 2011.

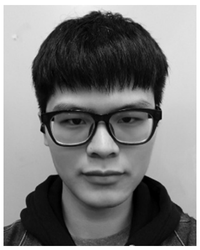
[62] Y. Xiang, Y. R. Zhou, M. Q. Li, and Z. F. Chen, "A vector angle-based evolutionary algorithm for unconstrained many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 131–152, Feb. 2017.

[63] I. Hisao, M. Hiroyuki, and N. Yusuke, "A study on performance evaluation ability of a modified inverted generational distance indicator," in *Proc. 16th Genet. Evol. Comput. Conf. (GECCO)*, Jul. 2015, pp. 695–702.

[64] K. Li, K. Deb, Q. F. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, Oct. 2015.

Zhengping Liang received the B.S. degree in computer science and technology from Hunan Normal University, Changsha, China, in 2001, and the Ph.D. degree in computer science and technology from Wuhan University, Wuhan, China, in 2006.

He is currently an Associate Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His main research interests include computational intelligence, multiobjective optimization, and big data analysis.

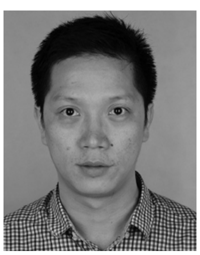


Weiqi Liang received the B.S. degree in software engineering from Zhongkai University, Guangdong, China, in 2017, and the M.S. degree in software engineering from Shenzhen University, Shenzhen, China, in 2020.

His research interests include evolutionary computation, multiobjective optimization, and evolutionary multitasking optimization.

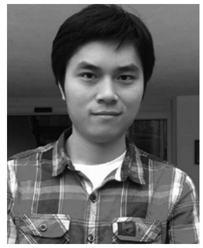
Zhiqiang Wang received the B.S. degree in automation and the M.S. degree in theory electrician from the Wuhan University of Technology, Hubei, China, in 1984 and 1991, respectively.

He is currently a Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include computational intelligence, big data analysis and applications, and multimedia technology and applications.



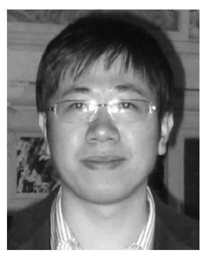
Xiaoliang Ma received the B.S. degree in computer science and technology from Zhejiang Normal University, Jinhua, China, in 2006, and the Ph.D. degree in computer application technology from Xidian University, Xi’an, China, in 2014.

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.



Ling Liu received the B.S. degree in electronic engineering from Nanjing University, Nanjing, China, in 2008, the M.S. degree in electronic engineering from Peking University, Beijing, China, in 2012, and the Ph.D. degree in communication and signal processing from the Electrical and Electronic Engineering Department, Imperial College London, London, U.K., in 2016.

He was a Research Assistant with the CSP Group, Imperial College London. From 2017 to 2019, he served as a Senior Engineer with the CTLab of Huawei Technologies, Shenzhen, China. He is currently an Assistant Professor with the Department of Computer Science and Software Engineering, Shenzhen University, Guangdong, China. His research interests include coding theory, information theory, and machine learning.



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.