# Orthogonal Transfer for Multitask Optimization

Sheng-Hao Wu, *Student Member, IEEE*, Zhi-Hui Zhan, *Senior Member, IEEE*,
Kay Chen Tan, *Fellow, IEEE*, and Jun Zhang, *Fellow, IEEE*

***Abstract*—Knowledge transfer (KT) plays a key role in multitask optimization. However, most of the existing KT methods still face two challenges. First, the tasks may commonly have different dimensionalities (DDs), making the KT between heterogeneous search spaces very difficult. Second, the tasks may have different degrees of similarity in different dimensions, making that treating all dimensions with equal importance may be harmful to the KT process. To address these two challenges, this article proposes a novel orthogonal transfer (OT) method that is enabled by a cross-task mapping (CTM) strategy, which can achieve high-quality KT among heterogeneous tasks. For the first challenge, the CTM strategy maps the global best individual of one task from its original search space to the search space of the target task via an optimization process, which can handle the difference in task dimensionality. For the second challenge, the OT method is performed on the CTM-obtained individual and a random individual of the target task to find the best combination of different dimensions in these two individuals rather than treating all the dimensions equally, so as to achieve high-quality KT. To verify the effectiveness of the proposed OT method and the resulted OT-based multitask optimization (OTMTO) algorithm, this article not only uses the existing multitask optimization benchmark but also proposes a new benchmark test suite named multitask optimization problems (MTOPs) with DDs. Comprehensive experimental results on the existing and the proposed benchmarks show that the proposed OT method and the OTMTO algorithm are very advantageous in providing high-quality KT and in handling the heterogeneity of search space in MTOPs compared to the existing competitive evolutionary multitask optimization (EMTO) algorithms.**

***Index Terms*—Differential evolution (DE), evolutionary computation (EC), evolutionary multitask optimization (EMTO), knowledge transfer (KT), orthogonal experimental design (OED), orthogonal transfer (OT).**

## I. Introduction

EVOLUTIONARY computation (EC), inspired by natural selection and genetics, is a kind of population-based approach for solving optimization problems [1], [2]. The population contains multiple individuals, and each individual represents a candidate solution to the problem. After random initialization, the population undergoes the reproduction process to produce the offspring of the next generation. Then, the fitness is evaluated to measure the solution quality of the individual to help maintain the elite individuals. This process is called fitness evaluation (FE). The evolutionary process executes repeatedly until a stopping criterion is met. EC algorithms have been successfully applied to complex optimization problems, such as large-scale [3]–[5], dynamic [6], [7], multimodal [8], [9], multi-/many-objective [10]–[12], expensive [13]–[15], and real-world applications [16]–[20].

Currently, an emerging trend to use EC algorithms to solve multiple optimization tasks simultaneously, namely, evolutionary multitask optimization (EMTO), has developed rapidly in the EC community [21]. The corresponding problem related to EMTO is called multitask optimization problem (MTOP). The basic idea of EMTO is that we seldom handle optimization problems independently from scratch. In contrast, we can make use of the problem similarity between the tasks to facilitate a more efficient search. Herein, a task corresponds to an optimization problem. In the EMTO search paradigm, we handle multiple tasks simultaneously and assume that there is some similarity among these tasks. Therefore, during the search process, some information from other similar tasks can be transferred and reused to improve the solving performance of the current task. This information-sharing mechanism between similar tasks is denoted as knowledge transfer (KT). Currently, many EMTO algorithms have been proposed based on various KT methods and have shown great success in improving the search ability on similar optimization tasks [22]–[24].

Despite the success brought by EMTO algorithms, the design of effective KT still faces two main challenges, including the difference in the task dimensionality and the unequal importance of different dimensions.

First, for the challenge of the difference in the task dimensionality, the MTOP may contain optimization tasks with

Manuscript received 3 August 2021; revised 2 November 2021 and 18 January 2022; accepted 3 March 2022. Date of publication 17 March 2022; date of current version 31 January 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102102; in part by the National Natural Science Foundations of China under Grant 62176094 and Grant 61873097; in part by the Key-Area Research and Development of Guangdong Province under Grant 2020B010166002; in part by the Guangdong Natural Science Foundation Research Team under Grant 2018B030312003; and in part by the National Research Foundation of Korea under Grant NRF-2021H1D3A2A01082705. *(Corresponding authors: Zhi-Hui Zhan; Jun Zhang.)*

Sheng-Hao Wu and Zhi-Hui Zhan are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, also with Pazhou Laboratory, Guangzhou 510330, China, and also with the Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, Guangzhou 510006, China (e-mail: zhanapollo@163.com).

Kay Chen Tan is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: kctan@polyu.edu.hk).

Jun Zhang is with Hanyang University, Ansan 15588, South Korea (e-mail: junzhanghk@gmail.com).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TEVC.2022.3160196.

Digital Object Identifier 10.1109/TEVC.2022.3160196

different dimensionalities [21]. Herein, the MTOP with different dimensionalities (DDs) is denoted as MTOP-DD. However, many of the existing KT methods are based on the unified solution representation (USR) strategy [22]–[24]. The USR strategy adds some redundant dimensions to the individuals in low-dimensional tasks to make different tasks with the same dimensionality. Then, the KT is implemented in the form of crossover between these redundant representations of individuals in different tasks. Therefore, these methods are not well suited to solve the MTOP-DD since the difference in the dimensionality between the tasks can lead to negative KT. For example, when performing KT from a low-dimensional task to a high-dimensional task, some redundant dimensions will be added to the individual in the low-dimensional task due to the USR strategy and will be transferred to the high-dimensional task. However, these redundant dimensions have no effect on the fitness calculation in the low-dimensional task, and their quality is unknown. Therefore, transferring these redundant dimensions to the high-dimensional task may cause low-quality or even negative KT.

Second, for the challenge of unequal importance of different dimensions, it is known that some dimensions of other tasks contain important information that can be beneficial to the evolutionary search of the current task, while some dimensions are less important or even contain misleading information. A simple example is that the two tasks have some similar dimensions in their global optima, while other dimensions differ greatly. In this case, it is not reasonable to treat all transferred dimensions with the same importance.

Besides the above two main challenges, another challenge in KT is how to make the best use of knowledge between different dimensions of different tasks because transferring information across the similar dimensions of different tasks may be useful. Some KT methods in the literature try to shuffle the dimensions of the tasks to transfer knowledge across different dimensions [25], [26]. However, if the KT based on dimension shuffle does not consider the similarity between different dimensions, it may cause negative transfer by transferring information between the dimensions that have a large gap in the search distribution.

To address the above challenges, we propose a novel KT method named orthogonal transfer (OT) to more efficiently solve the MTOP, especially for solving the MTOP-DD. Thus, an OT-based multitask optimization (OTMTO) algorithm is proposed, which has the following four main features.

First, we propose a cross-task mapping (CTM) strategy to enable the KT between heterogeneous tasks with DDs. Specifically, by modeling the mapping process as an optimization problem and solving it by a differential evolution (DE) algorithm, the global best individual of one task can be well mapped to the search space of the target task and the cross-task knowledge can be efficiently transferred to the target task.

Second, based on the CTM-obtained transferred individual, the OT method is performed on this individual and a random individual of the target task. Since these two individuals are under the same search space (i.e., the search space of the target task), the negative KT caused by the DDs can be reduced. Moreover, the OT method adopts the orthogonal experimental design (OED) method to find the best combination of dimensions between these two individuals (i.e., the CTM-obtained transferred individual and the randomly selected individual of the target task). The OED method implicitly learns the importance of different dimensions. In this way, some useful information of other tasks is transferred to the target task while the good information of the target task can be preserved.

Third, to make the best use of different dimensions between different tasks and to avoid negative transfer, a similarity-based cross-dimension transfer (CDT) method is proposed. The CDT method is based on the similarity of the search distribution on the dimensions between the tasks to transfer the knowledge. In the CDT method, the information of some dimensions in the source task is transferred to their most similar dimensions in the target task, which can further improve the quality of KT.

Fourth, the occurrences of the OT and CDT processes are determined by two probability parameters, respectively, which are both adaptively adjusted according to the benefit brought by the OT process and the CDT process, respectively. This mechanism can adaptively control the intensity of KT brought by OT and CDT, so as to encourage positive KT.

Therefore, the OT-based OTMTO algorithm is proposed enabled by the CTM strategy and together with the CDT method. The CTM and OT can well tackle with the two main challenges of the current KT methods, i.e., the challenge of difference in the task dimensionality and the challenge of unequal importance of different dimensions, respectively, while the CDT can well tackle with the challenge of making the best use of knowledge between different dimensions of different tasks. In the experiments, unlike the existing studies that are mainly conducted on same dimensional MTOPs, we verify the advantages of OTMTO not only on the traditional MTOP benchmark but also on a newly proposed MTOP-DD benchmark test suite, by comparing it with the baseline and state-of-the-art algorithms. The experimental results show that the advantage of the OTMTO algorithm is significant both in solving traditional MTOP and the proposed MTOP-DD.

The remainder of this article is organized as follows. Section II briefly introduces the concept and development of the MTOP, the motivation of this article, and the OED method. Then, Section III presents the details of the proposed OTMTO algorithm. Section IV provides the experimental studies. Finally, Section V gives the concluding remarks of this article.

## II. Preliminary

### A. Evolutionary Multitask Optimization

Herein, we give the formulation of the MTOP. Without loss of generality, we assume that there are $K$ tasks that are all single-objective minimization problems. We suppose that the $i$th task has an objective function $f_i : X_i \rightarrow \mathbb{R}$ and the solution space is $X_i \subseteq \mathbb{R}^{Di}$, where $D_i$ is the dimensionality of the $i$th task. Then, an MTOP that contains $K$ tasks can be formulated as

$$\left\{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_K^*\right\} = \{\operatorname{argmin} f_1(\boldsymbol{x_1}), \ldots, \operatorname{argmin} f_K(\boldsymbol{x_K})\}. \quad (1)$$

The traditional EC algorithm solves the MTOP in a single-task manner. That is, the traditional EC algorithm solves one task in a single run. However, in real-world applications, problems seldom exist in isolation, and there is some synergy among the problems of a specific class. Therefore, the key idea of EMTO is that we can reuse the knowledge learned from solving a problem to improve the performance of solving other similar problems. As a result, how to perform KT between tasks is essential to design an effective EMTO algorithm.

### *B. Related Work*

The proposal of EMTO is inspired by transfer learning that achieves better learning performance by sharing common knowledge between similar tasks. Different from transfer learning, the EMTO paradigm emphasizes the optimization process using EC algorithms. However, similar considerations can still be drawn from transfer learning [27] to improve multitask optimization performance. That is, researchers consider the issues of "what and how to transfer" and "when to transfer."

For the first issue of what and how to transfer, researchers are interested in designing effective KT methods to transfer common knowledge among tasks. These existing works can be mainly categorized as single population-based methods and multipopulation-based methods.

For the single population-based methods, researchers mainly consider evolving a single population that contains individuals of different tasks and performing KT between individuals. Multifactorial optimization is one of the most popular approaches aiming to solve MTOP with a single population [22]. The corresponding algorithm is called a multifactorial evolutionary algorithm (MFEA). The MFEA proposed assortative mating and vertical cultural transmission to enable KT among tasks and showed promising performance on continuous and discrete MTOPs. Following MFEA, many studies have been carried out to improve KT in EMTO. Bali *et al.* [23] proposed using linearized domain adaption to improve KT in MFEA by minimizing the distribution gap between two heterogeneous tasks. Zhou *et al.* [24] proposed an adaptive KT method that used multiple crossover operators to transfer knowledge and distinguished the suitable KT crossover operator toward the task based on the information collected during the evolutionary process.

For the multipopulation-based methods, researchers mainly consider evolving multiple populations for multiple tasks, respectively, and performing KT between populations based on the evolutionary information such as the search distribution. Wu and Tan [25] proposed a multitasking genetic algorithm (MTGA) to minimize the bias of the optima in different tasks and to enable KT between tasks with DDs by performing random shuffle on the dimensions. MTGA is a simple yet efficient EMTO algorithm. However, the random shuffle on the dimension does not consider the similarity between dimensions. Feng *et al.* [28] proposed a multipopulation-based EMTO algorithm with an explicit autoencoder, which allowed KT between tasks whose dimensionalities were different. Zhou *et al.* [29] proposed using a kernelized autoencoder to perform a nonlinear mapping between populations to achieve effective KT between tasks. Li *et al.* [30] proposed a multipopulation framework that included a new mutation operator to allow KT between multiple populations. Very recently, Li *et al.* [31] found that transferring task-specific knowledge like the high-quality solutions between tasks was not sufficient and proposed a novel KT method that could transfer meta-knowledge between populations of the tasks. The meta-knowledge is the "knowledge (e.g., how to do to obtain the) of knowledge (e.g., high-quality solution)," which is more general to be transferred between various kinds of tasks (e.g., high, medium, or low similar).

For the second issue of when to transfer, researchers are interested in studying in what situations the knowledge should or should not be transferred. Since the similarity between tasks is usually not known in advance, performing KT between tasks may not be always useful. Therefore, an important issue is to estimate the similarity between tasks and encourage KT when the tasks are similar. To this aim, Zheng *et al.* [32] proposed a self-regulated EMTO algorithm. In their proposed strategy, when the tasks show some similarity in the search process, the intensity of the cross-task KT will be enhanced. Bali *et al.* [33] proposed MFEA2 with an online parameter estimation strategy that can estimate the similarity between tasks, so that can encourage positive KT in suitable time when the tasks were considered to be similar.

To summarize, a good design of an EMTO algorithm usually considers the issues of "what and how to transfer" and "when to transfer" simultaneously. In addition, there are also other studies that focus on different types of problems by EMTO algorithms, such as many-task optimization problems [34]–[37], combinatorial problems [38]–[40], multiobjective problems [41]–[43], computationally expensive problems [44]–[46], theoretical analysis of the effectiveness of the EMTO algorithm [47]–[49], and applications of EMTO algorithms to real-world problems [50]–[53].

### *C. Motivation*

In this article, we also consider both the issues of "what and how to transfer" and "when to transfer" for the design of powerful KT. Many of the existing methods perform KT on the USR of individuals from different tasks. That is, all the individuals from all tasks are encoded to unified search space, denoted as $U$, and the dimensionality of $U$ is $D_U = \max\{D_i\}$, $i = 1, \ldots, K$. Note that all dimensions of the individuals from the $i$th task are normalized to the value in [0, 1] according to the lower and upper bound of the solution space of the $i$th task, denoted as $\text{xmin}_i$ and $\text{xmax}_i$, respectively. When the individual from the $i$th task is evaluated, it should be decoded into the solution space of the $i$th task, which is $X_i \subseteq \mathbb{R}^{Di}$. Suppose that $\boldsymbol{u}_i$ is an individual from the $i$th task in $U$, then the first $D_i$ dimensions of $\boldsymbol{u}_i$ are decoded as the candidate solution $\boldsymbol{x}_i$ for the $i$th task. That is, $\boldsymbol{x}_i = x\text{min}_i + \boldsymbol{u}_i(1 : D_i) \times (x\text{max}_i - x\text{min}_i)$. However, this KT method may lead to negative KT, which cannot fully realize the benefits of the synergy between similar tasks. For example, when performing KT from the $i$th task to the $j$th task in $U$, where $D_i < D_j$, redundant dimensions of the individual of the $i$th task (i.e., the last $(D_j - D_i)$ dimensions) could be transferred to the individual of the $j$th task. These redundant dimensions have no effect on calculating the fitness of the $i$th task, which can be poorly optimized. Therefore, transferring these redundant dimensions can lead to negative KT. To handle the MTOP-DD, a KT method that works on the original search spaces of the source and target tasks rather than the unified search space is needed. To enable such a KT method, the CTM strategy is proposed.

Moreover, it is known that different dimensions of two tasks often have different levels of importance and similarity. Most of the existing crossover-based KT methods use universal crossover probability to transfer knowledge. Therefore, some low-quality information (e.g., dimensions with a large gap in the global optima of the two tasks) of the source task may also be transferred to the target task. This kind of trivial crossover can severely break the building blocks and lead to the deterioration of the performance of EMTO algorithms. To handle this issue, the OT and CDT methods that are aware of the importance and similarity of dimensions are proposed.

Furthermore, the issue of "when to transfer" should also be considered to improve the KT in EMTO algorithms. Motivated by this, a simple and efficient adaptive control mechanism of the occurrence of the KT is proposed in this article.

### *D. Orthogonal Experimental Design*

OED is a method for the experimental design of multiple factors with multiple levels. The objective of the OED is to find the best combination of the levels associated with different factors, and the design corresponding to this combination can lead to the best performance according to the scientific or engineering requirement. The OED has enjoyed great success in many engineering applications [54]. The key idea of OED is to select representative combinations based on orthogonality and run experiments on these combinations to determine the possible best combination. Compared to exhaustive methods, which run all the possible combinations in exponential order, the OED requires fewer experiments to predict a high-quality combination. Hence, the OED is an economic and efficient experimental design method. Basically, the OED method contains two main components: 1) *orthogonal array* (OA) and 2) *factor analysis*. Suppose that there are $D$ factors and that each factor has two available levels in the experiment. Then, there are a total of $2^D$ possible experimental settings. To use the OED method, we obtain the OA by the process proposed in [55]. The OA is a predefined $M \times D$ table, denoted by $L_M(2^D)$, where $M = 2^{\lceil \log_2(D+1) \rceil}$. Each row in the OA represents a combination of the level of factors, and each element in the table is denoted as $\mathrm{OA}_{i,j}$, whose value represents the selected level of factor $j$ for the $i$th combination. In this two-level example, each element $\mathrm{OA}_{i,j}$ in the table can have a value of one or two, and each column of the OA contains an equal number of one or two. Notably, $M$ is much smaller than $2^D$ when $D$ is sufficiently large. For details of the construction process of the OA, refer to Section A in the supplementary material. Then, we conduct experiments on $M$ combinations according to the OA and obtain the experimental results of these combinations. After $M$ experiments of combinations are conducted, we perform the factor analysis to analyze the level effects of each factor and build the best combination from the experimental results. Let $f_i$ denote the experimental result of the $i$th combination and $S_{jk}$ denote the effect of the $k$th level of the $j$th factor; then, $S_{jk}$ is calculated as

$$S_{jk} = \frac{\sum_{i=1}^{M} f_i \times I\big(\mathrm{OA}_{i,j} == k\big)}{\sum_{i=1}^{M} I\big(\mathrm{OA}_{i,j} == k\big)} \tag{2}$$

where $I(\cdot)$ is a function that returns 1 if the condition in parentheses is satisfied and 0 otherwise. After all the $S_{jk}$ values are calculated, we can derive the predictive best combination by selecting the level $k$ for each factor $j$ that results in the best $S_{jk}$. Taking the minimization optimization problem as an example, the $j$th factor of the derived combination is the level $k$ having the smallest $S_{jk}$.

OED has been widely used in the traditional EC area (e.g., single-task optimization) to improve search performance. Leung and Wang [56] proposed using OED to help the population initialization and to improve the crossover operator to enhance the robustness of the EC algorithm. Ho *et al.* [57] proposed using OED to find better combinations of the partial vectors between the two individuals generated by cognitive learning and social learning, which could deal with high-dimensional optimization problems. Zhan *et al.* [55] proposed an orthogonal learning particle swarm optimization (OLPSO) algorithm, which used OED to construct a better guiding exemplar from the global best individual and the personal best individual to solve the complex single-objective optimization in a single-task manner and achieved encouraging results. Later, the orthogonal learning strategy has become a promising approach for enhancing the global search ability of various EC algorithms [58], [59]. However, most of the researches adopt the OED to improve the search performance for single-task optimization. To the best of our knowledge, no research attention has been given to using OED to perform cross-task KT in multitask optimization. Inspired by the success of the OED in improving single-task optimization performance, we propose a novel OT method to transfer knowledge between tasks to benefit the search process.

## III. OTMTO

In this section, the multipopulation framework for solving MTOP-DD is first introduced. Afterward, the CTM strategy operating on the original search space is introduced, followed by the details of the OT method. Then, the CDT method that further exploits the similarity between the global optima of the tasks is presented. Then, the complete OTMTO algorithm is given, followed by the time complexity analysis.

### *A. Multipopulation Framework*

Since the proposal of the MFEA that tries to utilize the implicit parallelism of the EC algorithms to solve the MTOP with a single population, many studies have followed up on this work, and many improved versions of MFEA have been proposed. Despite the rationality of MFEA and its variants, this single population-based search paradigm still faces three limitations. First, the scalability from multitask to many-task optimization is restricted, as discussed in [34]. When the number of tasks becomes larger (e.g., more than three), the MFEA becomes less effective. This is due to that the MFEA rarely considers the similarity between the source and target tasks, and some source tasks that differ greatly from the target task can lead to negative KT on the target task. Second, the MFEA performs crossover-based KT on the USR of the individuals between tasks. As mentioned in Section II-A, this can lead to negative transfer since the crossover-based KT may transfer irrelevant dimensions of the source task. Third, the MFEA adopts the implicit KT. That is, the MFEA uses the same evolutionary search operators for multiple tasks within a single population. However, different tasks might require different search operators, and the implicit KT restricts the flexibility of the EMTO algorithms.

Therefore, we adopt the multipopulation framework instead of the single-population framework to address the MTOP-DD in this article. Specifically, we maintain a population for each task. During the evolutionary process, each population evolves independently on each task. One of the most salient features of the multipopulation framework is the allowance of the use of multiple evolutionary operators [60], [61]. That is, we can use different EC algorithms or optimization tools to solve different tasks in different populations while performing KT between/among them. Since different tasks might require different search mechanisms and different intensities of KT from other tasks, the multipopulation framework for solving MTOP-DD is more flexible and general in use. In this article, we implement the multipopulation framework sequentially. That is, in every generation, after the population of one task finishes the evolutionary process, the KT process is carried out. Note that another advantage of the multipopulation framework is that it allows for parallel computing implementation.

Although some multipopulation-based EMTO algorithms have been studied in the literature, our proposed OTMTO algorithm is different from these algorithms in two aspects. First, the OTMTO algorithm is flexible because it not only can handle MTOP with the same dimensionality but also can handle MTOP-DD well. Second, the OTMTO algorithm achieves more effective KT between tasks by both considering the importance and similarity of different dimensions.

### *B. CTM Strategy*

In this section, we consider how to map the global best individual of the $j$th task (i.e., the source task), denoted as $\boldsymbol{x}_{gb,j}$, from its search space $[0, 1]^{D_j}$ to obtain the mapped global best individual named $\boldsymbol{x}_{mgb,j}$ that is suitable for the $i$th task (i.e., the target task) in the search space $[0, 1]^{D_i}$. The concept of the KT in optimization originates from the KT in the transfer learning area [7], [27]. In transfer learning, the goal is to improve the learning performance of one task by transferring knowledge from the other task to alleviate the burden of expensive data labeling. The KT in the transfer learning area has enjoyed great success. One of the key issues in transfer learning is domain knowledge alignment, which reduces the bias of the data distribution in the common knowledge space (e.g., latent feature space) and matches data from different domains correctly. Following the idea of domain knowledge alignment, we propose to formulate the CTM process as an optimization problem that aims to minimize the gap between the hypotheses of the target task $i$ and the source task $j$, denoted as $H_t$ and $H_s$. A hypothesis $H$ (i.e., $H : X \rightarrow Y$) is a mapping from search space $X$ to objective space $Y$. In the following, the optimization model in the CTM strategy is introduced in Section III-B1). Then, the design principle of hypothesis alignment and the connection of the optimization model to machine learning are explained and revealed in Section III-B2).

*1) Optimization Model:* We denote the population for the $i$th task (i.e., target task) as $\boldsymbol{pop_i}$ and the $k$th individual in $\boldsymbol{pop_i}$ as $\boldsymbol{pop_{k,i}}$, and the sizes of all populations are the same which are denoted as $ps$. Similarly, the $k$th individual for the $j$th task is $\boldsymbol{pop_{k,j}}$. First, the individuals of two populations are, respectively, sorted from best to worst. We denote the *n*umber of *b*est individuals of the two populations selected to construct the mapping as $nb$. Then, an $nb$-dimensional temporary feature vector $\boldsymbol{tvec_j}$ of the $j$th task is defined, and the $k$th $(1 \le k \le nb)$ dimension of $\boldsymbol{tvec_j}$ is the distance between $\mathbf{x}_{gb,j}$ and the $k$th best individual in $\boldsymbol{pop_j}$ (i.e., $\boldsymbol{pop_{k,j}}$) as

$$tvec_{k,j} = \left\|\boldsymbol{x}_{gb,j} - \boldsymbol{pop}_{k,j}\right\|_2. \tag{3}$$

Similarly, the $nb$-dimensional feature vector for the $i$th task, denoted as $\boldsymbol{fvec}_i$, is also defined, and the $k$th $(1\le k\le nb)$ dimension of $\boldsymbol{fvec_i}$ is the distance between $\mathbf{x}_{map}$ and the $k$th best individual in $\boldsymbol{pop_i}$ (i.e., $\boldsymbol{pop_k}, \boldsymbol{i}$) as

$$\boldsymbol{fvec}_{k,i} = \left\|\boldsymbol{x}_{map} - \boldsymbol{pop}_{k,i}\right\|_2 \tag{4}$$

where $\boldsymbol{x}_{map}$ is a candidate solution vector to be optimized to approach $\mathbf{x}_{mgb,j}$. Then, we perform a scale transformation between two populations to obtain the feature vector for the $j$th task, denoted as $\boldsymbol{fvec_j}$ as

$$\boldsymbol{fvec}_j = \boldsymbol{tvec}_j \times rad_i/rad_j \tag{5}$$

where $\mathrm{rad}_i$ and $\mathrm{rad}_j$ are the search radii of $\boldsymbol{pop_i}$ and $\boldsymbol{pop_j}$, respectively. Specifically, $\mathrm{rad}_i$ is calculated as

$$rad_i = \frac{1}{ps}\sum_{k=1}^{ps}\left\|\boldsymbol{pop_{k,i}} - \boldsymbol{ctr}_i\right\|_2 \tag{6}$$

where $\boldsymbol{ctr}_i$ denotes the arithmetic mean center of the $\boldsymbol{pop_i}$. The calculation of $rad_j$ is similar to that of $rad_i$. Then, the optimization model is formulated as

$$\boldsymbol{x_{mgb,j}} = \underset{\boldsymbol{x}_{map}\in[0,1]^{D_i}}{\mathrm{argmin}} \left\|\boldsymbol{fvec_j} - \boldsymbol{fvec_i}\right\|_2. \tag{7}$$

Once $\boldsymbol{fvec_j}$ is calculated, $\boldsymbol{fvec_j}$ is fixed through the optimization process. In contrast, $\boldsymbol{fvec_i}$ is calculated according to the candidate solution $\boldsymbol{x}_{\mathbf{map}}$. Note that (5) approximates the scale of the distance metric of the search space $[0, 1]^{D_j}$ to the search space $[0, 1]^{D_i}$. In this optimization problem, we directly find a mapped individual $\boldsymbol{x_{mgb,j}}$ in the original search space of the $i$th task by considering the information of the $j$th task. Hence, the CTM process can be viewed as extracting the knowledge from the source task. If $nb$ equals 1, the best-mapped individual $\boldsymbol{x_{mgb,j}}$ for this optimization problem

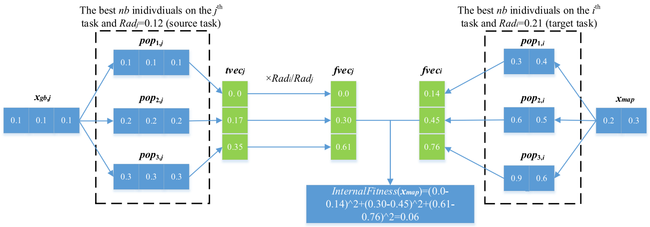

The best nb inidivdiuals on the j th task and Radj=0.12 (source task)
The best nb inidivdiuals on the i th task and Radi=0.21 (target task)
xgb,j
0.1 0.1 0.1
pop1,j
0.1 0.1 0.1
pop2,j
0.2 0.2 0.2
pop3,j
0.3 0.3 0.3
tvecj
0. 0
0.17
0.35
×Radi/Radj
fvecj
0.0
0.30
0.61
fveci
0.14
0.45
0.76
pop1,i
0.3 0.4
pop2,i
0.6 0.5
pop3,i
0.9 0.6
xmap
0.2 0.3
InternalFitness(xmap)=(0.0-0.14)^2+(0.30-0.45)^2+(0.61-0.76)^2=0.06

Fig. 1. Example of the internal fitness calculation for solution $\boldsymbol{x}_{\mathbf{map}}$ ($nb = 3$).

is $\boldsymbol{x}_{\boldsymbol{gb,i}}$, which leads to a zero value of $\|\boldsymbol{fvec_j} - \boldsymbol{fvec_i}\|_2$. To bring in diversity and provide $\boldsymbol{x}_{\boldsymbol{mgb,j}}$ with better dimensions for the OT process, we set $nb = 5$. The fitness of a candidate solution $\boldsymbol{x}_{\mathbf{map}}$ to be minimized is named internal fitness and is calculated as

$$InternalFitness(\boldsymbol{x}_{\mathbf{map}}) = \sum_{k=1}^{nb} \left(\boldsymbol{fvec_{k,j}} - \boldsymbol{fvec_{k,i}}\right)^2. \tag{8}$$

An example of the internal fitness calculation process, where the $j$th task is a 3-dimensional (3-D) optimization problem and the $i$th task is a 2-dimensional (2-D) optimization problem, is shown in Fig. 1. In this example ($nb = 3$), to calculate the fitness of a candidate mapped individual $\boldsymbol{x}_{\mathbf{map}}$ on the right-hand side of Fig. 1, the best $nb$ individuals of the $i$th task (i.e., target task) and the $j$th task (i.e., source task) are first selected. Then, $\boldsymbol{fvec_j}$ is calculated according to (3) and (5) while $\boldsymbol{fvec_i}$ is calculated according to (4). Finally, $InternalFitness(\boldsymbol{x}_{\mathbf{map}})$ is calculated according to (8).

Since the optimization problem can be complex when $nb>1$, we use DE/best/1, which is a DE variant with fast convergence ability. The reason for the using DE/best/1 algorithm is that we aim to locate the optimal or near-optimal solution as fast as possible. Suppose that the population size for the internal optimization process is denoted as *inps*. Then, we use the best *inps* individuals from the $\boldsymbol{pop_i}$ as the initial internal population, denoted as *inpop*, instead of uniform random initialization in the whole search space. This is because the individuals in $\boldsymbol{pop_i}$ tend to have better internal fitness compared to the uniformly sampled ones in the search space. As a result, the search efficiency is enhanced. The termination condition is when the number of internal FE, denoted as *INFE*, is larger than the predefined maximum number of FE, denoted as *maxINFE*. After the termination condition is met, we output the individual with the best internal fitness as $\boldsymbol{x}_{\boldsymbol{mgb,j}}$. The details of the CTM process are shown in Algorithm 1.

*2) Explanation for the CTM Mechanism:* Recall that the distance-weighted $k$-nearest neighbor (kNN) [62] is a canonical algorithm for the classification problem. The learned classifier $H$ on the data is a hypothesis. The predicted label $\tilde{y}$ for a solution vector $\boldsymbol{x}$ is

$$\tilde{y} = H(\boldsymbol{x}) = \sum_{\boldsymbol{x_i} \in N(\boldsymbol{x})} w_i y_i \tag{9}$$

$$w_i = \frac{\operatorname{dist}(\boldsymbol{x_i}, \boldsymbol{x})}{\sum_{\boldsymbol{x_i} \in N(\boldsymbol{x})} \operatorname{dist}(\boldsymbol{x_i}, \boldsymbol{x})} \tag{10}$$

**Algorithm 1:** CTM Strategy

```
Input: Task $i$ (i.e., target task), Task $j$ (i.e., source task),
       $nb, \boldsymbol{pop_i}, \boldsymbol{pop_j}, \boldsymbol{x_{gb,j}}$
Output: $\boldsymbol{x_{mgb,j}}$
Parameter: $INFE = 0$, $maxINFE = 500$, $inps = 50$
Note: $\boldsymbol{x_{mgb,j}} \in [0, 1]^{Di}$ and $\boldsymbol{x_{gb,j}} \in [0, 1]^{Dj}$
Begin
    Initialize the population $inpop$ as the $inps$ best individuals in $\boldsymbol{pop_i}$;
    While $INFE < maxINFE$
        Evolve the population for one generation by using DE/best/1 with the internal fitness function of Eq. (8);
    End While
    Find the best individual $\boldsymbol{x_{map}}$ with the best internal fitness in the population $inpop$ and output it as $\boldsymbol{x_{mgb,j}}$;
End
```

where dist(·, ·) is the distance metric of two vectors, $N(\boldsymbol{x})$ is a set containing $k$ nearest neighbors of $\boldsymbol{x}$ according to a distance metric, and $y_i$ ($i \in \{1, \ldots, k\}$) is the label of $\boldsymbol{x_i}$. Note that $N(\boldsymbol{x}) \subset \boldsymbol{X}$, where $\boldsymbol{X}$ is the whole training dataset.

In the proposed optimization model, the predicted label $\tilde{y}$ is the predicted rank of the solution $\boldsymbol{x}$ in the population of a task. Note that there are two learned hypotheses $H_s$ and $H_t$, and two training datasets $\boldsymbol{X_s}$ and $\boldsymbol{X_t}$ for the source and the target tasks, respectively. The two hypotheses aim to correctly predict the rank of solution $\boldsymbol{x}$ on their corresponding tasks based on the training data $\boldsymbol{X_s}$ and $\boldsymbol{X_t}$ from the two populations. Then, minimizing the difference in $\boldsymbol{fvec_j}$ and $\boldsymbol{fvec_i}$ in (8), which belong to the source task $j$ and the target task $i$, respectively, can reduce the difference in weights of the rank prediction processes [i.e., $H_s(\boldsymbol{x_{gb,j}})$ and $H_t(\boldsymbol{x_{mgb,j}})$] of the two solutions in (9). In this way, our proposed optimization model in CTM can be regarded as minimizing the gap between the learned $H_s(x)$ and $H_t(x)$. The aligned hypotheses are special cases of kNN when $nb = k$ and $\boldsymbol{X_s}$ contains the best $nb$ solutions in the populations of the source task while $\boldsymbol{X_t}$ contains the best $nb$ solutions in the populations of the target task. Thus, by implicitly aligning the hypothesis of the two tasks, the cross-task knowledge from the source task can be transferred to help the search process of the target task.

### C. OT Method

The occurrence of the CTM and OT process from the $j$th task to the $i$th task is controlled by a probability parameter, denoted as $p_{OT,i,j}$. Specifically, if the uniformly generated random number within [0,1] is less than $p_{OT,i,j}$, the CTM is carried out and the OT process is performed. Moreover, the parameter $p_{OT,i,j}$ is adaptively adjusted according to the benefit brought by the OT process. To achieve KT between tasks, we perform the OT method based on the original search space rather than the unified search space. Moreover, to address the difference in the search range of the decision variables (i.e., dimensions) and enable KT, we normalize the dimensions of the candidate solutions to [0,1] according to the lower and upper bounds of the original search space. During evolution, all the individuals of the $i$th task and the $j$th task are encoded into the search space $[0, 1]^{D_i}$and $[0, 1]^{D_j}$, respectively. Suppose that we are going to transfer knowledge from the $j$th task to the $i$th task, where $i \neq j$ and $1 \leq i, j \leq K$. The global best individual of the $j$th task is selected as transferred knowledge and denoted as $\boldsymbol{x_{gb,j}}$, which belongs to the normalized search spaces $[0, 1]^{D_j}$. We first execute the CTM process to map $\boldsymbol{x_{gb,j}}$ of the $j$th task from the search space $[0, 1]^{D_j}$ to obtain the mapped individual $\boldsymbol{x_{mgb,j}}$ that belongs to the search space $[0, 1]^{D_i}$ of the target task. Note that $D_i$ is not necessarily equal to $D_j$. Afterward, OED is performed on $\boldsymbol{x_{mgb,j}}$ and a randomly selected individual from $\boldsymbol{pop_i}$, denoted as $\boldsymbol{x_{rk,i}}$, where $rk \in \{1, \ldots, ps\}$. Since $\boldsymbol{x_{mgb,j}}$ is constructed according to the information from the $j$th task, it can be viewed as cross-task knowledge.

When performing OED on the two individuals $\boldsymbol{x_{rk,i}}$ and $\boldsymbol{x_{mgb,j}}$, a better OT solution, denoted as $\boldsymbol{x_{OT}}$, is generated to guide the evolution of the target task $i$. Specifically, the OED method aim to obtain a high-quality solution between two $D_i$-dimensional individuals by discovering the best combination of their dimensions. Therefore, the OED method works on a two-level $D_i$-factor experimental design problem. First, we build an OA $L_M(2^{D_i})$ by the process introduced in [55]. Second, $M$ individuals are constructed by selecting the corresponding value from $\boldsymbol{x_{rk,i}}$ or $\boldsymbol{x_{mgb,j}}$ according to the OA. That is, for the experimental individual $\boldsymbol{x_m}$ $(1 \leq m \leq M)$, the $k$th dimension of $\boldsymbol{x_m}$ will be the $k$th dimension of $\boldsymbol{x_{gb,i}}$ if $\mathrm{OA}_{m,k} = 1$ or will be the $k$th dimension of $\boldsymbol{x_{mgb,j}}$ if $\mathrm{OA}_{m,k} = 2$. Third, $M$ experimental individuals are evaluated on the $i$th task, and the best one is denoted as $\boldsymbol{x_b}$. Then, we perform the factor analysis process and derive a predictive best individual $\boldsymbol{x_p}$. After the evaluation of $\boldsymbol{x_p}$, the better individual of $\boldsymbol{x_b}$ and $\boldsymbol{x_p}$ is set as $\boldsymbol{x_{OT}}$. Finally, the $\boldsymbol{x_{OT}}$ will be compared with the $\boldsymbol{x_{rk,i}}$. If $\boldsymbol{x_{OT}}$ is better than $\boldsymbol{x_{rk,i}}$, $\boldsymbol{x_{OT}}$ will replace $\boldsymbol{x_{rk,i}}$ in $\boldsymbol{pop_i}$. The reward of this OT process, denoted as $r_{OT}$, is set to 1. Otherwise, $r_{OT}$ is set to 0. Inspired by the reinforcement learning technique that uses the accumulated discounted reward to learn for an optimal behavior policy [63], we design a parameter update mechanism for $p_{OT,i,j}$ as

$$p_{OT,i,j} = 0.95 \times p_{OT,i,j} + 0.05 \times r_{OT}. \tag{11}$$

In this way, we can adaptively control the intensity of KT. When the OT method brings in a better transferred individual $\boldsymbol{x_{OT}}$, the $p_{OT,i,j}$ will be increased, and the OT process will be encouraged. In contrast, the $p_{OT,i,j}$ will be reduced if the OT process cannot offer any benefits. The complete process of the OT method is shown in Algorithm 2.

**Algorithm 2:** OT Method

```
Input: Task $i$ (i.e., target task), Task $j$ (i.e., source task),
       $D_i$, $\boldsymbol{pop_i}$, $\boldsymbol{x_{mgb,j}}$
Output: $r_{OT}$, $\boldsymbol{pop_i}$
Note: $\boldsymbol{x_{rk,i}}$, $\boldsymbol{x_{mgb,j}} \in [0, 1]^{Di}$
Begin
    Generate OA as $\mathrm{L}_M(2^{Di})$;
    Randomly select one individual $\boldsymbol{x_{rk,i}}$ from $\boldsymbol{pop_i}$;
    Make up $M$ solutions by selecting the corresponding
      value from $\boldsymbol{x_{rk,i}}$ or $\boldsymbol{x_{mgb,j}}$ according to the OA;
    Evaluate each solution on $i$th task and record the best
      solution $\boldsymbol{x_b}$;
    Use factor analysis to derive a predictive solution $\boldsymbol{x_p}$
      from the above $M$ fitness;
    Evaluate the fitness of $\boldsymbol{x_p}$ on $i$th task;
    Compare $\boldsymbol{x_b}$ and $\boldsymbol{x_p}$, select the better one as the $\boldsymbol{x_{OT}}$;
    If $\boldsymbol{x_{OT}}$ is better than $\boldsymbol{x_{rk,i}}$, $\boldsymbol{x_{OT}}$ replaces $\boldsymbol{x_{rk,i}}$ and set
      $r_{OT} = 1$, else set $r_{OT} = 0$;
End
```

## *D. CDT Method*

To further exploit the similarity between tasks and obtain knowledge for solving different tasks, we propose the CDT method, which allows different dimensional knowledge from one task to transfer to another task. The occurrence of the CDT process is adaptively controlled by a probability parameter $p_{CDT,i,j}$ the same as the OT process. In the CDT process, from the $j$th task to the $i$th task, we calculate the mean and the standard deviation of every dimension of $\boldsymbol{pop_j}$ and $\boldsymbol{pop_i}$. The arithmetical mean center and the standard deviation of the $d$th dimension of $\boldsymbol{pop_i}$ are denoted as $ctr_{i,d}$ and $std_{i,d}$, respectively. Herein, we use a Gaussian distribution to describe the distribution of the dimension $d$ as $\mathcal{N}(ctr_{i,d}, std_{i,d})$. Then, we construct the transferred individual, denoted as $\boldsymbol{x_{CDT}}$, one dimension by one dimension. For the $d$th $(1 \leq d \leq D_i)$ dimension of $\boldsymbol{x_{CDT}}$, the CDT method will transfer the knowledge from a similar dimension in the $j$th task (i.e., the source task). To do this, we first calculate the similarity $\mathrm{sim}_k$ of the $k$th dimension of the $j$th task to the $d$th dimension of the $i$th task as

$$sim_k = \frac{1}{\mathrm{KL}\big(\mathcal{N}\big(ctr_{i,d}, std_{i,d}\big)||\mathcal{N}\big(ctr_{j,k}, std_{j,k}\big)\big) + \varepsilon} \tag{12}$$

where $\mathrm{KL}(A||B)$ refers to the Kullback–Leibler divergence [64] of two (i.e., $A$ and $B$) Gaussian distributions and infinitesimal $\varepsilon$ = 1e-6 is used to avoid dividing by zero. Then, a $D_j$-dimensional probability vector $p_{rs}$ is created for $r$oulette $s$election. The $k$th $(1 \leq k \leq D_j)$ dimension of $p_{rs}$ is calculated as

$$p_{rs,k} = \frac{sim_k}{\sum_{k=1}^{D_j} sim_k}. \tag{13}$$

Then, a dimension $sd$ in the $j$th task will be selected in a roulette scheme according to the probability $p_{rs}$. Afterward, the $d$th dimension of the $\boldsymbol{x_{CDT}}$ is randomly generated based on the Gaussian distribution $\mathcal{N}(ctr_{j,sd}, std_{j,sd})$. In this way, the CDT method can perform the information transfer crossing the similar dimensions between the source task and the target task for positive KT. After all the dimensions of $\boldsymbol{x_{CDT}}$ are built, $\boldsymbol{x_{CDT}}$ is evaluated on the $i$th task (i.e., the target task) and compared with a randomly selected individual $\boldsymbol{x_{rk,i}}$ from $\boldsymbol{pop_i}$. If $\boldsymbol{x_{CDT}}$ is better than $\boldsymbol{x_{rk,i}}$, $\boldsymbol{x_{CDT}}$ will replace $\boldsymbol{x_{rk,i}}$ in $\boldsymbol{pop_i}$. The reward of this CDT process denoted as $r_{\mathrm{CDT}}$ is set to 1. Otherwise, $r_{\mathrm{CDT}}$ is set to 0. Finally, similar to the update of $p_{OT,i,j}$, the parameter $p_{\mathrm{CDT},i,j}$ will be updated by the feedback

**Algorithm 3:** CDT Method

```
Input: Task $i$ (i.e., target task), Task $j$ (i.e., source task),
       $D_i, D_j, \boldsymbol{pop_i}, \boldsymbol{pop_j}$
Output: $r_{CDT}, \boldsymbol{pop_i}$
Begin
    Calculate $ctr_i, std_i$ of $\boldsymbol{pop_i}$ and $ctr_j, std_j$ of $\boldsymbol{pop_j}$;
    For $d$ in $\{1, \ldots, D_i\}$
        Calculate the probability vector $p_{rs}$ according to
        Eq. (12) and Eq. (13);
        Select a dimension $sd$ from $\{1, \ldots, D_j\}$ according
        to the probability vector $p_{rs}$ in a roulette selection
        scheme;
        Sample $x_{CDT,d} = \mathcal{N}(ctr_{j,sd}, std_{j,sd})$;
    End For
    Evaluate $\boldsymbol{x_{CDT}}$ and compare it with a randomly
    selected individual $\boldsymbol{x_{rk,i}}$ from $\boldsymbol{pop_i}$;
    If $\boldsymbol{x_{CDT}}$ is better than $\boldsymbol{x_{rk,i}}, \boldsymbol{x_{CDT}}$ replaces $\boldsymbol{x_{rk,i}}$ and
    set $r_{CDT} = 1$, else set $r_{CDT} = 0$;
End
```

of the transfer as

$$p_{\text{CDT},i,j} = 0.95 \times p_{\text{CDT},i,j} + 0.05 \times r_{\text{CDT}}. \tag{14}$$

The complete process of the CDT method is shown in Algorithm 3. Unlike the existing methods, we adopt a roulette selection to select the dimension of the source task that has a similar distribution to the current dimension of the target task. In this way, the CDT method can exploit the similarity between the global optima of the tasks.

### E. Complete Algorithm

This section describes the complete procedure of the OTMTO algorithm. The details of the complete algorithm are shown in Algorithm 4. First, we initialize the population that is uniformly sampled from the normalized search space of every task. Then, the populations of the tasks undergo the evolutionary process one by one. Herein, we use the DE/rand/1 algorithm as the problem solver for all tasks. After evolving all the individuals in the population, we perform the CTM strategy and the OT method to transfer knowledge between the source task and the target task which are shown in lines 7–11. Note that *rand* shown in lines 7 and 12 refers to a randomly generated number within [0,1]. When performing KT from other source tasks, we adopt the simple strategy that randomly chooses one task as the source task, which is shown in line 6 in Algorithm 4. Next, we perform the CDT method which is shown in lines 12–15. A generation is finished when all populations have finished the evolutionary process, the CTM process, the OT process, and the CDT process. The OTMTO algorithm iterates until the maximum number of FE denoted as *maxFE* is reached. Note that the extra FE caused by the OT method should also be counted and updated, which is shown in line 9 in Algorithm 4. Since different tasks might have DDs, we set $maxFE = 1000 \times \sum D_i$.

**Algorithm 4:** OTMTO

```
Input: Task 1, Task 2, ..., Task $K$
Output: $\{\boldsymbol{x_{gb,1}}, \boldsymbol{x_{gb,2}} \ldots, \boldsymbol{x_{gb,K}}\}$
Parameter: $p_{OT,i,j} = p_{CDT,i,j} = 0.5$ $(i, j \in \{1, \ldots, K\})$,
           $FE = 0$, $maxFE = 1000 \times \Sigma D_i$
1  Begin
2      Randomly initialize the population for each task;
3      While $FE < maxFE$
4          For $i$ in $\{1, \ldots, K\}$
5              Evolve $\boldsymbol{pop_i}$ by DE/rand/1 and update FE;
6              Randomly select a source task $j (j \neq i)$ in
               $\{1, \ldots, K\}$;
7              If rand $< p_{OT,i,j}$
8                  Perform CTM strategy between $i$th task
                   and $j$th task to obtain $\boldsymbol{x_{mgb,j}}$;
                   //Algorithm 1
9                  Perform OT method between $i$th task and
                   $j$th task to obtain $r_{OT}$ and update FE;
                   //Algorithm 2
10                 Update $p_{OT,i,j}$ with $r_{OT}$ by Eq. (11);
11             End If
12             If $rand < p_{CDT,i,j}$
13                 Perform CDT method between $i$th task
                   and $j$th task to obtain $r_{CDT}$ and update
                   FE; //Algorithm 3
14                 Update $p_{CDT,i,j}$ with $r_{CDT}$ by Eq. (14);
15             End If
16         End For
17     End While
18 End
```

### F. Time Complexity Analysis

The computational cost of OTMTO in a generation comes from DE for the optimization of the $i$th task ($i = 1, \ldots, K$), the *INFE* caused by an internal DE in the CTM strategy, and the extra *FE* caused by the OT method and the CDT method. The computational cost of the population reproduction process in DE is ignored since it is small. Since the computational cost of an internal fitness calculation mainly comes from the square calculation shown in (4) and (8), the computational cost of the CTM strategy for the $i$th task is $O(maxINFE \cdot (nb \cdot D + nb))$, where $D = \max(\{D_1, \ldots, D_K\})$. Furthermore, the extra *FE* caused by the OT method are $O(2^{\lceil \log_2 (D+1) \rceil})$ and the extra *FE* caused by the CDT method is $O(1)$ for the $i$th task. In summary, the worst-case computational cost (i.e., *INFE*) of the CTM strategy with $K$ tasks in OTMTO is $O(K \cdot maxINFE \cdot (nb \cdot D + nb))$. The time complexity of the CTM strategy can be reduced by reducing *maxINFE* and *nb*. The experimental study in Section IV-D shows that the performance of OTMTO does not change significantly with different settings of *maxINFE* and *nb*. Moreover, the occurrence of the CTM strategy is adaptively controlled by $p_{OT}$ according to the reward of the transfer. The extra *FE* caused by the OT and the CDT methods are $O(K \cdot (2^{\lceil \log_2 (D+1) \rceil} + 1))$. It can be seen that the extra FE grows linearly with the increase of the dimensionality $D$ when $D$ is small (e.g., $D < 200$). When the extra FE caused by the

OT method has become large, the performance of OTMTO may be worse. How to improve the scalability of OTMTO to high-dimensional problems and many-task problems is a future research direction.

## IV. Experimental Studies

Experimental tests on the MTOP-DD benchmark are carried out in this section to validate the effectiveness and efficiency of the proposed OTMTO algorithm. The performance of OTMTO will be compared with other existing baseline and state-of-the-art EMTO algorithms in the literature.

### *A. Experimental Setting*

*1) CEC17 Benchmark:* To verify the performance of the proposed OTMTO algorithm, we first run a numerical experiment on the CEC17 multitasking benchmark problems, which are widely adopted in benchmarking EMTO algorithms. The CEC17 benchmark includes nine MTOPs, and each problem contains two single-objective optimization tasks. The single-objective optimization functions include Sphere, Rosenbrock, Schwefel, rotated Ackley, rotated Griewank, and rotated Weierstrass. To rotate the function, the solution should be left multiplied with an orthogonal matrix before the calculation of the function value. The problems are divided into three groups, and in each group, the problems are arranged in the order from lowest to the highest similarity between the two tasks. As introduced in [65], to calculate the intertask similarity, 1e6 points are sampled in the unified search space $U$, and the $i$th solution $\boldsymbol{u_i}$ is decoded into $\boldsymbol{x}_{\mathbf{1},i}$ and $\boldsymbol{x}_{\mathbf{2},j}$, corresponding to the original search space of tasks 1 and task 2, respectively. Let rank($\boldsymbol{x}_{\mathbf{1},i}$) and rank($\boldsymbol{x}_{\mathbf{2},i}$) denote the ranks of the $i$th solution with respect to tasks 1 and 2. Then, the similarity denoted as sim, measured as Spearman's rank correlation coefficient, is calculated as

$$sim = \frac{\mathrm{cov}(\mathrm{rank}(\boldsymbol{x}_\mathbf{1}), \mathrm{rank}(\boldsymbol{x}_\mathbf{2}))}{\mathrm{std}(\mathrm{rank}(\boldsymbol{x}_\mathbf{1}))\mathrm{std}(\mathrm{rank}(\boldsymbol{x}_\mathbf{2}))}. \quad (15)$$

In particular, the first group includes three MTOPs with complete intersections. That is, the optimal solutions of the two tasks are the same in the unified search space. The second group includes three MTOPs with partial intersections in which only a part of the dimensions in the optimal solutions of the two tasks are the same. The third group contains three MTOPs with no intersection in which all dimensions are different in the optimal solutions of the two tasks. Note that most of the problems except problem 6 in the CEC17 benchmark have the same dimensionality for the two tasks. For detailed characteristics of these problems, refer to [65].

TABLE I
Functions Used and Similarity Between Tasks for the Proposed MTOP-DD Benchmark Problems With $D_1 = 30$ and $D_2 = 40$

| Problem | Task | Dimensionality | $sim_{go}$ | $sim_{fl}$ |
|---|---|---|---|---|
| 1 | Griewank | 30 | 1.000 | 0.834 |
| | Rastrigin | 40 | 0.750 | |
| 2 | Ackley | 30 | 1.000 | 0.202 |
| | Rastrigin | 40 | 0.750 | |
| 3 | Ackley | 30 | 1.000 | 0.197 |
| | Sphere | 40 | 0.750 | |
| 4 | Griewank | 30 | 0.500 | 0.842 |
| | Sphere | 40 | 0.375 | |
| 5 | Ackley | 30 | 0.500 | 0.198 |
| | Griewank | 40 | 0.375 | |
| 6 | Ackley | 30 | 0.500 | 0.142 |
| | Weierstrass | 40 | 0.375 | |
| 7 | Rastrigin | 30 | 0.000 | 0.833 |
| | Sphere | 40 | 0.000 | |
| 8 | Griewank | 30 | 0.000 | 0.591 |
| | Weierstrass | 40 | 0.000 | |
| 9 | Sphere | 30 | 0.000 | 0.193 |
| | Ackley | 40 | 0.000 | |

*2) Proposed MTOP-DD Benchmark:* Since the existing benchmark cannot satisfy the demand for testing the performance of the EMTO algorithm on MTOP-DD, we propose a benchmark test suite tailored to MTOP-DD, which is highly configurable. Following the design of CEC17, we consider two kinds of similarity between tasks, i.e., the global optima similarity and the function landscape similarity. Let $x^*_{gb,i}$ and $x^*_{gb,j}$ be the global optima of the $i$th task and $j$th task, respectively. To measure the relative global optima similarity, we denote the *n*umber of the same dimensions in $x^*_{gb,i}$ and $x^*_{gb,j}$ as $nsd_{i,j}$. Then, the global optima similarity for the $i$th task relative to the $j$th task denoted as $sim_{go,i,j}$, is calculated as

$$sim_{go,i,j} = \frac{nsd_{i,j}}{D_i}. \quad (16)$$

Since $D_i$ may not be equal to $D_j$, $sim_{go,i,j}$ is not necessarily equal to $sim_{go,j,i}$. $sim_{go,i,j}$ reflects the potential benefit for the $i$th task by transferring the knowledge from the $j$th task. For the function landscape similarity, denoted as $sim_{fl,i,j}$, we measure it by calculating Spearman's rank correlation coefficient according to (15) by sampling 1e6 points in the unified search space, the same way as [65] does. In particular, the proposed benchmark includes nine two-task optimization problems, where the problems can have DDs. They can be divided into three groups similar to the CEC17 benchmark, with different levels of global optima similarity. In each group, the problems have different fitness landscape similarities. In our experiment, we use the settings $D_1$, $D_2 \in \{20, 30, 40, 50\}$, which constitutes $4 \times 4 \times 9 = 144$ multitask problems with different task dimensionalities and different functions to achieve a comprehensive comparison. The detailed data of the benchmark, including the rotation matrix and the optima bias, can be found in the supplementary material. The functions used in the nine problems and the similarities between tasks of the MTOP-DD with $D_1 = 30$ and $D_2 = 40$ are shown in Table I as an example.

*3) Compared Algorithms:* The KT is the key component of the EMTO algorithms. An effective KT method can improve the search performance on multiple tasks compared to single-task optimization algorithms. Therefore, to validate the advantage brought by the proposed KT method, the OTMTO is compared with the single-objective evolutionary algorithm (SOEA) proposed in [65], which is a single-task optimization algorithm. Note that the EC algorithm used in SOEA is DE/rand/1, which is the same as the DE used in the OTMTO algorithm to

TABLE II

EXPERIMENTAL RESULTS ON THE CEC17 BENCHMARK PROBLEMS

| Algorithm | OTMTO | | MFEA | | MFEA2 | | MTGA | | EMTEA | | SODE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Problem | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 |
| 1 | 4.93E-04 | 1.92E+00 | 5.52e-01(+) | 2.36e+02(+) | 5.66e-02(+) | 7.17e+01(+) | 3.01e-02(+) | 5.89e+01(+) | 6.17e-02(+) | 7.33e+01(+) | 2.24e-01(+) | 4.36e+02(+) |
| 2 | 2.34E-08 | 3.76E-14 | 5.89e+00(+) | 2.87e+02(+) | 1.37e+00(+) | 7.27e+01(+) | 1.29e+00(+) | 5.88e+01(+) | 1.22e-01(+) | 4.88e+00(+) | 1.89e+01(+) | 4.35e+02(+) |
| 3 | 2.12E+01 | 1.74E+03 | 2.03e+01(-) | 2.54e+03(+) | 2.01e+01(-) | 2.41e+03(+) | 2.12e+01(=) | 4.91e+03(+) | 2.11e+01(=) | 1.24e+04(+) | 2.12e+01(=) | 1.24e+04(+) |
| 4 | 3.87E+01 | 1.58E-13 | 6.56e+02(+) | 2.28e+01(+) | 3.89e+02(+) | 4.31e-01(+) | 5.10e+01(=) | 1.13e-02(+) | 4.00e+02(+) | 2.84e-01(+) | 4.48e+02(+) | 3.79e-01(+) |
| 5 | 9.84E-06 | 7.05E+01 | 4.06e+00(+) | 1.35e+03(+) | 1.30e+00(+) | 1.67e+02(+) | 4.68e-01(+) | 3.48e+02(+) | 3.41e-01(+) | 1.09e+02(+) | 1.77e+01(+) | 1.09e+03(+) |
| 6 | 6.13E-07 | 2.40E-04 | 2.02e+01(+) | 1.95e+01(+) | 2.00e+01(+) | 2.03e+01(+) | 4.55e+00(+) | 9.06e+00(+) | 1.23e-02(+) | 3.31e-02(+) | 1.89e+01(+) | 3.17e-01(+) |
| 7 | 2.11E+02 | 3.14E+02 | 1.94e+03(+) | 3.36e+02(=) | 3.21e+02(+) | 1.79e+02(-) | 2.04e+02(=) | 6.43e+01(-) | 2.35e+02(+) | 3.08e+02(=) | 1.26e+03(+) | 4.36e+02(+) |
| 8 | 4.27E-04 | 6.82E+00 | 6.20e-01(+) | 2.72e+01(+) | 1.23e-01(+) | 2.37e+01(+) | 1.17e-02(+) | 9.58e+00(=) | 7.59e-02(+) | 5.66e+00(-) | 2.31e-01(+) | 2.89e+01(+) |
| 9 | 3.67E+02 | 1.65E+03 | 6.83e+02(+) | 2.58e+03(+) | 3.65e+02(=) | 2.48e+03(+) | 3.43e+02(=) | 3.74e+03(+) | 4.07e+02(+) | 1.23e+04(+) | 4.41e+02(+) | 1.23e+04(+) |
| Number of +/=/- | | | 8/0/1 | 8/1/0 | 7/1/1 | 8/0/1 | 5/4/0 | 7/1/1 | 8/1/0 | 7/1/1 | 8/1/0 | 9/0/0 |

TABLE III

EXPERIMENTAL RESULTS ON THE PROPOSED MTOP-DD BENCHMARK PROBLEMS WITH $D_1 = 30$ AND $D_2 = 40$

| Algorithm | OTMTO | | MFEA | | MFEA2 | | MTGA | | EMTEA | | SODE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Problem | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 |
| 1 | 9.85E-04 | 4.51E+01 | 1.41e-01(+) | 2.31e+02(+) | 8.73e-03(+) | 1.52e+02(+) | 2.23e-02(+) | 1.60e+02(+) | 4.99e-04(-) | 3.23e+02(+) | 1.20e-04(-) | 3.27e+02(+) |
| 2 | 8.34E-06 | 4.08E+01 | 2.41e+00(+) | 2.34e+02(+) | 1.15e+00(+) | 1.46e+02(+) | 7.13e-01(+) | 1.57e+02(+) | 1.85e-02(+) | 3.24e+02(+) | 1.34e+00(+) | 3.16e+02(+) |
| 3 | 3.98E-08 | 2.95E-15 | 2.39e+00(+) | 2.91e+01(+) | 2.27e-02(+) | 2.71e-01(+) | 4.01e-01(+) | 9.33e+00(+) | 1.72e-02(+) | 7.61e-01(+) | 3.77e+00(+) | 5.19e-01(+) |
| 4 | 3.06E+01 | 1.30E-10 | 1.73e+02(+) | 3.35e+01(+) | 9.64e+01(+) | 3.95e-01(+) | 5.70e+01(+) | 2.37e+01(+) | 6.76e+01(+) | 7.92e-01(+) | 2.06e+02(+) | 5.67e-01(+) |
| 5 | 5.07E-02 | 2.09E-03 | 2.74e+00(+) | 6.43e-01(+) | 1.50e+00(+) | 3.93e-02(+) | 4.01e-01(+) | 3.83e-01(+) | 1.96e-02(-) | 3.08e-02(+) | 4.34e+00(+) | 2.33e-02(+) |
| 6 | 4.66E-02 | 3.37E+00 | 1.29e+01(+) | 1.89e+01(+) | 1.08e+01(+) | 1.32e+01(+) | 5.89e-01(+) | 7.51e+00(+) | 1.92e-02(-) | 5.95e+00(+) | 3.19e+00(+) | 5.53e+00(+) |
| 7 | 2.80E+01 | 2.25E-11 | 1.28e+02(+) | 2.83e+01(+) | 9.35e+01(+) | 2.84e-01(+) | 4.85e+01(+) | 2.68e+01(+) | 8.30e+01(+) | 7.66e-01(+) | 2.06e+02(+) | 5.57e-01(+) |
| 8 | 3.02E-13 | 3.11E+00 | 1.86e-01(+) | 1.67e+01(+) | 1.54e-02(+) | 1.14e+01(+) | 2.47e-02(+) | 6.21e+00(+) | 8.83e-04(+) | 6.04e+00(+) | 1.39e-04(+) | 5.85e+00(+) |
| 9 | 6.81E-15 | 1.11E-01 | 3.14e+00(+) | 7.22e+00(+) | 4.85e-02(+) | 1.62e+01(+) | 1.48e-01(+) | 2.36e+00(+) | 1.88e-03(+) | 1.71e+01(+) | 1.53e-03(+) | 1.79e+01(+) |
| Number of +/=/- | | | 9/0/0 | 9/0/0 | 9/0/0 | 9/0/0 | 9/0/0 | 9/0/0 | 6/0/3 | 9/0/0 | 8/0/1 | 9/0/0 |

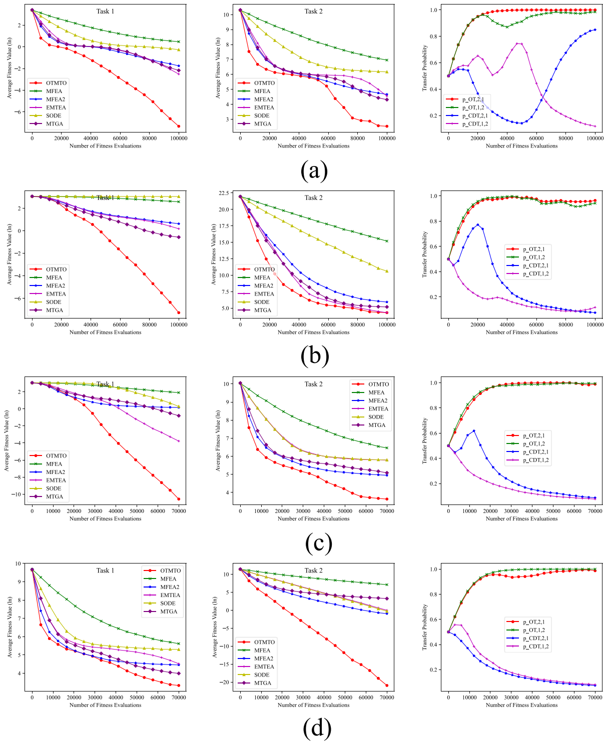
reflect the effects of the proposed KT methods. The compared SOEA using DE is denoted as SODE. To make a comprehensive comparison with OTMTO, we also implement multiple EMTO algorithms. First, MFEA [22] is regarded as the baseline EMTO algorithm. MFEA is an EMTO algorithm based on the USR and uses implicit KT. Next, multiple state-of-the-art EMTO algorithms are implemented and compared with the OTMTO algorithm. MFEA2 [33] is the improved version of MFEA using an online parameter estimation strategy. The EMTO algorithm with explicit autoencoder (EMTEA) proposed in [28] is the compared EMTO algorithm which also uses multipopulation framework and DE/rand/1 as the base solver. Since EMTEA and OTMTO use different mapping methods to facilitate the KT between tasks with heterogeneous search space, the comparison can show the advantage of the proposed CTM strategy and KT methods. Furthermore, the MTGA [25] is a simple and efficient multipopulation-based EMTO algorithm. To enable a fair comparison, all the parameter settings of the compared algorithms are the same as those in their original papers. For the OTMTO algorithm, the experimental settings are as follows.

1) *Maximum FEs: maxFE* $= 1000 \times \sum D_i$.
2) *EC Algorithm for Each Task:* DE/rand/1, *ps* $= 100$, $F = 0.5$, $Cr = 0.6$, *mutation scheme* = rand/1, *crossover scheme* = binary crossover, and *selection scheme* = elitism selection.
3) *Initial KT Probability:* $p_{OT,i,j} = p_{CDT,i,j} = 0.5$, $i \neq j$, and $i$, $j \in \{1, 2\}$.
4) *EC Algorithm for the CTM Process:* DE/best/1, $nb = 5$, inps $= 50$, $F = 0.5$, $Cr = 0.9$, *mutation scheme* = best/1, *crossover scheme* = binary crossover, *selection scheme* = elitism selection, and *maxINFE* $= 500$.

*4) Performance Measure:* All algorithms terminate and output the historical best fitness for the two tasks after the maximum FE is reached in a single run. To reduce the bias brought by randomness, all algorithms run 20 times independently, each time with a different random seed. The mean values of the best fitness obtained by the EMTO algorithms over 20 independent runs are used for comparisons. The Wilcoxon rank-sum test at the significance level of 0.05 is carried out on the experimental results. Furthermore, to quantitatively evaluate the performances of the EMTO algorithms, the performance metric in the CEC17 [65] is also used to analyze the experimental results.

## *B. Results and Comparison*

The experimental results of the OTMTO algorithm and the compared EMTO algorithms on the CEC17 multitask optimization benchmark problems are shown in Table II. There are nine two-task optimization problems in total, which add up to 18 tasks. Note that most of the problems in CEC17 are MTOPs with the same dimensionalities except problem 6. The listed results are the mean values over 20 independent runs. Moreover, the symbols "+", "=", and "−" indicate that the OTMTO algorithm is significantly better than, equal to, or significantly worse than the compared EMTO algorithms, respectively. The last row of the table summarizes the total number of obtained "+", "=", and "−" when comparing the OTMTO algorithm with the compared EMTO algorithms. First, we observe that OTMTO surpasses the single-task SODE on almost all tasks, which shows that the proposed EMTO algorithm gives a positive KT. That is, transferring knowledge from other tasks can improve the search performance compared to the manner that optimizes the tasks independently. Moreover, it can be seen that the OTMTO algorithm significantly outperforms the baseline EMTO algorithm MFEA on 16 tasks while performing significantly worse than MFEA on only one task. In addition, the OTMTO algorithm outperforms MFEA2 on 15 tasks. These results show that the OTMTO algorithm offers great advantages over MFEA and MFEA2,



(a)

(b)

(c)

(d)

Fig. 2. Convergence curves of the competing EMTO algorithms and the curves of the KT probabilities between the two tasks (i.e., $p_{OT}$ and $p_{CDT}$) of OTMTO on the tested problems. (a) Problem 1 of the CEC17 benchmark. (b) Problem 5 of the CEC17 benchmark. (c) Problem 2 of the proposed MTOP-DD benchmark. (d) Problem 7 of the proposed MTOP-DD benchmark.

indicating that the proposed OT method can significantly improve KT quality. Since the OTMTO algorithm is not the first to adopt the multipopulation framework, it is compared to the MTGA and EMTEA algorithms, which also adopt the multipopulation framework. The comparative results show that the OTMTO algorithm outperforms the EMTEA algorithm on most of the tasks. This indicates the advantage of the CTM strategy and the OT method in performing positive KT compared to the denoising autoencoder. Furthermore, the OTMTO algorithm is compared to the state-of-the-art EMTO algorithm MTGA. The results show that OTMTO significantly outperforms MTGA on 12 tasks and is worse than MTGA on only 1 task. Hence, OTMTO is generally better than MTGA on the CEC17 benchmark problems.

Next, we conduct experiments on the proposed MTOP-DD benchmark that contains 144 multitask problems with different task dimensionalities. Especially, the results on the nine proposed benchmark problems with $D_1 = 30$ and $D_2 = 40$ (i.e., the problems in Table I) are shown in Table III. The results on the benchmark problems with other different dimensionality combinations are all presented in Table S.I to Table S.XV in the supplementary material. Moreover, the comparative results of comparing the OTMTO algorithm with other algorithms using the Wilcoxon rank-sum test on the nine proposed benchmark problems with $D_1$, $D_2 \in \{20, 30, 40, 50\}$ (i.e., totally $4 \times 4 = 16$ combinations) are summarized shown in Table S.XVI in the supplementary material.

As can be observed from Table III, the OTMTO algorithm can achieve positive KT on most of the MTOP-DDs compared to the single-task algorithm SODE. Next, the OTMTO algorithm significantly outperforms MFEA, MFEA2, EMTEA, and MTGA on most of the tasks. This shows great improvement brought by the CTM strategy and the OT method. Note that we can observe that the EMTEA perform even worse than the single-task algorithm SODE on some tasks, such as the task 2 of problem 3, which indicates the negative KT. This shows that the existing EMTO algorithms are not well suited for the MTOP-DD and further validates the contribution of this article. The effectiveness of the OTMTO algorithm is also confirmed on the MTOP-DDs with different settings of $D_1$ and $D_2$. From Table S.XVI in the supplementary material, we can see that the OTMTO algorithm outperforms the compared algorithms on most of the problems. This shows that the OTMTO algorithm is an effective EMTO algorithm for handling MTOP-DD. Moreover, the quantitative analysis [65] is provided in Section C in the supplementary material. The conclusion in the analysis is that the OTMTO algorithm achieves the overall best performance among all the EMTO algorithms.

To further show the advantage of OTMTO in search efficiency, the convergence curves of the competing EMTO algorithms on several tested problems are plotted in Fig. 2. It can be observed that the OTMTO can obtain good solutions faster along the search process compared to the state-of-the-art EMTO algorithms. Moreover, the changing transfer probabilities (i.e., $p_{OT}$ and $p_{CDT}$) of the OT and CDT methods along the search process are also shown in Fig. 2. It can be observed that the $p_{CDT}$ tends to increase in the early stage and decrease in the late stage of the optimization process. This indicates that the CDT method is useful in the early stage to help locate promising region fast by transferring useful dimensions. Note that the general trend of $p_{OT}$ is increasing in Fig. 2. This indicates that the CTM strategy together with the OT method can always provide positive transfer no matter whether the two tasks of the test problem are with heterogeneous fitness landscapes (e.g., problem 5 of the CEC17 benchmark) or with DDs (e.g., problem 2 of the proposed MTOP-DD benchmark).

Moreover, to testify the scalability of the OTMTO algorithm, we carry out experiments on the three-task optimization problems. The detailed problem settings and the results are given in Table S.XIX in the supplementary material. The results show that our OTMTO algorithm significantly outperforms the SODE algorithm and MFEA.

## *C. Effects of the Components*

In this section, we investigate the effects of the components in the OTMTO algorithm, which are the CTM strategy, the OT method, and the CDT method.

To validate the effectiveness of the CTM strategy, we formulate multiple OTMTO variants that perform CTM by single-layer linearized autoencoder, kernelized autoencoder, and affine transformation, denoted as OTMTO-LA, OTMTO-KA, and OTMTO-AT, respectively. These variants differ from the OTMTO algorithm only in the mapping scheme that maps $\boldsymbol{x}_{gb,j}$ to $\boldsymbol{x}_{mgb,j}$. Specifically, for OTMTO-LA, the mapping

TABLE IV
EXPERIMENTAL RESULTS OF THE OTMTO ALGORITHM AND ITS VARIANTS FOR TESTING THE EFFECTIVENESS OF THE CTM STRATEGY ON THE PROPOSED MTOP-DD BENCHMARK PROBLEMS WITH $D_1 = 30$ AND $D_2 = 40$

| Algorithm | OTMTO | | OTMTO-LA | | OTMTO-KA | | OTMTO-AT | | OTMTO-ITS1 | |
|---|---|---|---|---|---|---|---|---|---|---|
| Problem | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 |
| 1 | 9.85E-04 | 4.51E+01 | 5.28e-04(-) | 3.23e+02(+) | 2.28e-03(+) | 2.32e+02(+) | 2.74e-05(-) | 3.34e+02(+) | 7.96e-07(-) | 2.33e+02(+) |
| 2 | 8.34E-06 | 4.08E+01 | 4.25e-02(+) | 3.14e+02(+) | 5.88e-02(+) | 3.25e+02(+) | 2.31e-02(+) | 3.41e+02(+) | 4.29e-03(+) | 2.42e+02(+) |
| 3 | 3.98E-08 | 2.95E-15 | 7.83e-02(+) | 7.52e-01(+) | 2.13e-03(+) | 5.55e-01(+) | 1.87e-02(+) | 4.02e+00(+) | 5.85e-03(+) | 1.33e-04(+) |
| 4 | 3.06E+01 | 1.30E-10 | 2.12e+02(+) | 7.77e-01(+) | 1.17e+02(+) | 1.44e-01(+) | 1.67e+02(+) | 3.90e+00(+) | 1.77e+02(+) | 9.92e-05(+) |
| 5 | 5.07E-02 | 2.09E-03 | 7.71e-02(+) | 3.26e-02(+) | 2.99e-03(-) | 2.25e-02(+) | 1.72e-02(-) | 1.58e-01(+) | 6.14e-03(-) | 1.15e-05(=) |
| 6 | 4.66E-02 | 3.37E+00 | 4.34e-02(-) | 3.48e+00(=) | 5.15e-02(+) | 4.83e+00(+) | 8.67e-02(+) | 9.72e+00(+) | 1.76e-02(-) | 9.83e-01(-) |
| 7 | 2.80E+01 | 2.25E-11 | 2.07e+02(+) | 8.08e-01(+) | 1.24e+02(+) | 1.69e-01(+) | 1.65e+02(+) | 3.44e+00(+) | 1.76e+02(+) | 8.44e-05(+) |
| 8 | 3.02E-13 | 3.11E+00 | 4.07e-04(+) | 3.66e+00(=) | 1.68e-03(+) | 4.32e+00(+) | 2.73e-05(+) | 3.94e+00(+) | 1.36e-06(+) | 7.88e-01(-) |
| 9 | 6.81E-15 | 1.11E-01 | 2.92e-03(+) | 7.79e-01(+) | 5.56e-03(+) | 1.21e-01(=) | 2.02e-04(+) | 1.36e+00(+) | 5.85e-06(+) | 1.46e-02(=) |
| Number of +/=/- | | | 7/0/2 | 7/2/0 | 8/0/1 | 8/1/0 | 7/0/2 | 9/0/0 | 6/0/3 | 5/2/2 |

TABLE V
EXPERIMENTAL RESULTS OF THE OTMTO ALGORITHM AND ITS VARIANTS FOR TESTING THE EFFECTIVENESS OF THE OT AND CDT METHODS ON THE PROPOSED MTOP-DD BENCHMARK PROBLEMS WITH $D_1 = 30$ AND $D_2 = 40$

| Algorithm | OTMTO | | OTMTO-SBX | | OTMTO-UBX | | OTMTO-ITS2 | | OTMTO-w/o-CDT | |
|---|---|---|---|---|---|---|---|---|---|---|
| Problem | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 |
| 1 | 9.85E-04 | 4.51E+01 | 2.76e-03(+) | 2.43e+02(+) | 1.36e-03(+) | 1.66e+02(+) | 1.23e-03(=) | 4.11e+01(=) | 2.96e-03(+) | 6.70e+01(+) |
| 2 | 8.34E-06 | 4.08E+01 | 3.00e-02(+) | 2.32e+02(+) | 4.12e-03(+) | 1.66e+02(+) | 4.66e-05(=) | 4.05e+01(=) | 4.12e-03(+) | 8.24e+01(+) |
| 3 | 3.98E-08 | 2.95E-15 | 1.23e-03(+) | 4.47e-04(+) | 9.93e-04(+) | 1.93e-04(+) | 2.05e-02(+) | 3.18e-13(+) | 1.29e-03(+) | 1.45e-09(+) |
| 4 | 3.06E+01 | 1.30E-10 | 1.81e+02(+) | 1.69e-01(+) | 1.41e+02(+) | 6.73e-03(+) | 2.75e+01(=) | 2.96e-11(=) | 4.95e+01(+) | 8.00e-06(+) |
| 5 | 5.07E-02 | 2.09E-03 | 1.31e-03(-) | 2.12e-03(=) | 1.58e-03(-) | 2.35e-03(=) | 1.05e-02(=) | 1.73e-03(=) | 8.86e-04(-) | 1.23e-03(=) |
| 6 | 4.66E-02 | 3.37E+00 | 2.48e-03(-) | 2.30e+00(-) | 2.38e-03(-) | 1.79e+00(-) | 1.00e-02(=) | 2.30e+00(=) | 2.44e-03(-) | 2.00e+00(-) |
| 7 | 2.80E+01 | 2.25E-11 | 1.45e+02(+) | 8.80e-02(+) | 1.29e+02(+) | 9.32e-03(+) | 2.72e+01(=) | 8.38e-10(=) | 5.19e+01(+) | 2.96e-06(+) |
| 8 | 3.02E-13 | 3.11E+00 | 7.40e-04(+) | 8.20e-01(-) | 1.85e-03(+) | 7.89e-01(-) | 1.60e-03(=) | 2.76e+00(=) | 1.97e-03(+) | 2.08e+00(=) |
| 9 | 6.81E-15 | 1.11E-01 | 2.26e-06(+) | 3.03e-02(=) | 1.92e-06(+) | 7.51e-03(=) | 2.91e-15(=) | 5.07e-02(=) | 9.05e-11(+) | 8.50e-04(=) |
| Number of +/=/- | | | 7/0/2 | 5/2/2 | 7/0/2 | 5/2/2 | 1/8/0 | 1/8/0 | 7/0/2 | 5/3/1 |

between tasks is implemented by left multiplying by a transformed matrix calculated in the same way as EMTEA. For OTMTO-KA, the mapping is achieved by learning a nonlinear polynomial kernel as [29]. For OTMTO-AT, the mapping is based on an affine transformation as [66]. Note that when the two tasks have different dimensionality, we pad zeros to the solutions of the task with lower dimensionality to satisfy the same dimensionality condition for these mapping methods. Moreover, an OTMTO variant named OTMTO-ITS1 that performs intratask sampling rather than CTM is formulated to investigate the effectiveness of the CTM strategy. That is, the OTMTO-ITS1 is without CTM and the CTM process is replaced by sampling an individual from the Gaussian distribution of the population of the target task, while the other processes including the OED and CDT process remain the same as the OTMTO algorithm.

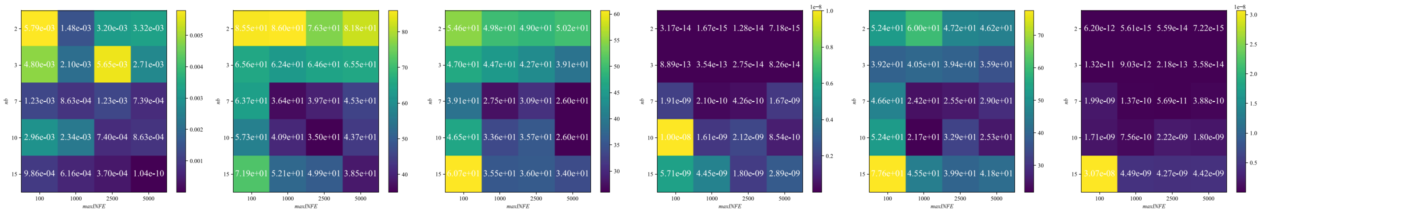
To validate the effectiveness of the OT method, we also formulate two OTMTO variants that perform simulated binary crossover (SBX) and uniform binary crossover (UBX) to transfer knowledge. They are named OTMTO-SBX and OTMTO-UBX, respectively. In particular, the crossover operation is performed on the mapped individual $\boldsymbol{x}_{\boldsymbol{mgb},\boldsymbol{j}}$ obtained from the $j$th task by the CTM strategy and $\boldsymbol{x}_{\boldsymbol{rk},\boldsymbol{i}}$ for $M + 1$ times (i.e., same extra FEs) to obtain the best transferred individual similar to the OT method does. Other components, such as CTM and CDT remain unchanged in these variants.

To validate the effectiveness of the CDT method, an OTMTO variant named OTMTO-ITS2 that performs intratask sampling rather than CDT is formulated. In OTMTO-ITS2, the CDT process is replaced by sampling an individual from the Gaussian distribution of the population of the target task and the rest processes, including the CTM and OED processes remain the same as the OTMTO algorithm. Moreover, an OTMTO variant named OTMTO-w/o-CDT where the CDT process is removed is formulated.

The results of the average final fitness and Wilcoxon rank sum test over 20 independent runs are shown in Tables IV and V. From Table IV, we observe that the OTMTO algorithm generally outperforms the OTMTO variants using other mapping methods like the OTMTO-LA, OTMTO-KA, OTMTO-AT, and OTMTO-ITS1. These results show that the proposed CTM strategy can obtain better transferred individuals for the OT process. From Table V, the OTMTO algorithm performs better than OTMTO-SBX and OTMTO-UBX, which validates the effectiveness of the OT method. Moreover, the OTMTO algorithm performs better than OTMTO-ITS2 and OTMTO-w/o-CDT, which validates the effectiveness of the CDT method.

### D. Parameter Sensitivity

In this section, we investigate the sensitivity of the parameter setting in the CTM strategy including *nb* and *maxINFE*. In particular, *nb* controls the quantity of information (number of top individuals) used to reconstruct the individual in the mapped search space, and *maxINFE* controls the number of internal FE for solving the CTM optimization problem. We run experiments on problems 1, 4, and 7 of the proposed MTOP-DD benchmark in Table I. The parameter settings of OTMTO are $nb \in \{2, 3, 7, 10, 15\}$ and $maxINFE \in \{100, 1000, 2500, 5000\}$. The experimental results of the average final fitness over 20 independent runs are shown in the form of a heatmap in Fig. 3. It can be seen that on problem 1 with high $sim_{go,1,2}$, the OTMTO algorithm tends to obtain better results with a higher *nb*. When the tasks of the problem have medium or low $sim_{go,1,2}$ (i.e., problems 4 and 7), the performance of



(a) (b) (c)

Fig. 3. Heatmap of performance of the OTMTO algorithm with different parameter settings of *nb* and *maxINFE* on the proposed MTOP-DD benchmark with $D_1 = 30$, $D_2 = 40$. (a) Tasks 1 and 2 in problem 1. (b) Tasks 1 and 2 in problem 4. (c) Tasks 1 and 2 in problem 7.

the OTMTO algorithm deteriorates as *nb* increases. It seems that the parameter setting of *nb* is related to the similarity of the global optima of the MTOP-DD. For the parameter *maxINFE*, it can be observed that the OTMTO tends to obtain better results with higher *maxINFE* (e.g., 5000) compared with the lower *maxINFE* (e.g., 100) in general. This is because with more *INFE* we can solve the CTM optimization problem better and then obtain a mapped individual with higher quality. Since the internal optimization process requires extra computational cost, we use the setting of a relatively small *maxINFE* = 500, which includes only ten internal generations for solving the CTM optimization.

### *E. Real-World Application Study*

To further investigate the performance of the proposed OTMTO algorithm, a real-world application study is carried out. The double pole balancing (DPB) problem is a classic practical application problem for testing the evolutionary learning system. In the DPB problem, the objective of a task is formulated as

$$\max_{\pi} R(\tau, \pi) \tag{17}$$

where $R$ is the accumulated reward of the controller under the environment of the task, $\tau$ is the parameter of the task, and $\pi$ denotes the controller. A double pole controlling task can be identified by the *l*ength of the *s*horter pole $l_s$ while the longer one is fixed to 1.0 m. Herein, $\tau$ is represented by $l_s$. For the controller $\pi$, the input contains six variables including the cart position, the cart velocity, the two rotation angles of the two poles, and the two angular velocities of the two poles. The output of the controller is the force applied to the cart. We use a simple three-layer feedforward neural network as the controller $\pi$ following the existing work [33]. Note that different hyperparameter settings, such as the *n*umber of *hidden* neurons $n_{hidden}$ in the neural network lead to different controllers with different number of parameters (i.e., dimensionality). For example, $n_{\text{hidden}} = 8$ means that $\pi$ contains 56 weights to be learned (i.e., 6 inputs × 8 hidden neurons + 8 hidden neurons × 1 output = 56). Therefore, the dimensionality of such an optimization task is 56. In our experiment, we formulate five tasks as follows.

1) *T1* : $l_s = 0.6$, $n_{\text{hidden}} = 8$, and dimensionality $= 6 \times 8 + 8 \times 1 = 56$.
2) *T2* : $l_s = 0.65$, $n_{\text{hidden}} = 8$, and dimensionality $= 6 \times 8 + 8 \times 1 = 56$.
3) *T3* : $l_s = 0.7$, $n_{\text{hidden}} = 8$, and dimensionality $= 6 \times 8 + 8 \times 1 = 56$.
4) *T4* : $l_s = 0.6$, $n_{\text{hidden}} = 6$, and dimensionality $= 6 \times 6 + 6 \times 1 = 42$.
5) *T5* : $l_s = 0.65$, $n_{\text{hidden}} = 6$, and dimensionality $= 6 \times 6 + 6 \times 1 = 42$.

TABLE VI
SUCCESS RATES OF THE OTMTO AND COMPETING ALGORITHMS OVER 50 INDEPENDENT RUNS ON THE TWO-TASK DPB PROBLEMS

| Algorithm | DPB1<br>(T1, T2) | DPB2<br>(T1, T3) | DPB3<br>(T2, T3) | DPB4<br>(T2, T4) | DPB5<br>(T3, T4) | DPB6<br>(T3, T5) |
|---|---|---|---|---|---|---|
| OTMTO | **92%, 74%** | 82%, 30% | **62%, 40%** | **74%, 86%** | **34%,** 78% | **36%, 78%** |
| SODE | 82%, 54% | 82%, 16% | 54%, 16% | 54%, **86%** | 16%, **86%** | 16%, 66% |
| EMTEA | 82%, 64% | **86%, 36%** | 52%, 32% | **72%,** 84% | **34%,** 84% | 32%, 66% |
| MTGA | 22%, 4% | 18%, 6% | 10%, 4% | 12%, 26% | 4%, 26% | 0%, 14% |

For the FE, a solution in OTMTO and other EMTO algorithms is a neural network controller with its weights, and the fitness of the solution is the time that the cart maintains stability by the controller in simulation process. A task is regarded to be solved (i.e., the algorithm is successful) if the controller can be optimized to make the cart maintain stability for more than 30 min in simulated time. Based on the five formulated tasks, we set up six multitask DPB problems with different combinations of the tasks to testify the performance of the EMTO algorithms. The DPB problem instances are denoted as DPB1, DPB2, ..., and DPB6, as shown in Table VI, with each DPB having two tasks. The EMTO algorithms run on each DPB instance with 50 independent times. The average success rates over the 50 independent runs of the EMTO algorithms are reported in Table VI. The best results are marked in *boldface*. In most cases, our OTMTO is superior to the SODE and the EMTEA that both use DE as the base solver, not only on the problems whose tasks are with the same dimensionalities but also on those problems whose tasks are with DDs. The proposed OTMTO also outperforms the state-of-the-art MTGA on all problems. The results show the effectiveness of OTMTO on real-world MTOPs in practical application.

To further investigate the time complexity of the OTMTO algorithm, the running time that the EMTO algorithms need to obtain the optimal controller is reported in Table S.XX in the supplementary material. The results in Table S.XX show that our OTMTO algorithm offers advantage in running time. This indicates the proposed methods in the OTMTO algorithm are effective and efficient.

TABLE VII
SUCCESS RATES OF THE OTMTO AND SINGLE-TASK ALGORITHM OVER 50 INDEPENDENT RUNS ON THE FIVE-TASK DPB PROBLEMS

| Algorithm | T1 | T2 | T3 | T4 | T5 |
|---|---|---|---|---|---|
| OTMTO | **94%** | **70%** | **30%** | **88%** | 52% |
| SODE | 82% | 54% | 16% | 86% | **66%** |

Moreover, to testify the scalability of the OTMTO algorithm, we take experiments on a five-task DPB problem that contains all the five tasks T1, T2, T3, T4, and T5. The results are given in Table VII. It can be seen that our OTMTO still outperforms the compared SODE algorithm on four tasks, which shows the positive transfer brought by the proposed methods.

## V. CONCLUSION

In this article, we mainly addressed two issues in the design of effective EMTO algorithms for MTOPs. First, the existing KT methods are not well suited to the MTOP-DD, which is quite common in real-world applications. Second, the tasks in MTOP may have different degrees of similarity in different dimensions, and the existing KT methods based on the crossover operator with universal probability can cause the negative transfer. To address the above issues, we proposed the OTMTO algorithm, which includes the CTM strategy and the OT method. The CTM strategy is carried out before the OT method to map the individual from a search space to another search space with DDs. In this way, the OT method can be carried out on the individuals under the same search space to find their best combination of dimensions. We showed that performing the OED process on the CTM-obtained mapped individual of the source task and a random selected individual of the target task is enough to offer a high-quality KT. Moreover, another advantage of the OT is that it requires much fewer FE to find the best combination than the exhaustive search. Furthermore, we proposed the CDT method, which allows the KT between different dimensions of two tasks to improve the KT quality. We verified the effectiveness and efficiency of the proposed OTMTO algorithm on both the commonly used CEC17 MTOP benchmark and the proposed MTOP-DD benchmark. To testify the effectiveness of the proposed OTMTO on a real-world problem, experiments on the DPB application problem were carried out and the results showed the superiority of OTMTO. In the future, we will further study how to improve and extend the OTMTO algorithm to solve many-task optimization problems effectively and efficiently.

## REFERENCES

[1] Z.-H. Zhan, L. Shi, K. C. Tan, and J. Zhang, "A survey on evolutionary computation for complex continuous optimization," *Artif. Intell. Rev.*, vol. 55, no. 1, pp. 59–110, 2022.

[2] Z.-H. Zhan *et al.*, "Matrix-based evolutionary computation," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 2, pp. 315–328, April 2022.

[3] H. T. T. Binh and S. H. Ngo, "Survivable flows routing in large scale network design using genetic algorithm," in *Advanced in Computer Science and Its Applications*, vol. 279. Berlin, Germany: Springer, 2014, pp. 345–351.

[4] J.-R. Jian, Z.-G. Chen, Z.-H. Zhan, and J. Zhang, "Region encoding helps evolutionary computation evolve faster: A new solution encoding scheme in particle swarm for large-scale optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 779–793, Aug. 2021.

[5] Z.-J. Wang, Z.-H. Zhan, S. Kwong, H. Jin, and J. Zhang, "Adaptive granularity learning distributed particle swarm optimization for large-scale optimization," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1175–1188, Mar. 2021.

[6] A. Muruganantham, K. C. Tan, and P. Vadakkepat, "Evolutionary dynamic multiobjective optimization via Kalman filter prediction," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2862–2873, Dec. 2016.

[7] X.-F. Liu *et al.*, "Neural network-based information transfer for dynamic optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1557–1570, May 2020.

[8] Y. Jiang, Z.-H. Zhan, K. C. Tan, and J. Zhang, "Optimizing niche center for multimodal optimization problems," *IEEE Trans. Cybern.*, early access, Dec. 17, 2021, doi: 10.1109/TCYB.2021.3125362.

[9] A. Ahrari, S. Elsayed, R. Sarker, D. Essam, and C. A. C. Coello, "Static and dynamic multimodal optimization by improved covariance matrix self-adaptation evolution strategy with repelling subpopulations," *IEEE Trans. Evol. Comput.*, early access, Oct. 1, 2021, doi: 10.1109/TEVC.2021.3117116.

[10] S.-C. Liu, Z.-H. Zhan, K. C. Tan, and J. Zhang, "A multiobjective framework for many-objective optimization," *IEEE Trans. Cybern.*, early access, Aug. 16, 2021, doi: 10.1109/TCYB.2021.3082200.

[11] Z. H. Zhan, J. Li, J. Cao, J. Zhang, H. S. H. Chung, and Y.-H. Shi, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, Apr. 2013.

[12] L. M. Antonio and C. A. C. Coello, "Coevolutionary multiobjective evolutionary algorithms: survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 851–865, Dec. 2018.

[13] J.-Y. Li, Z.-H. Zhan, and J. Zhang, "Evolutionary computation for expensive optimization: A survey," *Mach. Intell. Res.*, vol. 19, no. 1, pp. 3–23, 2022.

[14] S.-H. Wu, Z.-H. Zhan, and J. Zhang, "SAFE: Scale-adaptive fitness evaluation method for expensive optimization problems," *IEEE Trans. Evol. Comput.*, vol. 25, no. 3, pp. 478–491, Jun. 2021.

[15] J.-Y. Li, Z.-H. Zhan, H. Wang, and J. Zhang, "Data-driven evolutionary algorithm with perturbation-based ensemble surrogates," *IEEE Trans. Cybern.*, vol. 51, no. 8, pp. 3925–3937, Aug. 2021.

[16] Z. G. Chen, Y. Lin, Y.-J. Gong, Z.-H. Zhan, and J. Zhang, "Maximizing lifetime of range-adjustable wireless sensor networks: A neighborhood-based estimation of distribution algorithm," *IEEE Trans. Cybern.*, vol. 51, no. 11, pp. 5433–5444, Nov. 2021.

[17] X. F. Liu, Z.-H. Zhan, J. D. Deng, Y. Li, T. Gu, and J. Zhang, "An energy efficient ant colony system for virtual machine placement in cloud computing," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 113–128, Feb. 2018.

[18] X. Zhang, K.-J. Du, Z.-H. Zhan, S. Kwong, T.-L. Gu, and J. Zhang, "Cooperative coevolutionary bare-bones particle swarm optimization with function independent decomposition for large-scale supply chain network design with uncertainties," *IEEE Trans. Cybern.*, vol. 50, no. 10, pp. 4454–4468, Oct. 2020.

[19] S. Z. Zhou, Z. H. Zhan, Z.-G. Chen, S. Kwong, and J. Zhang, "A multiobjective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6784–6798, Nov. 2021, doi: 10.1109/TITS.2020.2994779.

[20] Z. G. Chen, Z.-H. Zhan, S. Kwong, and J. Zhang, "Evolutionary computation for intelligent transportation in smart cities: A survey," *IEEE Comput. Intell. Mag.*, vol. 17, no. 2, pp. 83–102, May, 2022.

[21] 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.

[22] 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.

[23] 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.*, 2017, pp. 1295–1302.

[24] L. Zhou *et al.*, "Toward adaptive knowledge transfer in multifactorial evolutionary computation," *IEEE Trans. Cybern.*, vol. 51, no. 5, pp. 2563–2576, May 2021, doi: 10.1109/TCYB.2020.2974100.

[25] D. Wu and X. Tan, "Multitasking genetic algorithm (MTGA) for fuzzy system optimization," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 6, pp. 1050–1061, Jun. 2020.

[26] 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.

[27] 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.

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

[29] L. Zhou, L. Feng, A. Gupta, and Y.-S. Ong, "Learnable evolutionary search across heterogeneous problems via kernelized autoencoding," *IEEE Trans. Evol. Comput.*, vol. 25, no. 3, pp. 567–581, Jun. 2021.

[30] G. H. Li, Q. Z. Lin, and W. F. Gao, "Multifactorial optimization via explicit multipopulation evolutionary framework," *Inf. Sci.*, vol. 512, pp. 1555–1570, Feb. 2020.

[31] J.-Y. Li, Z.-H. Zhan, K.-C. Tan, and J. Zhang, "A meta-knowledge transfer-based differential evolution for multitask optimization," *IEEE Trans. Evol. Comput.*, early access, Nov. 29, 2021, doi: 10.1109/TEVC.2021.3131236.

[32] X. Zheng, A. K. Qin, M. Gong, and D. Zhou, "Self-regulated evolutionary multitask optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 16–28, Feb. 2020.

[33] 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.

[34] Y. Chen, J. Zhong, L. Feng, and J. Zhang, "An adaptive archive-based evolutionary framework for many-task optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 3, pp. 369–384, Jun. 2020.

[35] R.-T. Liaw and C. Ting, "Evolutionary many-tasking based on biocoenosis through symbiosis: A framework and benchmark problems," in *Proc. IEEE Congr. Evol. Comput.*, 2017, pp. 2266–2273.

[36] Z. Liang, X. Xu, L. Liu, Y. Tu, and Z. Zhu, "Evolutionary many-task optimization based on multi-source knowledge transfer," *IEEE Trans. Evol. Comput.*, early access, Aug. 2, 2021, doi: 10.1109/TEVC.2021.3101697.

[37] T. B. Thang, T. C. Dao, N. H. Long, and H. T. T. Binh, "Parameter adaptation in multifactorial evolutionary algorithm for many-task optimization," *Memet. Comput.*, vol. 13, no. 4, pp. 433–446, Dec. 2021.

[38] L. Feng *et al.*, "Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3143–3156, Jun. 2021.

[39] F. Zhang, Y. Mei, S. Nguyen, M. Zhang, and K. C. Tan, "Surrogate-assisted evolutionary multitask genetic programming for dynamic flexible job shop scheduling," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 651–665, Aug. 2021, doi: 10.1109/TEVC.2021.3065707.

[40] 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.

[41] 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.

[42] Z. Liang, W. Liang, Z. Wang, X. Ma, L. Liu, and Z. Zhu, "Multiobjective evolutionary multitasking with two-stage adaptive knowledge transfer based on population distribution," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Jul. 26, 2021, doi: 10.1109/TSMC.2021.3096220.

[43] 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.

[44] H. Wang, L. Feng, Y. Jin, and J. Doherty, "Surrogate-assisted evolutionary multitasking for expensive minimax optimization in multiple scenarios," *IEEE Comput. Intell. Mag.*, vol. 16, no. 1, pp. 34–48, Feb. 2021.

[45] K. Chen, B. Xue, M. Zhang, and F. Zhou, "An evolutionary multitasking-based feature selection method for high-dimensional classification," *IEEE Trans. Cybern.*, early access, Dec. 31, 2020, doi: 10.1109/TCYB.2020.3042243.

[46] J. Luo, A. Gupta, Y. Ong, and Z. Wang, "Evolutionary optimization of expensive multiobjective problems with co-sub-Pareto front Gaussian process surrogates," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1708–1721, May 2019.

[47] L. Bai, W. Lin, A. Gupta, and Y.-S. Ong, "From multitask gradient descent to gradient-free evolutionary multitasking: A proof of faster convergence," *IEEE Trans. Cybern.*, early access, Mar. 11, 2021, doi: 10.1109/TCYB.2021.3052509.

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

[49] B. Da, A. Gupta, and Y.-S. Ong, "Curbing negative influences online for seamless transfer evolutionary optimization," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4365–4378, Dec. 2019.

[50] H. T. T. Binh, T. B. Thang, N. B. Long, N. V. Hoang, and P. D. Thanh, "Multifactorial evolutionary algorithm for inter-domain path computation under domain uniqueness constraint," in *Proc. IEEE Congr. Evol. Comput.*, 2020, pp. 1–8.

[51] C. Yang, J. Ding, Y. Jin, C. Wang, and T. Chai, "Multitasking multiobjective evolutionary operational indices optimization of beneficiation processes," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1046–1057, Jul. 2019.

[52] A. D. Martinez, J. D. Ser, E. Osaba, and F. Herrera, "Adaptive multi-factorial evolutionary optimization for multi-task reinforcement learning," *IEEE Trans. Evol. Comput.*, early access, May 25, 2021, doi: 10.1109/TEVC.2021.3083362.

[53] T. B. Thang, N. B. Long, N. V. Hoang, and H. T. T. Binh, "Adaptive knowledge transfer in multifactorial evolutionary algorithm for the clustered minimum routing cost problem," *Appl. Soft. Comput.*, vol. 105, Jul. 2021, Art. no. 107253.

[54] D. C. Montgomery, *Design and Analysis of Experiments*, 5th ed. NewYork, NY, USA: Wiley, 2000.

[55] Z.-H. Zhan, J. Zhang, Y. Li, and Y.-H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, pp. 832–847, Dec. 2011.

[56] Y.-W. Leung and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 5, no. 1, pp. 41–53, Feb. 2001.

[57] S.-Y. Ho, H.-S. Lin, W.-H. Liauh, and S.-J. Ho, "OPSO: Orthogonal particle swarm optimization and its application to task assignment problems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 2, pp. 288–298, Mar. 2008.

[58] W.-F. Gao, S.-Y. Liu, and L.-L. Huang, "A novel artificial bee colony algorithm based on modified search equation and orthogonal learning," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 1011–1024, Jun. 2013.

[59] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 11, pp. 6723–6742, Nov. 2021.

[60] Z.-H. Zhan *et al.*, "Cloudde: A heterogeneous differential evolution algorithm and its distributed cloud version," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 3, pp. 704–716, Mar. 2017.

[61] Z.-H. Zhan, Z.-J. Wang, H. Jin, and J. Zhang, "Adaptive distributed differential evolution," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4633–4647, Nov. 2020.

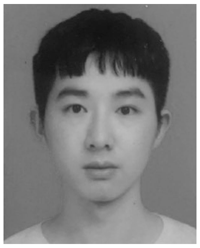
[62] S. A. Dudani, "The distance-weighted k-nearest-neighbor rule," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 4, pp. 325–327, Apr. 1976.

[63] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[64] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, Mar. 1951.

[65] B. Da *et al.*, "Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metric, and baseline results," School Comput. Sci. Eng., Nanyang Technol. Univ., Singapore, Rep., 2016. [Online]. Available: http://www.cil.ntu.edu.sg/mfo/download.html

[66] X. Xue *et al.*, "Affine transformation-enhanced multifactorial optimization for heterogeneous problems," *IEEE Trans. Cybern.*, early access, Dec. 15, 2020, doi: 10.1109/TCYB.2020.3036393.



**Sheng-Hao Wu** (Student Member, IEEE) received the B.S. degree in computer science and technology from the South China University of Technology, Guangzhou, China, in 2019, where he is currently pursuing the Ph.D. degree in computer science and technology with the School of Computer Science and Engineering.

His research interests mainly include computational intelligence, machine learning, and their applications in real-world problems and in environments of distributed computing and big data.

**Zhi-Hui Zhan** (Senior Member, IEEE) received the bachelor's and Ph.D. degrees in computer science from the Sun Yat-Sen University, Guangzhou, China, in 2007 and 2013, respectively.

He is currently the Changjiang Scholar Young Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. His current research interests include evolutionary computation, swarm intelligence, and their applications in real-world problems and in environments of cloud computing and big data.

Dr. Zhan was a recipient of the IEEE Computational Intelligence Society (CIS) Outstanding Early Career Award in 2021, the Outstanding Youth Science Foundation from the National Natural Science Foundations of China in 2018, and the Wu Wen-Jun Artificial Intelligence Excellent Youth from the Chinese Association for Artificial Intelligence in 2017. His doctoral dissertation was awarded the IEEE CIS Outstanding Ph.D. Dissertation and the China Computer Federation Outstanding Ph.D. Dissertation. He is one of the World's Top 2% Scientists for both Career-Long Impact and Year Impact in Artificial Intelligence and one of the Highly Cited Chinese Researchers in Computer Science. He is currently the Chair of the Membership Development Committee in IEEE Guangzhou Section and the Vice-Chair of IEEE CIS Guangzhou Chapter. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the *Neurocomputing*, and the *Memetic Computing*.

**Kay Chen Tan** (Fellow, IEEE) received the B.Eng. (First Class Hons.) and Ph.D. degrees from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively.

He is currently a Chair Professor of Computational Intelligence with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. He has published over 300 refereed articles and seven books.

Prof. Tan is currently the Vice-President (Publications) of IEEE Computational Intelligence Society, USA. He served as the Editor-in-Chief for the *IEEE Computational Intelligence Magazine* from 2010 to 2013 and the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION from 2015 to 2020. He currently serves as an editorial board member for more than ten journals. He is an IEEE Distinguished Lecturer Program Speaker and the Chief Co-Editor of Springer Book Series on *Machine Learning: Foundations, Methodologies, and Applications*.

**Jun Zhang** (Fellow, IEEE) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, SAR, in 2002.

He is currently a Korea Brain Pool Fellow Professor with Hanyang University, Seoul, South Korea. His current research interests include computational intelligence, cloud computing, operations research, and power electronic circuits. He has published over more than 150 IEEE Transactions papers in his research areas.

Dr. Zhang was a recipient of the Changjiang Chair Professor from the Ministry of Education, China, in 2013, The National Science Fund for Distinguished Young Scholars of China in 2011, and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON CYBERNETICS.