Block-Level Knowledge Transfer for Evolutionary Multitask Optimization

Yi Jiang, *Student Member, IEEE*, Zhi-Hui Zhan, *Senior Member, IEEE*, Kay Chen Tan, *Fellow, IEEE*, and Jun Zhang, *Fellow, IEEE*

***Abstract*—Evolutionary multitask optimization is an emerging research topic that aims to solve multiple tasks simultaneously. A general challenge in solving multitask optimization problems (MTOPs) is how to effectively transfer common knowledge between/among tasks. However, knowledge transfer in existing algorithms generally has two limitations. First, knowledge is only transferred between the aligned dimensions of different tasks rather than between similar or related dimensions. Second, the knowledge transfer among the related dimensions belonging to the same task is ignored. To overcome these two limitations, this article proposes an interesting and efficient idea that divides individuals into multiple blocks and transfers knowledge at the block-level, called the block-level knowledge transfer (BLKT) framework. BLKT divides the individuals of all the tasks into multiple blocks to obtain a block-based population, where each block corresponds to several consecutive dimensions. Similar blocks coming from either the same task or different tasks are grouped into the same cluster to evolve. In this way, BLKT enables the transfer of knowledge between similar dimensions that are originally either aligned or unaligned or belong to either the same task or different tasks, which is more rational. Extensive experiments conducted on CEC17 and CEC22 MTOP benchmarks, a new and more challenging compositive MTOP test suite, and real-world MTOPs all show that the performance of BLKT-based differential evolution (BLKT-DE) is superior to the compared state-of-the-art algorithms. In addition, another interesting finding is that the BLKT-DE is also promising in solving single-task global optimization problems, achieving competitive performance with some state-of-the-art algorithms.**

***Index Terms*—Block-level knowledge transfer (BLKT), differential evolution (DE), evolutionary computation (EC), evolutionary multitask optimization (EMTO).**

Manuscript received 1 November 2022; revised 24 January 2023 and 11 April 2023; accepted 23 April 2023. Date of publication 22 May 2023; date of current version 19 December 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2022ZD0120001; in part by the National Natural Science Foundations of China (NSFC) under Grant 62176094; 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-2022H1D3A2A01093478. This article was recommended by Associate Editor A. M. Abu-Mahfouz. *(Corresponding authors: Zhi-Hui Zhan; Jun Zhang.)*

Yi Jiang and Zhi-Hui Zhan are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: zhanapollo@163.com).

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

Jun Zhang is with the Key Laboratory of Intelligent Education Technology and Application of Zhejiang Province, Zhejiang Normal University, Jinhua 321017, China, and also with the Hanyang University (ERICA), Ansan 15588, South Korea (e-mail: csjun@scut.edu.cn.).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2023.3273625.

Digital Object Identifier 10.1109/TCYB.2023.3273625

I. Introduction

THE EVOLUTIONARY computation (EC) algorithm is a class of population-based optimization algorithms inspired by the evolution and competition of natural species [1], [2], [3]. Various kinds of EC algorithms have been proposed in the literature, mainly including the genetic algorithm [4], [5], [6], differential evolution (DE) [7], [8], [9], and particle swarm optimization [10], [11], [12]. EC algorithms have been utilized to tackle many real-world optimization problems [13], [14], [15] and have gained significant success due to their ease of implementation and superior efficiency [16], [17], [18].

Traditional EC algorithms are originally designed to solve independently existing optimization problems. However, many real-world optimization problems usually have some related or similar optimization tasks [19], [20], [21]. With this in mind, an emerging research trend of EC has been proposed, called evolutionary multitask optimization (EMTO), which aims to efficiently solve multiple tasks simultaneously by sharing common knowledge among tasks [22]. Here, each task is also an optimization problem. According to the empirical experimental results obtained by existing EMTO algorithms, it is apparent that improvements in both convergence speed and solution accuracy can be achieved by sharing knowledge among the related tasks [23].

A great challenge in solving multitask optimization problems (MTOPs) is how to effectively transfer knowledge among tasks. Many EMTO algorithms have been proposed to solve MTOPs with effective knowledge transfer strategies. We simply classify these EMTO algorithms into two categories based on their knowledge transfer strategies. The first category is called the EMTO algorithm with individual-level knowledge transfer strategy, which transfers intertask knowledge through crossover or mutation operations between individuals belonging to different tasks. Some typical EMTO algorithms with individual-level knowledge transfer strategies include the multifactorial evolutionary algorithm (MFEA) [24], MFEA with adaptive knowledge transfer (MFEA-AKT) [25], MFEA-II [26], and multifactorial DE [27]. The second category is called the EMTO algorithm with population-level knowledge transfer strategy, which usually contains multiple populations, each of which corresponds to dealing with a task. Intertask knowledge transfer is achieved by operations, such as mapping and shifting on the populations. For example, existing EMTO algorithms with population-level knowledge transfer strategies include evolutionary multitasking via explicit autoencoding (EMEA) [28], multifactorial DE with the aligned subspace continuity transfer strategy (ASCMFDE) [29], and multitasking genetic algorithm [30].

Although many EMTO algorithms with either individual-level or population-level knowledge transfer strategies have been proposed, they generally have two limitations. First, knowledge transfer at the dimension-level or block-level (i.e., a block contains several dimensions) is ignored. The knowledge is only transferred between aligned dimensions (i.e., aligned by the index) of different tasks rather than between similar or related dimensions. For example, in EMTO algorithms with individual-level knowledge transfer strategies, such as MFEA [24] and MFEA-II [26], the crossover operation can only be carried out on aligned dimensions, which means that the knowledge can only be transferred between the dimension of one task and the corresponding aligned dimensions of other tasks. However, originally aligned dimensions of two tasks only have the same index number, but perhaps without any physical meaning. Thus, they may not be similar or related, so knowledge transfer may be inefficient. Second, existing EMTO algorithms almost always consider the knowledge transfer between different tasks, but ignore the knowledge transfer among similar dimensions belonging to the same task. The dimensions coming from the same task are mutually related in many optimization problems. For example, the optimal values of some dimensions may be the same, or some dimensions may have the same monotonicity or physical meaning. Although some studies implicitly noted that transferring knowledge between unaligned dimensions of the same task can be effective [30], knowledge transfer is done between random dimensions rather than between similar dimensions.

This article proposes an interesting and effective idea, called block-level knowledge transfer (BLKT) framework to overcome the above two limitations. Different from the existing individual-level knowledge transfer and population-level knowledge transfer strategies, the BLKT is more effective and efficient with the idea of dividing each individual with a large number of dimensions into multiple small blocks and transferring knowledge at the block-level. BLKT first divides the individuals from all populations (each population corresponds to solving one task) into many blocks to obtain a block-based population (called block population). The blocks have the same length (denoted as block length) and each block contains a small number of consecutive dimensions of an individual. Then, blocks are clustered via a K-means algorithm. In this way, blocks with similar dimensions from either different tasks or the same task can be grouped together and the dimensions that are similar but originally not aligned can also be aligned in the block population. After clustering, mutation and crossover are performed on similar blocks in each cluster to enable knowledge transfer among similar dimensions belonging to either the same task or different tasks. In addition, to achieve the best performance of BLKT, we propose a feedback-based adaptive strategy (FAS) to dynamically adjust the parameters (i.e., the block length and the cluster number) in BLKT. The contributions of this article are summarized as follows.

1) An interesting and effective BLKT is proposed to address the limitations of existing individual-level knowledge transfer and population-level knowledge transfer strategies. BLKT enables common knowledge to be transferred among similar dimensions that are originally either aligned or unaligned or belong to either the same task or different tasks, which enhances the quality of the transferred knowledge.
2) FAS is proposed to find the optimal parameters for BLKT to achieve the best performance. With the help of FAS, the parameters of BLKT can be set as relatively optimal values to help BLKT obtain better performance.
3) We combine BLKT with DE to propose BLKT-based DE (BLKT-DE). Extensive experiments for BLKT-DE and several state-of-the-art EMTO algorithms are conducted on both the benchmark MTOPs and the real-world MTOPs. The benchmark MTOPs include two commonly used MTOP test suites, CEC17 [31] and CEC22 [32], and a new and challenging compositive MTOP (cMTOP) test suite [33]. Real-world MTOPs include the multitask planar kinematic arm control problems with different numbers of tasks [34], [35]. Experimental results indicate that BLKT-DE shows superior performance to the state-of-the-art algorithms on three MTOP test suites with up to 29 problems and a real-world MTOP application scenario. This shows that the idea of transferring knowledge at the block-level is effective and efficient.
4) To evaluate whether the knowledge transfer between dimensions belonging to the same task is also effective, we further use BLKT for solving single-task global optimization problems and perform experimental validation on CEC2017 [36] and CEC2022 [37] single-objective global optimization benchmarks. Interestingly, BLKT-DE with only a simple DE/rand/1 mutation operation achieves generally better or at least competitive performance to several compared state-of-the-art algorithms, even though those compared algorithms use adaptive parameter strategies and/or additional complex evolutionary operations. This shows that BLKT is a general framework for transferring knowledge that helps not only multitask optimization but also single-task optimization.

The remainder of this article is organized as follows. The description and related works of EMTO, as well as the motivation of this article, are given in Section II. The BLKT-DE is described in Section III. The experimental studies are provided in Section IV. Section V concludes this article.

II. Preliminaries

A. EMTO

Inspired by the human brain's ability to process things in parallel, EMTO aims to handle numerous optimization tasks simultaneously in a single run of the algorithm [22], [38]. Solving similar optimization tasks by sharing common knowledge can improve the algorithm's efficiency, as intrinsic correlations and similarities generally exist among real-world optimization problems. The corresponding optimization problem of EMTO is called MTOP. Specifically, the optimization goal of solving an MTOP with m minimization tasks is to find each task T_t an optimal solution x_t, which satisfies

$$x_t = \operatorname{argmin}(f_t(X_t)),\ \ t = 1, 2, \ldots, m \tag{1}$$

where $f_t(.)$ indicates the objective function of the t-th task. In MTOP, each task is an optimization problem and can be solely solved.

In the tasks of MTOP, the search spaces of different dimensions are usually different. For the convenience of knowledge transfer among different tasks, all the dimensions are normalized and encoded in the unified search space [0, 1]. Specifically, let x_t^j stands for the jth dimension of x_t in the original search space, the process of encoding x_t^j as y_t^j in the unified search space is shown as

$$y_t^j = \frac{x_t^j - L_t^j}{U_t^j - L_t^j} \tag{2}$$

where U_t^i and L_t^i stand for the upper bound and lower bound of the jth dimension in task T_t, respectively. Also, the solution x_t in the original search space can be decoded from y_t via

$$x_t^j = y_t^j \cdot \left(U_t^j - L_t^j\right) + L_t^j. \tag{3}$$

During the evolution, each individual is encoded in the unified search space $[0, 1]^D$ and is decoded into the original search space via (3) when evaluating the fitness.

B. Related Work

Since common knowledge is likely to exist between similar tasks, populations, individuals, and dimensions, transferring knowledge among similar tasks, populations, individuals, and dimensions can be more effective and is more likely to better solve the tasks. However, in many MTOPs, the tasks are usually not similar to each other. Therefore, the difficulty of transferring knowledge among these tasks is how to extract useful knowledge from relatively dissimilar tasks and use this knowledge to enhance evolution. Many knowledge transfer strategies have been proposed to transfer effective knowledge among similar populations and similar individuals. We simply classify the existing EMTO algorithms into two categories: 1) EMTO algorithms with individual-level knowledge transfer strategies and 2) EMTO algorithms with population-level knowledge transfer strategies.

Generally, in EMTO algorithms with individual-level knowledge transfer strategies, individuals of different tasks are placed in the same population and knowledge transfer across tasks is achieved via crossover and mutation operations on individuals. For instance, Gupta et al. [24] proposed the idea of multifactorial optimization and incorporated it with the evolutionary algorithm to propose MFEA. The individuals of different tasks are put into a population and knowledge is transferred between aligned dimensions by crossover. Zhou et al. [25] proposed an MFEA-AKT, which adaptively selected the best one among four different knowledge transfer strategies to effectively transfer knowledge. Bali et al. [26] extended MFEA to MFEA-II, where a matrix was kept to represent the transfer parameters and was adaptively updated to effectively transfer knowledge in each generation. Gupta et al. [39] extended MFEA for solving multiobjective MTOPs to propose the MO-MFEA algorithm. Following MO-MFEA, other individual-level knowledge transfer strategies were proposed to solve multiobjective MTOPs, such as the two-stage assortative mating-based MO-MFEA [40] and MO-MFEA-II [41]. Wang et al. [42] proposed a domain adaptation-based mapping strategy, which can reduce the difference between individuals to benefit the interindividual knowledge transfer. In solving multiobjective MTOPs, Liang et al. [43] proposed a multiobjective MFEA with self-adaptive DE, where a subspace alignment strategy was designed to map the individuals for different tasks in an aligned space to achieve effective knowledge transfer. Ji et al. [44] transformed the expensive multimodal optimization problems into an MTOP and proposed a transfer-based multitasking niche PSO, where the knowledge was transferred among particles to help find multiple optima. In addition, some studies applied the EMTO algorithms with individual-level knowledge transfer strategies in solving the challenging real-world optimization problem. For example, Feng et al. [45] applied the idea of EMTO to solve the vehicle routing problem with occasional drivers. Liu et al. [46] introduced a MO-MTO algorithm to efficiently solve the electric power dispatch problem. These existing individual-level knowledge transfer strategies are effective in transferring knowledge among individuals whose aligned dimensions are similar or contain useful knowledge.

In the EMTO algorithm with population-level knowledge transfer strategies, individuals of different tasks are placed in different populations and the intertask knowledge is transferred from one population to the other population. Feng et al. [28] first introduced the idea of explicit knowledge transfer and proposed the EMEA. In EMEA, the knowledge of one population is explicitly mapped into another population via an autoencoder. Liang et al. [47] proposed a multisource knowledge transfer strategy to transfer knowledge among populations. Zhou et al. [48] proposed a kernelized autoencoding strategy to nonlinearly transfer knowledge among the tasks. Li et al. [49] proposed a meta-knowledge transfer-based DE, where a meta-knowledge transfer strategy is designed to transfer knowledge between shifted populations. Jiang et al. [50] proposed a bi-objective knowledge transfer framework that can accurately measure intertask similarity via a bi-objective measurement and effectively transfer knowledge between similar populations. Wu et al. [51] proposed an orthogonal learning knowledge transfer strategy, which showed encouraging performance in interpopulation knowledge transfer. In another study, Wu et al. [52] proposed a transferable adaptive DE, which measured the intertask similarity via shift invariant and transferred the knowledge of successful parameters among similar tasks. For solving multiobjective MTOPs, Lin et al. [53] proposed an effective knowledge transfer strategy, which selected several solutions from the other populations based on the Euclidean distance as the transferred knowledge. Wu et al. [54] proposed a diversified knowledge transfer strategy, which aimed to expand the

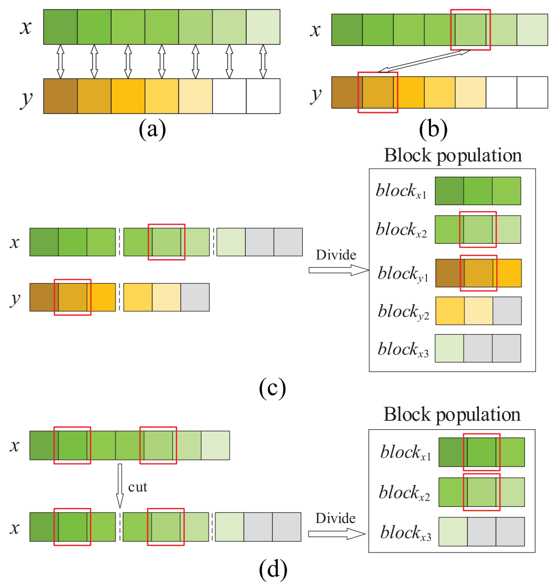


Fig. 1. Illustration of the knowledge transfer between two individuals x and y, which belong to two different tasks. (a) Knowledge can only be transferred between aligned dimensions. (b) Suppose the fifth dimension of x is similar to the second dimension of y. But they are not aligned. (c) In the block population, unaligned but similar dimensions of x and y can transfer knowledge. (d) Suppose the second and the fifth dimensions of x are similar, these two unaligned similar dimensions belonging to the same task can transfer knowledge.

transferred knowledge diversity. Additionally, several EMTO algorithms with population-level knowledge transfer strategies were proposed to solve real-world MTOPs, such as the vehicle routing problem [20] and the fuzzy system design problem [30]. The advantage of the existing population-level knowledge transfer strategies is that they can put the populations of different tasks into similar regions by shifting or mapping operations and, thus, can achieve effective knowledge transfer among these populations.

C. Motivation

In most of the existing EMTO algorithms with either individual- or population-level knowledge transfer strategies, the knowledge transfer only occurs on the aligned dimensions according to the index. However, in many real-world applications, the index-based aligned dimensions are hardly similar. Transferring knowledge among dissimilar dimensions is not effective enough, or can even cause negative knowledge transfer and slow down the search speed. This motivates us to propose BLKT to achieve effective and positive knowledge transfer among similar but unaligned dimensions.

For example, as shown in Fig. 1, there are two individuals x and y that belong to two different tasks. Individual x has seven dimensions, while individual y has five dimensions. The dimensions of x and y are colored green and yellow, respectively. The white grids indicate the padded dimensions, which make x and y have equal dimensionality. As shown in Fig. 1(a), knowledge transfer can only occur between aligned dimensions, such as the first dimension of x and the first dimension of y. However, if the fifth dimension of x is similar to the second dimension of y, as shown in Fig. 1(b), evolving these two dimensions together can better share common knowledge as knowledge transfer is more effective between similar dimensions. Because the fifth dimension of x is not originally aligned to the second dimension of y, if the knowledge transfer strategy that only transfers knowledge between aligned dimensions is applied, knowledge transfer between the fifth dimension of x and the second dimension of y can never occur.

As a result, the first motivation for proposing BLKT is to achieve knowledge transfer between unaligned dimensions to fill the research gap in existing EMTO algorithms. To effectively transfer knowledge between similar but unaligned dimensions belonging to different tasks, we divide individuals into several blocks of the same size and transfer knowledge between these blocks. As shown in Fig. 1(c), if x and y are divided into blocks of length three, then the fifth dimension of x and the second dimension of y are aligned in the block population (i.e., the second dimension of the block $block_{x2}$ is aligned with the second dimension of the block $block_{y1}$). Knowledge transfer at the block-level can be achieved via crossover and mutation of the block population. This way, knowledge is transferred between these unaligned similar dimensions of two different tasks.

The second motivation for the study is to effectively transfer knowledge between similar dimensions from the same task. In most of the current existing EC algorithms (not only the EMTO algorithms), the dimensions are optimized independently within the same task [55], [56], [57]. However, there may exist correlations between certain dimensions of the same task. For example, several similar dimensions may have the same optimal value, the same monotonicity, or similar physical meanings. By introducing additional knowledge, transferring knowledge among these similar dimensions rather than evolving them independently can enhance the efficiency of the search for the global optimum. As illustrated in Fig. 1(d), suppose the second and fifth dimensions of x are similar, these two similar dimensions are aligned in the block population by dividing x into several blocks of size three. In this way, knowledge can be transferred among similar dimensions to improve search efficiency.

III. BLKT-DE

A. General Framework

In this section, we integrate BLKT with DE to propose BLKT-DE to describe how to combine BLKT with EC algorithms to solve MTOPs. The pseudocode of BLKT-DE is given in Algorithm 1. For an MTOP with m tasks, BLKT-DE maintains m populations, each of which corresponds to solving a task.

First, each population containing n individuals is randomly initialized. The cluster size K and block length B are integers that are randomly selected in $[K_{\min}, K_{\max}]$ and $[B_{\min}, B_{\max}]$, respectively. Then, in the main loop of BLKT-DE, BLKT is executed to transfer knowledge among similar blocks to generate n offspring for each population, which is detailed in Section III-B. Next, in line 6, we adopt the DE/rand/1 mutation and crossover to evolve individuals in the same population, which also generates n offspring. After that, BLKT-DE selects

Algorithm 1: BLKT-DE for MTOP

Input: T_1, T_2,..., T_m-the MTOP with m tasks;
n-population size;
$K_{\min}$, $K_{\max}$-minimum and maximum values for cluster number K;
$B_{\min}$, $B_{\max}$ -minimum and maximum values for block length B;
Output: S_1, S_2,..., S_m-the optimal solutions for tasks T_1, $T_2, \ldots, T_m$;
Begin
1 Initialization populations $P_1, P_2, \ldots, P_m$;
2 Randomly sample K and B in $[K_{\min}, K_{\max}]$ and $[B_{\min}, B_{\max}]$;
3 **While** not termination
4 Generate n offspring of each population via BLKT as Fig. 3;
5 **For** each population P_i
6 Generate n offspring via DE/rand/1 mutation and crossover;
7 Randomly select n offspring of BLKT and DE;
8 Evaluate the fitness of the selected n offspring;
9 Select n fitter solutions from n parents and n offspring;
10 **End For**
11 **If** none of the best solutions in all populations are updated
12 Randomly sample K and B in $[K_{\min}, K_{\max}]$ and $[B_{\min}, B_{\max}]$;
13 **Else**
14 Randomly sample K and B in $[K$-1, K+1] and $[B$-1, B+1];
15 **End If**
16 **End While**
17 Set S_i as the optimal solution of T_i, where $i = 1, 2, \ldots, m$;
End

n offspring randomly from the $2n$ offspring that are generated by BLKT and DE. Then, the selected n offspring are evaluated. Selection is carried out on these n evaluated offspring and the n parents in the current population to choose the fitter n individuals as the population in the next generation (i.e., line 9). Finally, in lines 11–15, we design a simple but effective parameter adjustment strategy, called FAS, to dynamically and adaptively control K and B. The idea of FAS can be briefly introduced as follows: if the current settings for K and B cannot improve the best solution in any population, then we randomly reinitialize them since the current parameters are far from optimal; otherwise, if the current settings for K and B can improve at least one of the best solutions, then we just add a slight change to the values of K and B, which randomly samples K and B from [K–1, K+1] and [B–1, B+1], respectively. The flowchart of BLKT-DE for solving MTOPs with two tasks is illustrated in Fig. 2. Note that, we color the offspring generated by BLKT in pink and the offspring generated by DE in yellow for clarity.

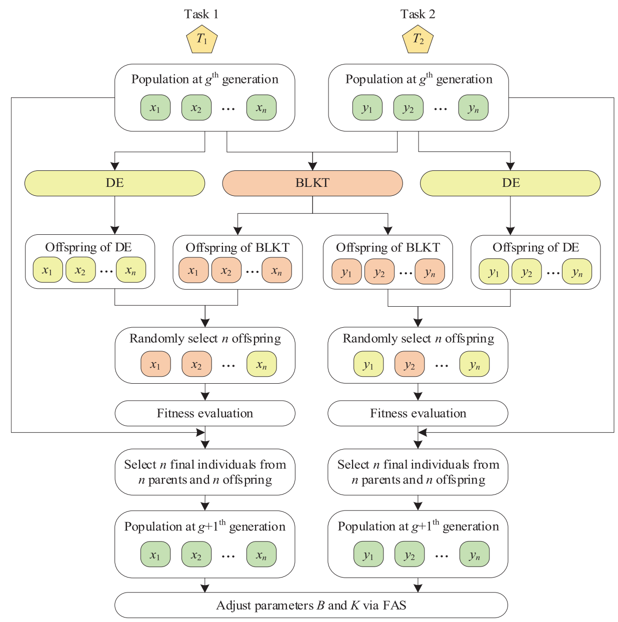


Fig. 2. Flowchart of BLKT-DE for solving MTOPs with two tasks.

B. BLKT

To describe BLKT more intuitively, the process of BLKT for solving MTOPs with two tasks is illustrated in Fig. 3. The individuals of two populations P_1 and P_2 corresponding to the two tasks are colored green and yellow and have seven and five dimensions, respectively. The BLKT process includes five steps: 1) divide; 2) get block population; 3) K-means clustering; 4) mutation and crossover; and 5) reconstruct.

In the first step, the individuals in P_1 and P_2 are divided into several blocks with lengths of B. Note that the block length B is an integer that is dynamically adjusted in the value range $[B_{\min}, B_{\max}]$ through FAS, which is introduced later in Section III-C. Herein, for convenience, the value of B is set as three in the example of Fig. 3. As neither seven nor five is a multiple of three, the last block of each individual is padded with zeros (i.e., dimensions colored in white) to make its length equal to three. Generally, if the individual's last block has a length less than B, this block will be padded with zeros to make its length equal to B. In total, w blocks will be formed with

$$w = \sum_{t=1}^{m} \left\lceil \frac{D_t}{B} \right\rceil \cdot n \tag{4}$$

where m indicates the number of tasks, D_t is the number of dimensions of the t-th task, and n is the population size.

Then, in the second step, all the blocks from either P_1 or P_2 are gathered together to form a block population. Dividing the individuals and gathering the blocks into a block population can benefit knowledge transfer from two aspects. First, enabling unaligned dimensions that come from different tasks to share common knowledge. As illustrated in Fig. 3, although the two blocks $block_2$ and $block_{w-2}$ are from two different tasks and are originally unaligned, they can be aligned in the block population and knowledge can be transferred between them via mutation and crossover operations. Second, we enable unaligned dimensions that belong to the same tasks to

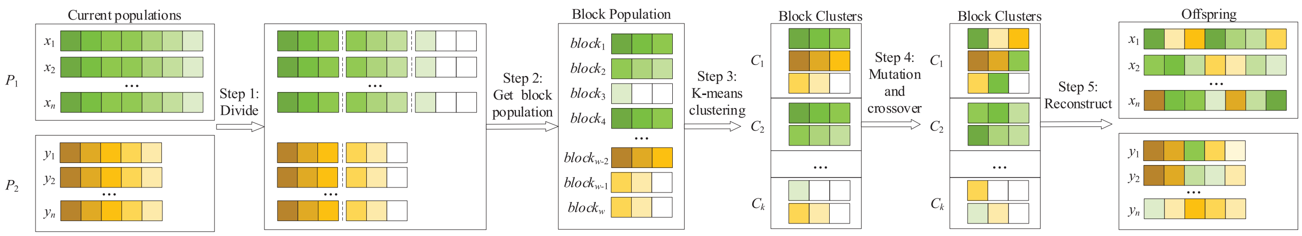


Fig. 3. BLKT process for solving MTOPs with two tasks.

share common knowledge. For example, although $block_1$ and $block_2$ correspond to different dimensions of the same task, the knowledge can be transferred between them via mutation and crossover operations. As stated in Section II-C, since the dimensions belonging to the same task may have intrinsic correlations, sharing knowledge between these dimensions can help efficiently locate the global optimum.

In the third step, K-means clustering is executed to cluster similar blocks together, as the knowledge transfer is more efficient on similar blocks. In Fig. 3, the notation C_k indicates the kth block cluster. The advantage of dividing the block population into several clusters can be described as follows. First, not only similar blocks from different tasks but also those from the same task can be put into the same cluster to achieve positive knowledge transfer. Second, the blocks that are not similar are placed in different clusters, which helps to avoid the negative knowledge transfer among irrelated blocks. In the K-means algorithm, the cluster number K is adaptively selected from the interval $[K_{\min}, K_{\max}]$ via FAS.

Then, in the fourth step, mutation and crossover operations are performed in each cluster to effectively transfer knowledge among similar blocks. Specifically, the DE/rand/1 mutation operation and binomial crossover operation [58], [59] are adopted to generate block offspring of $block_i$ ($1 \leq i \leq w$), which are, respectively, shown as

$$v_i = block_{r1} + F \times (block_{r2} - block_{r3}) \tag{5}$$

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if rand} < CR \text{ or } j = jrand \\ block_{i,j}, & \text{otherwise} \end{cases} \tag{6}$$

where $r1$, $r2$, and $r3$ are indexes of three mutually exclusive blocks randomly selected from the cluster of $block_i$. F and CR are the amplifier factor and the crossover rate, respectively. $jrand$ is a randomly selected dimension. Note that, the above mutation and crossover operations are not carried out if the size of the cluster is smaller than three.

After the mutation and crossover operations, in the fifth step, offspring blocks are put into the corresponding dimensions to reconstruct offspring individuals. Offspring block u_1 serves as an example, its parental block $block_1$ corresponds to the first three dimensions of individual x_1, and, thus, the offspring block u_1 is used to replace the first three dimensions of individual x_1. As seen in Fig. 3, some dimensions of the offspring are colored in different colors from those in the original population, which means the knowledge is transferred among similar dimensions of either the same task or different tasks.

C. FAS

In the BLKT, two parameters, that is, block length B and cluster number K, are introduced, whose values can influence the performance of BLKT. If B and K are both set as the optimal values, BLKT will show better performance. However, the optimal settings of B and K depend on the population distribution of each task, which is dynamically changed in every generation. In other words, the optimal settings of B and K are different in different generations.

To obtain the optimal settings of B and K in every generation, FAS is proposed to adaptively control the parameters according to the feedback of performance. The basic idea of FAS is simple yet effective: if the current settings of B and K show relatively good performance, that is, achieving improvement on at least one task, then it indicates that the optimal settings of B and K in the next generation may be close to those in the current generation. In this case, a slight adjustment is performed on K and B, which is shown as

$$\begin{cases} B = \text{randInt}([B-1, B+1]) \\ K = \text{randInt}([K-1, K+1]) \end{cases} \tag{7}$$

where the function randInt(.) generates a random integer from a given interval.

Otherwise, if the current settings of B and K show relatively worse performance, that is, getting nonimprovement on all the tasks, the optimal settings of B and K in the next generation will be far away from those in the current generation. Therefore, we reinitialize the two parameters in their value ranges as follows:

$$\begin{cases} B = \text{randInt}([B_{\min}, B_{\max}]) \\ K = \text{randInt}([K_{\min}, K_{\max}]). \end{cases} \tag{8}$$

Note that the initial values of B and K are also sampled via (8).

IV. Experimental Studies

To evaluate the effectiveness and efficiency of the proposed BLKT, five experiments are designed in this section. First, we compare the performance of BLKT-DE and several state-of-the-art EMTO algorithms on the widely used CEC17 and CEC22 MTOP benchmarks. Second, we conduct a comparison on a novel and more challenging MTOP benchmark to further show BLKT-DE's effectiveness and efficiency. Third, we evaluate the performance of BLKT-DE with different value ranges of the parameters B and K, give the optimal settings of the value ranges and analyze the effect of the FAS. Fourth, we design a comparison of BLKT-DE and the BLKT-DE variant without K-means to evaluate the effect of the K-means algorithm and to validate that knowledge transfer among similar dimensions is more effective than knowledge transfer among dissimilar dimensions. Fifth, we conduct experiments on the multitask planar kinematic arm control problems

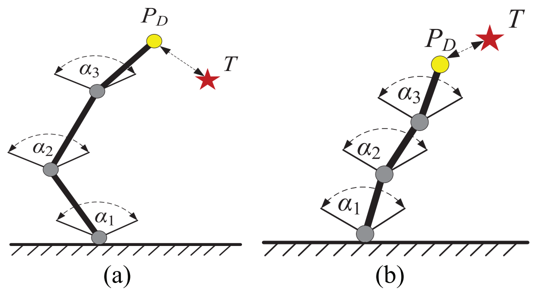


Fig. 4. Illustration of a multitask planar kinematic arm control problem with two tasks. (a) Task 1. (b) Task 2.

with different numbers of tasks to show the effectiveness and efficiency of BLKT-DE in solving real-world MTOPs. Sixth, to show the effects of knowledge transfer among similar dimensions belonging to the same task, we use BLKT-DE to solve the single-task global optimization problems in the CEC17 and CEC22 single-objective global optimization benchmarks and compare the performance of BLKT-DE and several state-of-the-art DE-based algorithms.

A. Benchmark Functions

Two MTOP Benchmarks, CEC17M and CEC22M: First, the experiments are conducted on the commonly used CEC17 and CEC22 MTOP test suites. For convenience, we use CEC17M and CEC22M to denote the CEC17 MTOP and the CEC22 MTOP benchmarks, respectively. In CEC17M, there are nine MTOP instances. In CEC22M, there are ten complex MTOP instances. The detailed properties of CEC17M and CEC22M can be found in [31] and [32].

A New and More Challenging MTOP Benchmark, cMTOP: We conduct experiments on the more challenging cMTOP benchmark [33]. There are ten instances in cMTOP, each of which contains two tasks. Each task is a compositive function composed of at least one basic function. Solving cMTOP is more difficult. First, the global optima of the tasks are not all zeros and the optimal values of any two aligned dimensions are not the same. Second, each two aligned dimensions in cMTOP have irrelated physical meanings as these dimensions are from different basic functions. Third, the two tasks in each cMTOP instance have different dimensions. To solve the cMTOP, the algorithm is required to have a smart ability to transfer knowledge between similar dimensions rather than only between aligned dimensions.

Real-World MTOPs: Third, comparisons are made on several multitask planar kinematic arm control problems [34], [35] with different numbers of tasks to assess the performance of BLKT-DE in solving real-world MTOPs. Fig. 4 shows an example of a two-task planar kinematic arm control problem. Here, the objective of each task is to find a set of optimal angles (i.e., $\alpha_1, \alpha_2, \ldots, \alpha_d$) of all joints to minimize Euclidean distance between the tip of the arm (i.e., P_D) and the target (i.e., T). The objective function of the t-th task is shown as

$$f_t\left(\alpha_1, \alpha_2, \ldots, \alpha_d, \left[L^t, \alpha^t_{\max}\right]\right) = \|P_D - T\| \tag{9}$$

where L^t and $\alpha^t{}_{\max}$ denote the total length of the arms and the maximum range of the angles, respectively. As in [34] and [35], multiple different tasks are created by taking different values of L^t and $\alpha^t{}_{\max}$. The position of the target is set as [0.5, 0.5]. Five multitask planar kinematic arm control problems with different numbers of tasks are created (i.e., 2-tasks, 5-tasks, 10-tasks, 50-tasks, and 100-tasks). The number of dimensions (i.e., the number of joints) of each task is set as 20.

Two Single-Task Global Optimization Benchmarks, CEC17S and CEC22S: To evaluate the performance of BLKT-DE in solving single-task global optimization problems, the CEC17 and CEC22 single-objective global optimization benchmarks, which are denoted as CEC17S and CEC22S, respectively, are used in comparison. There are 29 global optimization problems in CEC17S, and 12 global optimization problems in CEC22S. The properties of CEC17S and CEC22S can be found in [36] and [37], respectively.

B. Compared Algorithms and Parameter Settings

First, for the experiments conducted on MTOP benchmarks (i.e., CEC17M and cMTOP), the proposed BLKT-DE is compared with four state-of-the-art EMTO algorithms: 1) MFEA-AKT [25]; 2) MFEA-II [26]; 3) EMEA [28]; and 4) ASCMFDE [29]. The parameter settings of BLKT-DE and the compared algorithms are summarized as follows.

1) Simulated binary crossover [60] and polynomial mutation [61] in MFEA-AKT, MFEA-II, and EMEA: $\eta_c = 2$, $\eta_m = 5$.
2) *F and CR in DE of EMEA, ASCMFDE, and BLKT-DE:* $F = 0.5$, $CR = 0.7$. Note that the F and CR in (5) and (6) also adopt these two values.
3) *Value Ranges of B and K in BLKT-DE:* $[B_{\min}, B_{\max}] = [1, \min(D_1, D_2)]$, $[K_{\min}, K_{\max}] = [2, n/2]$, where $\min(D_1, D_2)$ indicates the smaller one of the dimension number of the first task D_1 and the dimension number of the second task D_2.
4) *Population Size:* $n = 100$ for EMEA, ASCMFDE, and BLKT-DE, and $n = 200$ for MFEA-AKT and MFEA-II.
5) *Maximum Function Evaluations:* $MaxFEs = 200\,000$.
6) The parameters whose values are not given are set as the optimal settings given in the corresponding papers.

Second, for the experiments conducted on the single-task global optimization benchmarks (i.e., CEC17S and CEC22S), BLKT-DE is compared with four state-of-the-art DE-based algorithms for single-task global optimization: 1) JADE [62]; 2) hybrid-adaptive DE with a decay function (HyDE-DF) [63]; 3) DE with difference vector reuse (DE-DVR) [64]; and 4) adaptive distributed DE (ADDE) [65]. The parameter settings of BLKT-DE and the compared algorithms for solving the CEC17S and CEC22S problems are shown as follows.

1) *F and CR in DE:* $F = 0.5$, $CR = 0.9$.
2) *Value Ranges of B and K in BLKT-DE:* $[B_{\min}, B_{\max}] = [1, D]$, $[K_{\min}, K_{\max}] = [2, n/2]$.
3) *Population Size:* $n = 100$.
4) *Maximum Function Evaluations:* $MaxFEs = 10000 \times D$, where D is the number of dimensions.
5) *Number of Dimensions:* $D = 50$ for CEC17S and $D = 20$ for CEC22S.

TABLE I
EXPERIMENTAL RESULTS OBTAINED BY BLKT-DE AND THE COMPARED EMTO ALGORITHMS ON CEC17M AND CEC22M

Problem	BLKT-DE		MFEA-AKT		MFEA-II		EMEA		ASCMFDE	
	Task 1	Task 2	Task 1	Task 2	Task 1	Task 2	Task 1	Task 2	Task 1	Task 2
CEC17M-P1	**7.23E-08**	7.31E+01	1.97E-01 (+)	1.70E+02 (+)	4.89E-03 (+)	1.46E+01 (+)	2.04E-01 (+)	4.11E+02 (+)	1.28E-03 (+)	**1.41E+00 (−)**
CEC17M-P2	**3.21E-05**	6.59E+01	3.87E+00 (+)	1.71E+02 (+)	9.63E-01 (+)	5.48E+01 (≈)	3.44E+00 (+)	4.12E+02 (+)	2.45E-03 (+)	**7.05E-02 (−)**
CEC17M-P3	2.12E+01	2.15E+03	2.01E+01 (−)	2.95E+03 (+)	**1.27E+01 (−)**	**5.23E+02 (−)**	2.11E+01 (≈)	1.12E+04 (+)	2.12E+01 (≈)	1.29E+04 (+)
CEC17M-P4	**7.42E+01**	**8.76E-08**	4.31E+02 (+)	3.01E+00 (+)	8.58E+01 (≈)	7.11E-03 (+)	3.37E+02 (+)	2.62E-04 (+)	3.86E+02 (+)	8.23E-04 (+)
CEC17M-P5	**3.49E-05**	**4.86E+01**	2.71E+00 (+)	2.52E+02 (+)	6.63E-01 (+)	1.12E+02 (+)	3.48E+00 (+)	5.71E+01 (+)	1.35E-02 (+)	4.97E+01 (+)
CEC17M-P6	**2.75E-05**	1.65E+00	3.55E+00 (+)	3.97E+00 (+)	8.10E-01 (+)	1.10E+00 (≈)	3.49E+00 (+)	9.70E-01 (≈)	8.64E-01 (+)	**2.16E-01 (≈)**
CEC17M-P7	**4.75E+01**	7.14E+01	3.46E+02 (+)	2.22E+02 (+)	1.34E+02 (+)	7.80E+01 (≈)	1.29E+03 (+)	4.15E+02 (+)	4.75E+01 (+)	**2.22E+00 (−)**
CEC17M-P8	**7.18E-06**	**2.90E-01**	2.26E-01 (+)	1.88E+01 (+)	7.40E-03 (+)	7.52E+00 (+)	2.16E-01 (+)	7.08E+00 (+)	2.35E-03 (+)	5.22E-01 (+)
CEC17M-P9	**7.07E+01**	1.98E+03	4.47E+02 (+)	3.19E+03 (+)	9.71E+01 (≈)	**6.39E+02 (−)**	3.49E+02 (+)	7.45E+03 (+)	4.19E+02 (+)	1.88E+03 (≈)
CEC22M-P1	6.05E+02	6.04E+02	6.23E+02 (+)	6.23E+02 (+)	6.15E+02 (+)	6.14E+02 (+)	6.38E+02 (+)	6.07E+02 (+)	**6.01E+02 (−)**	**6.01E+02 (−)**
CEC22M-P2	**7.00E+02**	**7.00E+02**	7.01E+02 (+)	7.01E+02 (+)	7.00E+02 (+)	7.00E+02 (+)	7.01E+02 (+)	7.00E+02 (+)	7.00E+02 (+)	7.00E+02 (+)
CEC22M-P3	1.24E+06	1.15E+06	**6.30E+05 (−)**	**7.05E+05 (−)**	1.30E+06 (≈)	1.31E+06 (≈)	1.71E+06 (≈)	4.19E+07 (+)	3.87E+07 (+)	4.15E+07 (+)
CEC22M-P4	1.30E+03	1.30E+03	1.30E+03 (+)	1.30E+03 (−)	**1.30E+03 (−)**	**1.30E+03 (−)**	1.30E+03 (≈)	1.30E+03 (+)	1.30E+03 (+)	1.30E+03 (+)
CEC22M-P5	1.53E+03	1.53E+03	1.54E+03 (+)	1.54E+03 (+)	**1.51E+03 (−)**	**1.51E+03 (−)**	1.53E+03 (−)	1.53E+03 (+)	1.53E+03 (+)	1.54E+03 (+)
CEC22M-P6	**8.02E+05**	**7.24E+05**	1.01E+06 (≈)	9.83E+05 (≈)	8.66E+05 (≈)	7.34E+05 (≈)	1.28E+06 (+)	1.56E+07 (+)	1.71E+07 (+)	1.59E+07 (+)
CEC22M-P7	**2.86E+03**	**2.95E+03**	2.93E+03 (≈)	3.06E+03 (≈)	3.11E+03 (≈)	3.18E+03 (≈)	2.95E+03 (≈)	4.28E+03 (+)	4.29E+03 (+)	4.24E+03 (+)
CEC22M-P8	5.21E+02	5.21E+02	**5.20E+02 (−)**	**5.20E+02 (−)**	5.21E+02 (≈)	5.21E+02 (+)	5.21E+02 (≈)	5.21E+02 (≈)	5.21E+02 (≈)	5.21E+02 (≈)
CEC22M-P9	7.65E+03	**1.62E+03**	7.96E+03 (≈)	1.62E+03 (+)	7.89E+03 (≈)	1.62E+03 (+)	**7.18E+03 (−)**	1.62E+03 (+)	1.47E+04 (+)	1.62E+03 (+)
CEC22M-P10	1.90E+04	**8.59E+05**	**1.74E+04 (≈)**	2.07E+06 (+)	2.31E+04 (≈)	1.30E+06 (+)	2.54E+04 (+)	1.75E+07 (+)	5.26E+04 (+)	1.77E+07 (+)
Number of + / ≈ / −			12 / 4 / 3	14 / 2 / 3	8 / 8 / 3	9 / 6 / 4	12 / 5 / 2	17 / 2 / 0	16 / 2 / 1	12 / 3 / 4

6) The remaining parameters are set as the same as those in the corresponding papers.

Each algorithm is executed for 30 independent runs to obtain the experimental results. To evaluate the experimental results from the statistical view, Wilcoxon's rank-sum test [66] at $\alpha = 0.05$ is adopted. The notations "+ / ≈ / −" indicate that the results obtained by BLKT-DE are "significantly superior/equal/significantly inferior" to those obtained by the compared algorithm based on Wilcoxon's rank-sum test.

C. Comparison on CEC17M and CEC22M

The experimental results for mean fitness obtained by BLKT-DE, MFEA-AKT, MFEA-II, EMEA, and ASCMFDE on CEC17M and CEC22M are listed in Table I. The best experimental result is marked in **boldface** for each task. According to the results of Wilcoxon's rank-sum test, on the CEC17M and CEC22M benchmarks, BLKT-DE generally outperforms the compared state-of-the-art algorithms. For the 19 total MTOPs, for task 1, BLKT-DE outperforms MFEA-AKT, MFEA-II, EMEA, and ASCMFDE on 12, 8, 12, and 16 tasks, respectively, and is only worse on 3, 3, 2, and 1 tasks, respectively. For task 2, BLKT-DE outperforms MFEA-AKT, MFEA-II, EMEA, and ASCMFDE on 14, 9, 17, and 12 tasks, respectively. It indicates that when dealing with MTOPs where similar dimensions are not aligned, BLKT-DE performs significantly better than these compared algorithms.

In addition, to study the convergence behavior of BLKT-DE and the compared algorithms, their convergence curves on two representative problems of CEC17M and CEC22M are illustrated in Fig. 5, which are CEC17M-P8 and CEC22M-P6. First, as shown in Fig. 5(a), in both task 1 and task 2 of CEC17M-P8, the convergence speed of BLKT-DE is relatively faster than that of other algorithms. Second, as shown in Fig. 5(b), in both task 1 and task 2 of CEC22M-P6 although the convergence speed of BLKT-DE is slightly slower than that of MFEA-II and MFEA-AKT in the previous generations,

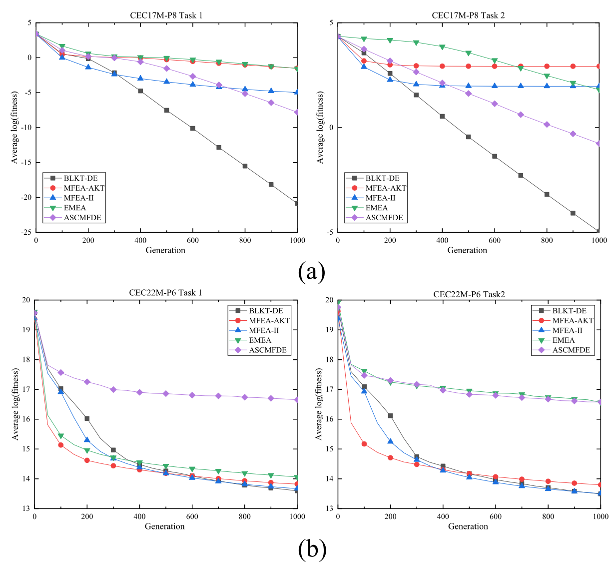


Fig. 5. Convergence curves obtained by BLKT-DE and compared algorithms on two representative MTOPs in CEC17M. (a) CEC17M-P8. (b) CEC22M-P6.

BLKT-DE obtains better final results than all the compared algorithms.

D. Comparison on cMTOP

The experimental results obtained by BLKT-DE, MFEA-AKT, MFEA-II, EMEA, and ASCMFDE on the ten MTOPs of cMTOP are listed in Table II. In cMTOP, aligned dimensions are generally dissimilar, so the MTOPs of cMTOP are relatively challenging. Therefore, it can be observed that the knowledge transfer of the compared algorithm on most of the problems is less effective than that on CEC17M. On cMTOP-P1, for example, ASCMFDE performs well in task 1 but performs poorly in task 2. This is because in cMTOP, similar dimensions are more likely to be unaligned, and the knowledge transfer strategies of the compared algorithms

TABLE II
EXPERIMENTAL RESULTS OBTAINED BY BLKT-DE AND THE COMPARED EMTO ALGORITHMS ON cMTOP

Problem	BLKT-DE		MFEA-AKT		MFEA-II		EMEA		ASCMFDE	
	Task 1	Task 2	Task 1	Task 2	Task 1	Task 2	Task 1	Task 2	Task 1	Task 2
cMTOP-P1	2.93E-17	**1.47E-14**	5.23E-03 (+)	1.40E+02 (+)	2.91E-04 (+)	4.97E+01 (+)	6.30E-04 (+)	2.12E+02 (+)	**3.47E-23** (−)	2.31E+02 (+)
cMTOP-P2	**0.00E+00**	1.06E-01	1.87E+01 (+)	1.28E+00 (+)	9.61E+00 (+)	2.91E-01 (+)	1.42E+01 (+)	1.86E-02 (−)	4.64E+00 (+)	**1.51E-02** (−)
cMTOP-P3	**6.35E-06**	**1.28E+02**	1.12E-01 (+)	1.36E+03 (+)	2.50E-03 (+)	1.32E+02 (+)	8.62E-02 (+)	8.44E+02 (+)	1.06E-04 (+)	1.68E+02 (+)
cMTOP-P4	**7.99E-15**	1.97E+02	1.19E+00 (+)	1.43E+02 (−)	3.74E-01 (+)	**4.92E+01** (−)	5.55E-02 (+)	1.76E+02 (−)	7.99E-15 (≈)	1.89E+02 (≈)
cMTOP-P5	**1.49E+01**	1.54E+03	2.18E+02 (+)	2.18E+03 (+)	1.35E+02 (+)	**4.38E+02** (−)	9.34E+01 (+)	7.33E+03 (+)	1.60E+01 (+)	6.95E+03 (+)
cMTOP-P6	**1.10E+03**	1.58E+00	2.63E+03 (+)	8.25E+00 (+)	2.88E+03 (+)	1.78E+00 (+)	3.11E+03 (+)	3.45E+00 (+)	1.33E+03 (≈)	**6.37E-01** (−)
cMTOP-P7	**8.34E+02**	**4.27E+01**	1.68E+03 (+)	1.32E+02 (+)	8.74E+02 (≈)	5.16E+01 (+)	1.68E+03 (+)	1.56E+02 (+)	2.29E+03 (+)	1.24E+02 (+)
cMTOP-P8	**5.10E-04**	**2.63E+01**	4.38E+00 (+)	7.16E+01 (+)	5.00E+00 (+)	3.11E+01 (≈)	1.94E+01 (+)	1.25E+02 (+)	1.23E-01 (≈)	1.07E+02 (+)
cMTOP-P9	**7.50E+00**	**2.77E+01**	3.71E+01 (+)	8.96E+01 (+)	2.51E+01 (+)	3.08E+01 (≈)	7.82E+01 (+)	1.34E+02 (+)	7.54E+00 (+)	8.12E+01 (+)
cMTOP-P10	**6.52E+00**	**1.11E+02**	4.95E+01 (+)	6.85E+02 (+)	1.45E+02 (+)	1.67E+02 (+)	1.03E+02 (+)	1.85E+02 (+)	8.34E+00 (+)	1.70E+02 (≈)
Number of + / ≈ / −			10 / 0 / 0	9 / 0 / 1	9 / 1 / 0	6 / 2 / 2	10 / 0 / 0	8 / 0 / 2	6 / 3 / 1	6 / 2 / 2

are insufficiently effective. Unlike the compared algorithms, BLKT-DE shows good performance in cMTOP. Considering the results of Wilcoxon's rank-sum test, BLKT-DE outperforms MFEA-AKT, MFEA-II, EMEA, and ASCMFDE on 19, 15, 18, and 12 tasks out of the 20 tasks, respectively.

In addition, the convergence curves obtained by BLKT-DE and the compared algorithms on four representative problems are illustrated in Fig. 6. The four representative problems are cMTOP-P1, cMTOP-P3, cMTOP-P6, and cMTOP-P10. The convergence trends of BLKT-DE on the two tasks of each problem in cMTOP-P1, cMTOP-P3, and cMTOP-P10 are similar, which indicates that effective knowledge transfer in BLKT-DE helps to solve both tasks simultaneously and efficiently. However, the compared algorithms may become trapped in the local optima and their convergence curves tend to stagnate. In cMTOP-P1 task 2, for example, whereas ASCMFDE exhibits outstanding convergence behavior in task 1, its convergence curve tends to stagnate in task 2. Between the 300th and 600th generations, BLKT-DE tends to jump out of the local optima and produces the best final results in cMTOP-P8 task 2.

We can draw three conclusions based on the analysis above. First, the accuracy of BLKT-DE's solutions on all tasks is generally superior to that of the state-of-the-art compared algorithms for most MTOPs, notably on challenging MTOPs. Second, BLKT-DE's intertask knowledge transfer is effective, allowing the algorithm to perform well in the tasks of the cMTOP. Third, BLKT-DE can obtain extra knowledge due to knowledge transfer between similar dimensions and hence is more likely to jump out of the local optima.

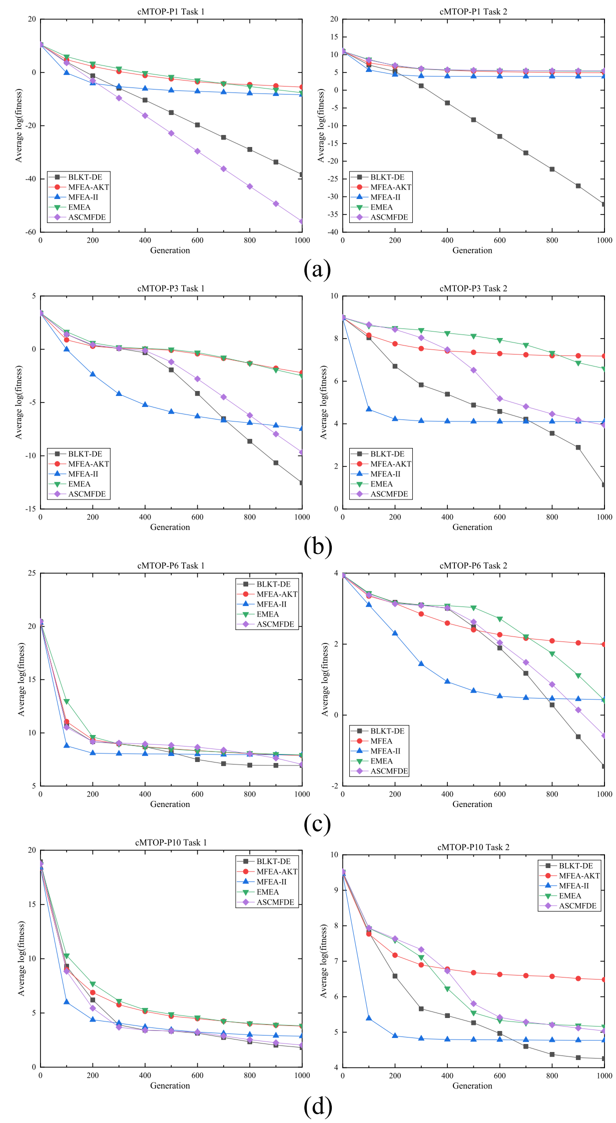


Fig. 6. Convergence curves obtained by BLKT-DE and compared algorithms on four representative MTOPs in cMTOP. (a) cMTOP-P1. (b) cMTOP-P3. (c) cMTOP-P6. (d) cMTOP-P10.

E. Analysis of FAS

1) Parameter Analysis: In BLKT-DE, the two parameters, block length B and cluster size K, are controlled via FAS in the value ranges $[B_{\min}, B_{\max}]$ and $[K_{\min}, K_{\max}]$, respectively. The two lower bounds $B_{\min}$ and $K_{\min}$ are set to 1 and 2, respectively, since the length of each block is at least one and the number of block subpopulations is at least two. In the above experiments, the two upper bounds $B_{\max}$ and $K_{\max}$ are manually set to $\min(D_1, D_2)$ and $n/2$, respectively, where D_1, D_2, and n denote the number of dimensions of the first task, the number of dimensions of the second task, and the population size, respectively. However, first, these two manually set parameters may greatly affect the performance of BLKT-DE, in other words, BLKT-DE may be sensitive to these two parameter settings. Second, to bring out the best performance of BLKT-DE, we also need to determine the optimal settings of $B_{\max}$ and $K_{\max}$. Therefore, it is necessary to test the performance of BLKT-DE under different parameter settings. For the convenience of description, two additional parameters are introduced, here, to replace $B_{\max}$ and $K_{\max}$, which are named H_B and H_K, respectively. $B_{\max}$ and $K_{\max}$ are calculated via $B_{\max} = \min(D_1, D_2)/H_B$ and $K_{\max} = n/H_K$, respectively. In this way, the problem of finding the optimal settings of $B_{\max}$ and $K_{\max}$ can be transformed into the problem of finding the optimal settings of H_B and H_K.

Nine BLKT-DE variants with different combinations of H_B and H_K are designed according to three different H_B settings (i.e., H_B= 1, 2, and 4) and three different H_K settings (i.e., H_K= 1, 2, and 4). We record the performance of the nine BLKT-DE variants on all 10 cMTOPs in Table S.I of the supplemental material due to space limitation. The number of best (denoted as NoB) and the mean rank (denoted as MR) are presented. The experimental results for either NoB or MR show that BLKT-DE with $[H_B, H_K] =$ [1], [2] can achieve the best performance. Therefore, H_B and H_K are set as 1 and 2, respectively, which indicates $B_{\max}$ and $K_{\max}$ are recommended to be set as min(D_1, D_2) and n/2, respectively.

2) Effect of FAS: FAS enables the values of K and B to be adaptively adjusted according to their performance. To analyze the effect of FAS, several BLKT-DE variants with fixed values of K and B are designed and compared with BLKT-DE. These twelve BLKT-DE variants are combined by taking $K = 1$, 2, or 5 and $B = 2$, 5, 10, and 20.

The numbers of $+$ / $\approx$ / $-$ obtained by comparing BLKT-DE and these BLKT-DE variants with fixed K and B on task 1 and task 2 of cMTOP are shown in Tables S.II and S.III of the supplemental material, respectively. According to the results, BLKT-DE with FAS generally outperforms these BLKT-DE variants with fixed K and B. Specifically, the results of BLKT-DE are significantly better than those of the BLKT-DE variants in more than eight tasks over the total ten tasks on either task 1 or task 2. These experimental results indicate that FAS can obtain the relatively optimal values of K and B. Therefore, the FAS is necessary for obtaining the optimal performance of the BLKT framework.

F. Discussion on BLKT

This section discusses when and how using BLKT is effective. The effects of BLKT can be theoretically analyzed according to the genetic schema theorem [67], [68]. In MTOPs, promising schemas of different tasks and the same task can have several similar dimensions. On the one hand, the block in BLKT seems similar to the building block in the building block hypothesis of the genetic schema theorem. This is because the optimal solution is composed of several promising building blocks according to the building block hypothesis and the BLKT is executed on the blocks. On the other hand, different from the building block hypothesis, the blocks in BLKT are transferred from different tasks or even from unaligned dimensions. Therefore, we will further discuss the effects of BLKT to investigate when and how to transfer blocks that are effective and efficient.

1) Discussion on When to Transfer Blocks: It is important to further analyze when the transfer of the blocks among tasks would improve the quality of solutions. In the standard BLKT-DE, knowledge is transferred among tasks via BLKT in every generation. Suppose the BLKT framework is executed in every few generation gaps, and the population can only obtain some knowledge from other tasks. If the population converges to a local optimum, it is hard to jump out of the local optimum without the knowledge of the other tasks as the converged population cannot obtain sufficient knowledge. Therefore, it is intuitive to transfer knowledge in every generation via the BLKT framework.

In addition, to analyze when to transfer blocks via experiments, we compare the standard BLKT-DE with four BLKT-DE variants. In these BLKT-DE variants, BLKT is executed in several generation gaps to transfer intertask knowledge. These four BLKT-DE variants include BLKT-DE-G2, BLKT-DE-G5, BLKT-DE-G10, and BLKT-DE-G20, whose generation gaps are 2, 5, 10, and 20, respectively. By comparing the performance of BLKT-DE and these variants, we can determine when to transfer blocks is more effective.

The experimental results obtained by BLKT-DE and four variants on the cMTOP test suite are shown in Table S.IV of the supplementary material. According to the results, we can find that the standard BLKT-DE generally outperforms these variants. In most cMTOP tasks, BLKT-DE achieves the best performance. In addition, the convergence curves on cMTOP-P3 and cMTOP-P7 are shown in Fig. S1 of the supplementary material. Another observation that can be found according to the results of the table and figure is that the performance of these BLKT-DE variants gradually decreases as the generation gaps increase. Therefore, we can conclude that executing BLKT for knowledge transfer in every generation is the optimal strategy.

2) Discussion on How to Transfer Blocks: Analyzing how the transfer of the blocks among tasks would improve the quality of solutions has two important parts. The first is to analyze whether knowledge transfer among similar dimensions is more effective than knowledge transfer among dissimilar dimensions. The second is to analyze whether the intertask knowledge transfer is more effective.

First, to analyze whether knowledge transfer among similar dimensions is more effective than knowledge transfer among dissimilar dimensions, we design a BLKT-DE variant without the K-means algorithm, namely BLKT-DE-w/o-km. In BLKT, similar blocks are clustered together via the K-means algorithm. If the K-means is eliminated the knowledge can be transferred among either similar or dissimilar dimensions.

The comparison between BLKT-DE and BLKT-DE-w/o-km is conducted on both CEC17M and cMTOP. The summarized results for the number of tasks where the BLKT-DE is "superior/equal/inferior" to the BLKT-DE-w/o-km are provided in Table S.V of the supplemental material. In the results, BLKT-DE generally outperforms BLKT-DE-w/o-km in CEC17M and cMTOP. Specifically, BLKT-DE outperforms BLKT-DE-w/o-km on 17 tasks on CEC17M, while BLKT-DE outperforms BLKT-DE-w/o-km on 18 tasks on cMTOP. Additionally, we plot the convergence curves on CEC17M-P5 and cMTOP-P8 in Fig. S2. The convergence speed of BLKT-DE is faster than that of BLKT-DE-w/o-km. Therefore, we can conclude that knowledge transfer among similar dimensions is more effective than knowledge transfer among dissimilar dimensions.

Second, to analyze whether the intertask knowledge transfer is more effective, we design a BLKT-DE variant without intertask knowledge transfer, namely BLKT-DE-w/o-IKT. In BLKT-DE-w/o-IKT, the BLKT framework is carried out to only transfer intratask knowledge. Specifically, the block in task A can only evolve with the blocks that also come from task A. Therefore, the BLKT-DE-w/o-IKT can only transfer intratask knowledge rather than intertask knowledge.

The experimental results obtained by BLKT-DE and BLKT-DE-w/o-IKT on cMTOP are shown in Table S.VI of the supplemental material. Additionally, the convergence curves obtained by these two algorithms on cMTOP-P3 and cMTOP-P7 are shown in Fig. S3. According to the experimental results, BLKT-DE generally outperforms BLKT-DE-w/o-IKT. In all 20 tasks, BLKT-DE is superior to BLKT-DE-w/o-IKT in eight tasks, while BLKT-DE is inferior to BLKT-DE-w/o-IKT in only two tasks. Therefore, it can be concluded that intertask knowledge transfer is important and BLKT can effectively transfer knowledge among tasks.

G. Comparison on Real-World MTOPs

1) Comparison on Real-World MTOPs With Few-Tasks: On multitask planar kinematic arm control problems with few-tasks (i.e., 2-tasks, 5-tasks, and 10-tasks), the compared algorithms include MFEA-AKT, MFEA-II, adaptive evolutionary multitask optimization (AEMTO) [34], and ASCMFDE. In the experiment, the population size corresponding to each task is set as 100 and the number of generations is set as 100 for each algorithm. Since the scales of the fitness values of different tasks are different, the normalized fitness is adopted, here, to evaluate the performance of the algorithms, which is defined as

$$F_t = \frac{f_t - f_t^{\min}}{f_t^{\max} - f_t^{\min}} \tag{10}$$

where f_t is the fitness before normalization, and $f_t^{\min}$ and $f_t^{\max}$ are the minimal and maximal fitness values on the t-th task obtained by all the algorithms over all the executions.

The final mean normalized fitness values obtained by BLKT-DE and the compared EMTO algorithms are listed in Table III. Herein, MNF denotes the mean normalized fitness. It can be observed that BLKT-DE outperforms the compared algorithms in all the cases, and BLKT-DE achieves the best fitness values of all the compared algorithms.

2) Comparison on Real-World MTOPs With Many Tasks: Multitask planar kinematic arm control problems with many tasks include problems with 50-tasks and 100-tasks. The state-of-the-art evolutionary many-task optimization algorithms and AEMTO are adopted as compared algorithms. These evolutionary many-task optimization algorithms include the evolution of biocoenosis through symbiosis-based DE (EBS-DE) [69], many-task DE (MaTDE) [70], and the evolutionary

TABLE III
MEAN NORMALIZED FITNESS OBTAINED BY BLKT-DE AND THE COMPARED ALGORITHMS ON PLANAR KINEMATIC ARM CONTROL PROBLEMS WITH FEW-TASKS

Problem	BLKT-DE	MFEA-AKT	MFEA-II	AEMTO	ASCMFDE
	MNF*	MNF	MNF	MNF	MNF
2-tasks	**1.08E-04**	1.90E-02 (+)	3.24E-03 (+)	7.60E-03 (+)	6.53E-04 (+)
5-tasks	**3.29E-04**	8.49E-03 (+)	1.10E-03 (+)	1.01E-03 (+)	7.18E-04 (+)
10-tasks	**3.68E-04**	3.76E-03 (+)	7.02E-04 (+)	8.99E-04 (+)	6.63E-04 (+)
Number of + / ≈ / −		3 / 0 / 0	3 / 0 / 0	3 / 0 / 0	3 / 0 / 0

*MNF denotes the mean normalized fitness.

TABLE IV
MEAN NORMALIZED FITNESS OBTAINED BY BLKT-DE AND THE COMPARED ALGORITHMS ON PLANAR KINEMATIC ARM CONTROL PROBLEMS WITH MANY TASKS

Problem	BLKT-DE	AEMTO	EBS-DE	MaTDE	EMaTO-MKT
	MNF	MNF	MNF	MNF	MNF
50-tasks	**6.05E-04**	8.66E-04 (+)	7.64E-04 (+)	8.67E-04 (+)	9.12E-04 (+)
100-tasks	**6.60E-04**	8.62E-04 (+)	7.47E-04 (≈)	8.24E-04 (+)	7.91E-04 (≈)
Number of + / ≈ / −		2 / 0 / 0	1 / 1 / 0	2 / 0 / 0	1 / 1 / 0

many-task optimization algorithm based on a multisource knowledge transfer mechanism (EMaTO-MKT) [47]. The parameters of the compared algorithms are set as the optimal values as recommended in their papers.

The convergence curves obtained by BLKT-DE and the compared algorithms are shown in Fig. S4 of the supplemental material. The results for the mean normalized fitness are shown in Table IV. On both the 50-tasks case and the 100-tasks case, the proposed BLKT-DE shows comparable performance to the state-of-the-art evolutionary many-task optimization algorithms. In the 50-tasks problem, the convergence speed of BLKT-DE is faster than the compared algorithms and the final results obtained by BLKT-DE significantly better than the compared algorithms. In the 100-tasks problem, based on the result of Wilcoxon's rank-sum test, BLKT-DE outperforms AEMTO and MaTDE, while BLKT-DE achieves equivalent performance to EBS-DE and EMaTO-MKT.

H. Comparison on CEC17S and CEC22S

The second motivation for this article, as stated in Section II-C, is to improve efficiency by sharing knowledge among related dimensions belonging to the same task. If BLKT can meet this motivation, then BLKT-DE should also be able to solve the single-task global optimization problem effectively and efficiently. To show this, we first describe the difference between BLKT-DE for single-task global optimization and that for EMTO, and then conduct a comparison between BLKT-DE and several state-of-the-art DE-based algorithms on the CEC17S and CEC22S benchmarks. The pseudocode of BLKT-DE in solving the single-task global optimization problem is given in Algorithm 2. First, the individuals of a single population are divided into multiple blocks. Then, K-means clustering, mutation and crossover, and reconstruction operations are executed to obtain the offspring individuals. Different from BLKT-DE for the MTOP, BLKT-DE for single-task global optimization problems only maintains a single population. Additionally, the condition of FAS is different from that in BLKT-DE for MTOP.

Algorithm 2: BLKT-DE for Single-Task Global Optimization Problem

Input: f-the single-task optimization problem;
n -population size;
$K_{\min}$, $K_{\max}$-minimum and maximum values for cluster number K;
$B_{\min}$, $B_{\max}$-minimum and maximum values for block length B;
Output: S-the optimal solution;
Begin
Initialization population P; //Note that there is only one population.
Randomly sample K and B in $[K_{\min}, K_{\max}]$ and $[B_{\min}, B_{\max}]$;
While not termination
  Generate n offspring via BLKT as Fig. 3;
  Generate n offspring via DE/rand/1 mutation and crossover;
  Randomly select n offspring of BLKT and DE;
  Evaluate the fitness of the selected n offspring;
  Select n fitter solutions from n parents and n offspring;
  If the best solution is not updated
    Randomly sample K and B in $[K_{\min}, K_{\max}]$ and $[B_{\min}, B_{\max}]$;
  Else
    Randomly sample K and B in [K-1, K+1] and [B-1, B+1];
  End If
End While
Set S as the optimal solution of f;
End

TABLE V
EXPERIMENTAL RESULTS OBTAINED BY BLKT-DE AND THE COMPARED ALGORITHMS ON CEC22S

CEC22S Problem	BLKT-DE	JADE	HyDE-DF	DE-DVR	ADDE
P1	1.21E-07	4.09E+04 (+)	8.84E+01 (+)	**0.00E+00** (−)	3.16E+04 (+)
P2	4.84E+01	4.91E+01 (+)	**4.14E+01** (−)	4.85E+01 (≈)	4.91E+01 (+)
P3	**0.00E+00**	**0.00E+00** (≈)	4.22E-02 (+)	7.44E-08 (+)	**0.00E+00** (≈)
P4	3.03E+01	1.27E+02 (+)	**2.06E+01** (−)	3.75E+01 (+)	2.29E+01 (−)
P5	**0.00E+00**	**0.00E+00** (≈)	2.07E+00 (+)	**0.00E+00** (≈)	**0.00E+00** (≈)
P6	1.74E+03	4.01E+01 (−)	1.32E+02 (−)	**4.54E+00** (−)	3.01E+05 (+)
P7	**2.08E+01**	5.69E+01 (+)	2.71E+01 (+)	2.60E+01 (+)	4.11E+01 (+)
P8	**2.08E+01**	3.37E+01 (+)	2.27E+01 (+)	2.33E+01 (+)	2.51E+01 (+)
P9	**1.81E+02**	**1.81E+02** (≈)	1.81E+02 (+)	**1.81E+02** (≈)	**1.81E+02** (≈)
P10	1.12E+02	1.07E+02 (−)	1.13E+02 (+)	2.53E+02 (+)	**1.01E+02** (−)
P11	3.43E+02	3.17E+02 (−)	3.37E+02 (−)	3.20E+02 (−)	**3.10E+02** (−)
P12	2.39E+02	2.36E+02 (−)	2.49E+02 (+)	2.38E+02 (≈)	**2.36E+02** (−)
Number of +/≈/−		5 / 3 / 4	8 / 0 / 4	5 / 4 / 3	5 / 3 / 4

To validate BLKT-DE's performance, the comparison is conducted on the CEC17S and CEC22S problems. The results for the mean error obtained by BLKT-DE and the compared algorithms on CEC17S and CEC22S are shown in Table S.VII of supplementary material and Table V, respectively. The error is defined as the difference between the fitness value obtained by the algorithm and the true global optimum of this problem. Any error that is less than 10^{-8} is rounded to zero. In the results, it is interesting to find that BLKT-DE achieves generally better or comparable performance to the compared algorithms. In particular, BLKT-DE outperforms JADE, HyDE-DF, DE-DVR, and ADDE in 23, 16, 11, and 12 cases, respectively, while it is inferior in only 3, 8, 6, and 12 cases in CEC17S. BLKT-DE outperforms JADE, HyDE-DF, DE-DVR, and ADDE in 5, 8, 5, and 5 cases, respectively, while it is inferior in only 4, 4, 3, and 4 cases in CEC22S. Besides, we plot the convergence curves obtained by BLKT-DE and the compared algorithm on CEC17S-P5, CEC17S-P24, CEC17S-P27, and CEC22S-P9 in Fig. S5 of the supplemental material. As seen in the figures, BLKT-DE achieves comparable convergence rates to these compared algorithms. Remarkably, in some cases, BLKT-DE converges faster than the state-of-the-art DE-based algorithms.

Based on the above experiments and analysis, we can draw three conclusions. First, BLKT-DE outperforms or at least equals the state-of-the-art DE-based algorithms in terms of accuracy. With only the basic DE/rand/1 mutation operation, BLKT-DE achieves comparable performance to the compared algorithms, even though these compared algorithms adopt several adaptive parameter strategies and/or complex evolutionary operations. Second, BLKT-DE also achieves competitive performance to the compared algorithms in convergence speed. Third, and most importantly, the promising performance of BLKT-DE shows that transferring knowledge among related dimensions of the same task can significantly increase the algorithm's effectiveness and efficiency.

V. CONCLUSION

To address the limitation that the majority of the current EMTO algorithms concentrate solely on knowledge transfer between aligned dimensions, this article proposes an interesting and effective framework named BLKT to assist the knowledge transfer information between unaligned dimensions belonging to either different tasks or the same task. By performing mutation and crossover on the small blocks, BLKT achieves the purpose of transferring knowledge between similar dimensions belonging to either different tasks or the same task. In addition to the backbone of BLKT, FAS is further proposed to adaptively adjust the parameters in BLKT to achieve the best performance. We combine BLKT with DE and propose the BLKT-DE algorithm. To validate the performance of BLKT-DE, comparisons between BLKT-DE and several state-of-the-art EMTO algorithms are conducted on the widely used CEC17M, CEC22M, cMTOP, and real-world MTOPs. The experimental results on both benchmark MTOPs and real-world MTOPs show that the proposed BLKT-DE generally outperforms the compared state-of-the-art algorithms. Additionally, to validate whether transferring knowledge between different dimensions belonging to the same task can enhance the efficiency of BLKT, we design a BLKT-DE variant that only transfers knowledge between dimensions of the same task. We evaluate its performance for solving single-task global optimization problems on the CEC17S and CEC22S benchmarks. It is interesting and encouraging that the BLKT-DE with a very simple DE/rand/1 mutation operation can even achieve competitive performance compared to the DE-based state-of-the-art algorithms.

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, Jan. 2022.

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

[3] Z.-H. Zhan, J.-Y. Li, S. Kwong, and J. Zhang, "Learning-aided evolution for optimization," *IEEE Trans. Evol. Comput.*, early access, Dec. 29, 2022, doi: 10.1109/TEVC.2022.3232776.

[4] S.-C. Liu, Z.-G. Chen, Z.-H. Zhan, S.-W. Jeon, S. Kwong, and J. Zhang, "Many-objective job-shop scheduling: A multiple populations for multiple objectives-based genetic algorithm approach," *IEEE Trans. Cybern.*, vol. 53, no. 3, pp. 1460–1474, Mar. 2023.

[5] M. Zhou et al., "Adaptive genetic algorithm-aided neural network with channel state information tensor decomposition for indoor localization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 5, pp. 913–927, Oct. 2021.

[6] 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.

[7] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, and X.-Y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Inf. Sci.*, vol. 507, no. 1, pp. 67–85, Jan. 2020.

[8] X. Qiu, J.-X. Xu, Y. Xu, and K. C. Tan, "A new differential evolution algorithm for minimax optimization in robust design," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1355–1368, May 2018.

[9] Z.-G. Chen, Z.-H. Zhan, H. Wang, and J. Zhang, "Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 708–719, Aug. 2020.

[10] X.-F. Song, Y. Zhang, Y.-N. Guo, X.-Y. Sun, and Y.-L. Wang, "Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 882–895, Oct. 2020.

[11] Y. Xiang, Y. Zhou, Z. Chen, and J. Zhang, "A many-objective particle swarm optimizer with leaders selected from historical solutions by using scalar projections," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 2209–2222, May 2020.

[12] X.-F. Liu, Z.-H. Zhan, Y. Gao, J. Zhang, S. Kwong, and J. Zhang, "Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 587–602, Aug. 2019.

[13] 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.

[14] Z.-H. Zhan, J.-Y. Li, and J. Zhang, "Evolutionary deep learning: A survey," *Neurocomputing*, vol. 483, pp. 42–58, Apr. 2022.

[15] 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, Jan. 2022.

[16] M. Liu, B. Xu, and X. Peng, "Cooperative path planning for multi-AUV in time-varying ocean flows," *J. Syst. Eng. Electron.*, vol. 27, no. 3, pp. 612–618, Jun. 2016.

[17] Y. Zhang, D.-W. Gong, and J.-H. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing*, vol. 103, no. 1, pp. 172–185, Mar. 2013.

[18] Y. Xiang, X. Yang, Y. Zhou, and H. Huang, "Enhancing decomposition-based algorithms by estimation of distribution for constrained optimal software product selection," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 245–259, Apr. 2020.

[19] 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.

[20] 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.

[21] Y. Zhou, T. Wang, and X. Peng, "MFEA-IG: A multi-task algorithm for mobile agents path planning," in *Proc. IEEE Congr. Evol. Comput.*, 2020, pp. 1–7.

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

[23] C. Wang, J. Liu, K. Wu, and Z. Wu, "Solving multitask optimization problems with adaptive knowledge transfer via anomaly detection," *IEEE Trans. Evol. Comput.*, vol. 26, no. 2, pp. 304–318, Apr. 2022.

[24] 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.

[25] L. Zhou et al., "Toward adaptive knowledge transfer in multifactorial evolutionary computation," *IEEE Trans. Cybern.*, vol. 51, no. 5, pp. 2563–2576, May 2021.

[26] 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.

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

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

[29] Z. Tang, M. Gong, Y. Wu, W. Liu, and Y. Xie, "Regularized evolutionary multitask optimization: Learning to intertask transfer in aligned subspace," *IEEE Trans. Evol. Comput.*, vol. 25, no. 2, pp. 262–276, Apr. 2021.

[30] 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.

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

[32] L. Feng et al. "IEEE CEC 2022 competition on evolutionary multitask optimization." 2022. [Online]. Available: http://www.bdsc.site/websites/MTO_competition_2021/MTO_Competition_WCCI_2022.html

[33] Y. Jiang, Z.-H. Zhan, and J. Zhang, "A new and more challenging compositive multi-task optimization problem test suite," in *Proc. Int. Conf. Inf. Sci. Technol.*, 2022, pp. 132–138.

[34] H. Xu, A. K. Qin, and S. Xia, "Evolutionary multitask optimization with adaptive knowledge transfer," *IEEE Trans. Evol. Comput.*, vol. 26, no. 2, pp. 290–303, Apr. 2022.

[35] J.-B. Mouret and G. Maguire, "Quality diversity for multi-task optimization," in *Proc. Genet. Evol. Comput. Conf.*, 2020, pp. 121–129.

[36] N.-H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization," Nanyang Technol. Univ., Singapore, Rep., 2016.

[37] A. Kumar, K. V. Price, A. W. Mohamed, A. A. Hadi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the 2022 special session and competition on single objective bound constrained numerical optimization," Nanyang Technol. Univ., Singapore, Rep., 2021.

[38] H. Hu, Y. Zhou, T. Wang, and X. Peng, "A multi-task algorithm for autonomous underwater vehicles 3D path planning," in *Proc. Int. Conf. Unmanned Syst.*, 2020, pp. 972–977.

[39] 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.

[40] C. Yang, J. Ding, K. C. Tan, and Y. Jin, "Two-stage assortative mating for multi-objective multifactorial evolutionary optimization," in *Proc. IEEE 56th Annu. Conf. Decis. Control*, 2017, pp. 76–81.

[41] K. K. Bali, A. Gupta, Y.-S. Ong, and P. S. Tan, "Cognizant multitasking in multiobjective multifactorial evolution: MO-MFEA-II," *IEEE Trans. Cybern.*, vol. 51, no. 4, pp. 1784–1796, Apr. 2021.

[42] X. Wang, Q. Kang, M. Zhou, S. Yao, and A. Abusorrah, "Domain adaptation multitask optimization," *IEEE Trans. Cybern.*, early access, Nov. 29, 2022, doi: 10.1109/TCYB.2022.3222101.

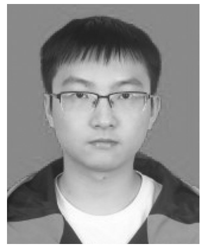
[43] Z. Liang, H. Dong, C. Liu, W. Liang, and Z. Zhu, "Evolutionary multitasking for multiobjective optimization with subspace alignment and adaptive differential evolution," *IEEE Trans. Cybern.*, vol. 52, no. 4, pp. 2096–2109, Apr. 2022.

[44] X. Ji, Y. Zhang, D. Gong, X. Sun, and Y. Guo, "Multisurrogate-assisted multitasking particle swarm optimization for expensive multimodal problems," *IEEE Trans. Cybern.*, vol. 53, no. 4, pp. 2516–2530, Apr. 2023.

[45] L. Feng et al., "Solving generalized vehicle routing problem with occasional drivers via evolutionary multitasking," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3171–3184, Jun. 2021.

[46] J. Liu, P. Li, G. Wang, Y. Zha, J. Peng, and G. Xu, "A multitasking electric power dispatch approach with multi-objective multifactorial optimization algorithm," *IEEE Access*, vol. 8, pp. 155902–155911, 2020.

[47] Z. Liang, X. Xu, L. Liu, Y. Tu, and Z. Zhu, "Evolutionary many-task optimization based on multisource knowledge transfer," *IEEE Trans. Evol. Comput.*, vol. 26, no. 2, pp. 319–333, Apr. 2022.
[48] 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.
[49] 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.*, vol. 26, no. 4, pp. 719–734, Aug. 2022.
[50] Y. Jiang, Z.-H. Zhan, K. C. Tan, and J. Zhang, "A bi-objective knowledge transfer framework for evolutionary many-task optimization," *IEEE Trans. Evol. Comput.*, early access, Sep. 29, 2022, doi: 10.1109/TEVC.2022.3210783.
[51] S.-H. Wu, Z.-H. Zhan, K. C. Tan, and J. Zhang, "Orthogonal transfer for multitask optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 1, pp. 185–200, Feb. 2023.
[52] S.-H. Wu, Z.-H. Zhan, K. C. Tan, and J. Zhang, "Transferable adaptive differential evolution for many-task optimization," *IEEE Trans. Cybern.*, early access, Feb. 8, 2023, doi: 10.1109/TCYB.2023.3234969.
[53] J. Lin, H.-L. Liu, K. C. Tan, and F. Gu, "An effective knowledge transfer approach for multiobjective multitasking optimization," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3238–3248, Jun. 2021.
[54] X. Wu, W. Wang, H. Yang, H. Han, and J. Qiao, "Diversified knowledge transfer strategy for multitasking particle swarm optimization," *IEEE Trans. Cybern.*, early access, Jan. 9, 2023, doi: 10.1109/TCYB.2022.3232113.
[55] X. Xia et al., "Triple archives particle swarm optimization," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4862–4875, Dec. 2020.
[56] 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.
[57] J.-Y. Li, Z.-H. Zhan, C. Wang, H. Jin, and J. Zhang, "Boosting data-driven evolutionary algorithm with localized data generation," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 923–937, Oct. 2020.
[58] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[59] Z.-J. Wang et al., "Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 894–908, Dec. 2018.
[60] R. B. Agrawal and K. Deb, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.
[61] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Comput. Sci. Inform.*, vol. 26, no. 4, pp. 30–45, 1996.
[62] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.
[63] F. Lezama, J. Soares, R. Faia, and Z. Vale, "Hybrid-adaptive differential evolution with decay function (HyDE-DF) applied to the 100-digit challenge competition on single objective numerical optimization," in *Proc. Genet. Evol. Comput. Conf.*, 2019, pp. 7–8.
[64] A. Ghosh, S. Das, A. K. Das, and L. Gao, "Reusing the past difference vectors in differential evolution—A simple but significant improvement," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4821–4834, Nov. 2020.
[65] 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.
[66] J. Carrasco, S. García, M. M. Rueda, S. Das, and F. Herrera, "Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review," *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100665.
[67] Z.-H. Cui and J.-C. Zeng, "Schema theorem of real-coded nonlinear genetic algorithm," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2002, pp. 1429–1431.
[68] R. Poli, "Why the schema theorem is correct also in the presence of stochastic effects," in *Proc. Congr. Evol. Comput.*, 2000, pp. 487–492.
[69] R.-T. Liaw and C.-K. Ting, "Evolutionary many-tasking based on biocoenosis through symbiosis: A framework and benchmark problems," in *Proc. IEEE Congr. Evol. Comput.*, 2017, pp. 2266–2273.
[70] 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.



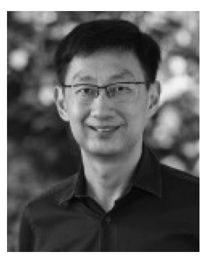
Yi Jiang (Student Member, IEEE) received the B.S. degree in computer science and technology from the South China University of Technology, Guangzhou, China, in 2020, where he is currently pursuing the Ph.D. degree in computer science and technology.

His research interests mainly include computational intelligence, evolutionary computation, multitask optimization, machine learning, and their applications in real-world problems.

Zhi-Hui Zhan (Senior Member, IEEE) received the bachelor's degree and the Ph.D. degree in computer science from 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 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. 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 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*, the *Memetic Computing*, and the *Machine Intelligence Research*.



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 (Computational Intelligence) with the Department of Computing, The 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 the Chief Co-Editor of Springer Book Series on *Machine Learning: Foundations, Methodologies, and Applications*, and an editorial board member for more than ten journals.

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, Ansan, 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 CYBERNETICS and the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.