Evolutionary Multitask Framework With Bi-Knowledge Transfer for Multimodal Optimization Problems

Hong Zhao, *Member, IEEE*, Xu-Hui Ning, Jian-Yu Li, *Member, IEEE*, and Jing Liu, *Senior Member, IEEE*

***Abstract*—Solving multimodal optimization problems (MMOPs) is a challenging task which needs locating multiple global optimal solutions simultaneously with high accuracy. Current popular niching-based evolutionary algorithms (EAs) for solving MMOPs usually divide the population into several separate species to search for different optimal solutions. However, achieving effective information exchange between species to enhance the performance of overall algorithm remains a challenge in current niching-based EAs, which will directly affect the efficiency of the multimodal optimization algorithm. In this article, the process of the different species locating peaks in MMOPs is regarded as an evolutionary multitask (EMT) optimization problem and an EMT framework with bi-knowledge transfer for MMOPs is proposed. An explicit knowledge transfer (E-KT) strategy is designed to transfer the optimal individual of the species with the fastest convergence speed to other species, thereby facilitating the acceleration their convergence. Moreover, in order to further improve the information exchange between species, a species-center-based implicit knowledge transfer (I-SCKT) strategy is designed to improve the diversity of the population. The performance of $\mathbf{MTBKT_{MMOP}}$ is tested on the widely used CEC'2013 benchmark and five practical flexible job shop problems. The experimental results of $\mathbf{MTBKT_{MMOP}}$ are compared with nine state-of-the-art MMOPs algorithms and show that our $\mathbf{MTBKT_{MMOP}}$ is superior to all of them. Besides, the experimental results also show that the $\mathbf{MTBKT_{MMOP}}$ achieves breakthroughs in handling with a large number of optimal solutions or high-dimensional MMOPs, which provides a new and effective method for dealing with MMOPs.**

***Index Terms*—Differential evolution (DE), evolutionary computation, knowledge transfer, multimodal optimization problem (MMOP), multitasking, nearest-better clustering (NBC).**

Received 21 October 2024; revised 17 January 2025 and 19 February 2025; accepted 7 March 2025. Date of publication 17 March 2025; date of current version 2 February 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62306225, Grant 62471371, and Grant 62406152; and in part by the Natural Science Foundation of Tianjin under Grant 24JCQNJC02100. This article was approved by Associate Editor A. Gupta. *(Corresponding author: Hong Zhao.)*

Hong Zhao, Xu-Hui Ning, and Jing Liu are with the Guangzhou Institute of Technology, Xidian University, Guangzhou 510555, China (e-mail: hongzhao@xidian.edu.cn).

Jian-Yu Li is with the School of Artificial Intelligence, Nankai University, Tianjin 300071, China.

This article has supplementary downloadable material available at https://doi.org/10.1109/TEVC.2025.3551728, provided by the authors.

Digital Object Identifier 10.1109/TEVC.2025.3551728

I. Introduction

MULTIMODAL optimization problems (MMOPs) aim to simultaneously find multiple global optimal solutions that are existed in different regions in the same search space. The comprehensive information of these optimal solutions can serve as a valuable reference for decision-makers [1]. In the process of solving MMOPs using evolutionary algorithms (EAs) [2], how to maintain population diversity is a key technique. However, the traditional EAs are designed for searching a single global optimum, the population in traditional EAs often lacks the diversity, making it easy to miss out on finding all the optima.

In order to effectively solve MMOPs, combining EAs with niching techniques is a commonly adopted approach to enhance population diversity in dealing with MMOPs, such as clearing [3], fitness sharing [4], neighborhood mutation [5], etc. By utilizing niching techniques, EAs can track multiple promising regions within the search space, enabling more efficient solving of MMOPs. However, the existing niching techniques usually divides the population into independent niches or overlapping niches, and each niche evolves independently or interactively during the evolutionary process, which often leads to the insufficient of information exchange between each niche. Therefore, designing appropriate strategies to facilitate information exchange between species to achieve efficient MMOPs solving remains a challenge.

In this article, in order to achieve effective information exchange between species during evolution, the process of the different species locating peaks in MMOPs is treated as a multitask optimization problems (MTOPs) [6]. Fig. 1 illustrates the process of transforming a MMOPs into a MTOPs. In Fig. 1(a), an MMOP with five optimal solutions is presented, where red circles denote the initial distribution of the population, and the dashed-line ellipses highlight the five optima to be found, labeled as Peak 1 to Peak 5. Traditional evolutionary processes gradually locate these optima over iterations, as shown in Fig. 1(a1)–(a4), which depict the population distributions after the 10th, 20th, 50th, and final generations, respectively. It can be observed that the convergence rate of the population is relatively slow. Given that each optimum in an MMOP has an equal fitness value, once one of the optima is located, it is possible to record the evolutionary path taken to find this solution. This

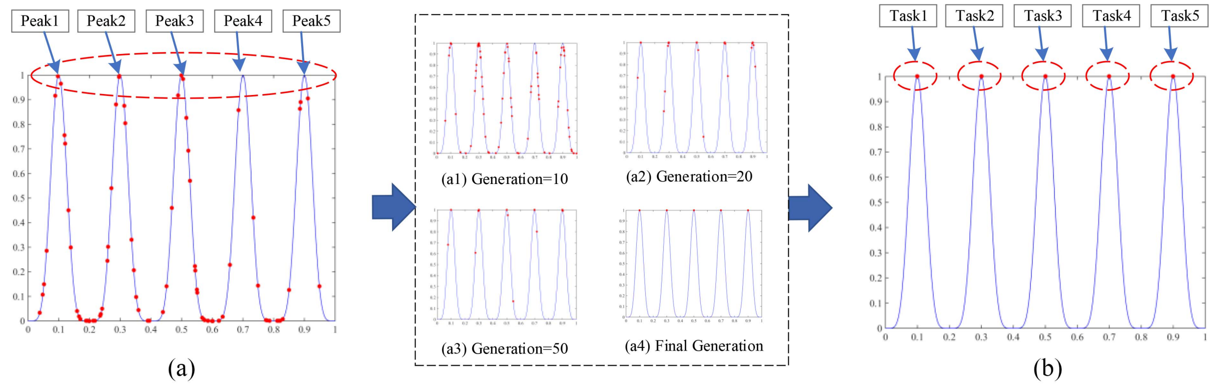

Fig. 1. Schematic of (a) MMOPs transforming into (b) MTOPs.

recorded information can then serve as heuristic guidance for locating the remaining optima. By treating the search for each optimum as a separate task, the process of found all global optima transforms into a multitask optimization problem. The beneficial heuristic information obtained while searching for a single optimum can thus be viewed as positive knowledge transfer aiding the search for other optima [7], [8]. Fig. 1(b) provides a schematic representation of the multitask framework corresponding to the search for each peak (Peak 1 to Peak 5), referred to as Task 1 to Task 5. The aim is to leverage knowledge transfer among tasks to accelerate the convergence of the population and identify more optimal solutions efficiently. Through this transformation, not only is the speed of finding multiple optima increased but also the effectiveness of the optimization process is enhanced by utilizing the shared information across different tasks.

Based on the above comprehensive analysis, this article proposes for the first time the use of multitask optimization methods to solve MMOPs. Concretely, the population is divided into multiple species. By conducting an in-depth analysis of the relationship between the optimal solutions and the species centers within the best-performing and worst-performing species, effective knowledge transfer strategies are designed. Evolutionary information from the best-performing species is used to assist the evolution of individuals in the worst-performing species, thereby accelerating the convergence of the entire population. Simultaneously, to prevent individuals within species from getting trapped in local optima, two species are randomly selected, and their species centers undergo random perturbation. The newly generated individuals from this process replace the worst-performing individuals within those species. In conclusion, two improved knowledge transfer strategies are designed to promote information exchange between species during evolution. First, an explicit knowledge transfer (E-KT) strategy is designed to achieve explicitly information exchange between species, which promotes the convergence of the population in current task. In detail, E-KT transfers the optimal individual from species with fastest convergence to those species with slower convergence, thereby facilitating the acceleration of convergence in the latter species, that is, E-KT can promote the overall convergence of the algorithm. Moreover, a species-center-based implicit knowledge transfer (I-SCKT) strategy is designed to achieve implicitly information exchange between species, which improves the diversity of the population. Therefore, in this article, an evolutionary multitask (EMT) framework with bi-knowledge transfer is proposed to assist niching-based EAs in solving MMOPs.

Based on the above method and framework, an EMT algorithm with bi-knowledge transfer is proposed for solving MMOPs ($\text{MTBKT}_{\text{MMOP}}$). First, the nearest-better clustering (NBC) [9] strategy is employed as the niching technique to divide population into several species because NBC does not require predefining the size of species and allows the population to be divided into species of any size [10]. This enables the algorithm to explore the search space more effectively. Then the differential evolution (DE) [11], [12] is used as an optimizer for the species evolution. The main contributions of this article can be summarized as follows.

1) A novel multitask framework is proposed for solving MMOPs, which can help niching-based EAs deal with MMOPs more efficiently.
2) The novel E-KT strategy is designed for two specific different species in population to enhance information exchange between species. The E-KT strategy transfers the optimal individual from species that exhibit fastest convergence to those species with slower convergence, which can accelerate the convergence of these species, resulting in improving the overall convergence speed of the algorithm.
3) The novel I-SCKT strategy is designed for two randomly different species in population to further strengthen the information exchange between species, which can improve the diversity of the population.
4) The proposed $\text{MTBKT}_{\text{MMOP}}$ algorithm has demonstrated exceptional performance on CEC'2013 benchmark and practical flexible job scheduling problems (FJSP). Specifically, in the CEC'2013 benchmark $\text{MTBKT}_{\text{MMOP}}$ can locate all global optimal solutions on test function F_1-F_{19}, 70% of global optimal solutions on test function F_{20}, among which Fxx-F_{20} are high dimensional MMOPs. This represents a significant breakthrough for multimodal optimization algorithms dealing with high-dimensional MMOPs.

The remainder of this article is structured as follows. Section II provides a review of fundamental concepts, including DE, EMT, and MMOPs. In Section III, the process of $\text{MTBKT}_{\text{MMOP}}$ is described. In Section IV, the comprehensive experimental results and analysis of $\text{MTBKT}_{\text{MMOP}}$ are presented. Then, $\text{MTBKT}_{\text{MMOP}}$ is compared with different recently advanced algorithms, and the analysis of different components is discussed. In Section IV, $\text{MTBKT}_{\text{MMOP}}$ is applied to solve the FJSP. Finally, in Section VI, a comprehensive conclusion is provided.

II. Related Work

In this section, first, a brief description of DE is provided, followed by a review of the current DE methods for handling MMOPs. Finally, the existing works on EMT are reviewed.

A. DE

DE is one of the most powerful population-based algorithms designed for continuous optimization [11], [12], [13]. The DE process includes initialization, mutation, crossover, and selection, which starts with a randomly generated initial population, as shown in

$$x_{i,j}^0 = x_{\text{lb},j} + \text{rand}(0,1) \times \left(x_{\text{ub},j} - x_{\text{lb},j}\right) \tag{1}$$

where $x_{i,j}^0$ represents the jth dimension value of the ith individual of the initial population, $x_{lb,j}$ and $x_{ub,j}$ are the lower and upper bounds of the jth dimension, respectively, rand(0,1) is a uniform random number from 0 to 1, i and j are in the range of [1, NP] and [1, D], respectively, where NP is the population size and D is the dimension of the optimization problem.

1) Mutation: In each generation, the trail vector $\mathbf{v}_i$ is generated by the mutation according to the parent individuals $\mathbf{x}_i$. The five common mutation operators are listed as (2)–(6):

DE/rand/1

$$v_i^g = x_{r_1}^g + F \times \left(x_{r_2}^g - x_{r_3}^g\right). \tag{2}$$

DE/rand/2

$$v_i^g = x_{r_1}^g + F \times \left(x_{r_2}^g - x_{r_3}^g\right) + F \times \left(x_{r_4}^g - x_{r_5}^g\right). \tag{3}$$

DE/best/1

$$v_i^g = x_{\text{best}}^g + F \times \left(x_{r_2}^g - x_{r_3}^g\right). \tag{4}$$

DE/best/2

$$v_i^g = x_{\text{best}}^g + F \times \left(x_{r_1}^g - x_{r_2}^g\right) + F \times \left(x_{r_3}^g - x_{r_4}^g\right). \tag{5}$$

DE/current-to-best/1

$$v_i^g = x_i^g + F \times \left(x_{\text{best}}^g - x_i^g\right) + F \times \left(x_{r_1}^g - x_{r_2}^g\right) \tag{6}$$

where the indexes r_1, r_2, r_3, r_4 and r_5 are randomly selected from $\{1, 2, \ldots, NP\} \setminus \{i\}$ and g denotes the current generation. $\mathbf{x}^{\text{g}}{}_{\text{best}}$ is the individual with the best fitness value in the gth generation. The factor F is a control parameter for scaling the difference vector.

2) Crossover: The crossover is employed to exchange some components of the trail vector $\mathbf{v}_i$ and parent individuals $\mathbf{x}_i$ to generate the offspring $\mathbf{u}_i$, which can be expressed as

$$u_{i,j}^g = \begin{cases} v_{i,j}^g, & \text{if rand}(0,1) \le CR \text{ or } j = j_{\text{rand}} \\ v_{i,j}^g, & \text{otherwise} \end{cases} \tag{7}$$

where CR is the crossover probability which is usually a fixed value from [0, 1]. j_{rand} is a randomly selected dimension index of dimension, which is used to ensure $\mathbf{u}^{\text{g}}{}_i$ has at least one dimension different from $\mathbf{x}^{\text{g}}{}_i$.

3) Selection: The selection operation is used to select the better individuals from offspring and parent individuals to enter the next generation. For a maximization problem, the individual with a superior objective function value can be selected into the next generation. The process of selection is shown as

$$x_i^{g+1} = \begin{cases} u_i^g, & \text{if } f\left(x_i^g\right) \le f\left(u_i^g\right) \\ x_i^g, & \text{otherwise} \end{cases} \tag{8}$$

where $f(.)$ is the fitness value of the maximization problem.

B. DE for MMOPs

The definition of the MMOPs depends on the type of optimization problem (minimization or maximization), herein, for a maximization problem, given f : H→R, we would like to find all global maximums of f in a single run. A global maximum $h^* \in$H of the objective function f : H→R is an input element with $f(h^*) \geq f(h)$ for all h neighboring h^* ($\forall h \in$H).

To enable DE to effectively handle MMOPs, Goldberg and Richardson [14] proposed sharing DE (SDE), which is based on the basic idea of penalizing individuals occupying the same region in the search space by considering the number of nearby individuals, thereby locating multiple peaks simultaneously. Thromsen [15] introduced the concept of crowding DE (CDE) that can protect species diversity, which can provide better chances of finding multiple optimal solutions. Qu et al. [5] proposed a neighborhood mutation strategy for CDE (NCDE) which is performed within each Euclidean neighborhood of individuals, this mutation strategy can maintain the multiple optima discovered during the evolutionary process. Similarly, by modifying the DE mutation operator, Gao et al. [16] proposed a new mutation operation to implement adaptive control of F and used a simple adaptive strategy to dynamically update CR. Biswas et al. [17] proposed a parent-centric normalized mutation with proximity-based crowding DE (PNPCDE) to improve the convergence speed of the algorithm.

To obtain better species division in niching-based DE, Biswas et al. [18] proposed an information sharing mechanism among individuals in DE (LoICDE) algorithm to induce effective species division. Zhao et al. [19] proposed a local binary pattern (LBP)-based adaptive DE (LBPADE) algorithm for MMOPs that uses LBP to divide the population into multiple species, thereby enhancing the ability of DE to locate multiple peak regions. Wang et al. [20] used the distribution information of each individual estimated by nearby individuals to adaptively determine the size of the species and proposed a parameter-free niching method based on adaptive estimation distribution and develop a distributed differential evolution algorithm for solving MMOPs. Jiang et al. [21] treated the niche center distinguish (NCD) problem is treated as an optimization problem and an NCD-based differential evolution (NCD-DE) algorithm is proposed to solve MMOP. By determining the appropriate position of the niche center, which can help individuals belonging to the niche center accelerate convergence to the corresponding optima.

Besides, Hui and Suganthan [22] proposed an approach for DE that applies arithmetic recombination with speciation to enhance exploration and uses neighborhood mutation with ensemble strategies to improve exploitation of individual peaks. Zhou et al. [23] proposed a modification to DE by incorporating a sorting *CR* method. In this approach, individuals with better fitness values are assigned higher *CR* values, increasing their probability of being selected for the offspring. This modification helps to improve the exploitation of high-quality solutions, thereby enhancing the overall performance of the algorithm. Chen et al. [24] treated each individual as a distributed unit capable of finding a peak and each individual uses a virtual population controlled by an adaptive range adjustment strategy to assist with the mutation process. This approach can effectively maintain sufficient diversity to locate more peaks. Wei et al. [25] proposed a penalty strategy with a DE (PMODE) algorithm to penalize solutions in the neighboring areas of previously discovered local and global optima, which can help explore new regions and locate more global optima.

Recently, many scholars have utilized some new technologies and theories to better address MMOPs. Chen et al. [26] utilized community information from complex networks to automatically divide niches and employs the nodes and edges of the network to record the positions of individuals and their historical evolutionary paths, respectively. Based on this methodology, a network community-based differential evolution algorithm has been developed for MMOPs (NetCDE). Li et al. [27] proposed a minimum spanning tree niching-based differential evolution (TNDE) with knowledge-driven update strategy to avoid local optima refining the accuracy of solutions.

These methods have offered new perspectives on solving MMOPs and have achieved considerable success. Nevertheless, there has been no attempt to convert MMOPs into multitask optimization problems. Thus, this article analyzes the features of MMOPs and proposes a novel approach to solve them by framing MMOPs as multitask optimization problems.

C. EMT

EMT optimization refers to the evolutionary process of optimizing N tasks simultaneously. A basic minimization EMT optimization problem can be expressed as

$$z_n^* = \operatorname{argmin}_{z\in\mathbb{Z}_n} f_n(z), n = 1, \dots, N. \tag{9}$$

Suppose H_n denotes the nth task, the objective function of the nth task H_n is defined as $f_n : \mathbb{Z}_n \rightarrow R$, where $\mathbb{Z}_n$ is a D_n dimension search space. While the knowledge transfer occurs between different tasks $[H_1, H_2, \dots, H_N]$, N means the number of tasks.

The concept of EMT optimization draws inspiration from multitask learning [28] and transfer learning [29], EMT optimization actually means a set of approaches that are designed by EAs to solve multiple related tasks simultaneously. Moreover, EMT works on the principle that if some common useful knowledge exists in solving a task, then the useful knowledge gained in the process of solving this task may help to solve another task that is related to it. In fact, EMT optimization makes full use of the implicit parallelism of population-based search.

There are two main representations of knowledge. The first is the straightforward representation [6], which considers elite individuals as knowledge. These elite individuals are transferred to the population of target task to aid in its evolution. The second method of knowledge representation considers promising search directions [30] obtained during task-solving as knowledge. This type of knowledge can guide the evolution of the population. E-KT aims to use species with fastest convergence to assist species with slow convergence, thus choosing straightforward representation to achieve the purpose of promoting the convergence of the target species [31].

The first attempt of EMT is the multifactorial EA, Gupta et al. [6] introduced EMT as a new paradigm in the field of optimization and evolutionary computation and first formalized the concept of EMT. In recent years, there has been a significant amount of work utilizing EMT to solve practical problems. Gong et al. [32] designed a novel EMT algorithm with an online dynamic resource allocation strategy, which allocates resources to each task adaptively according to the requirements of tasks. Qiao et al. [33] proposed an EMT optimization framework for constrained multiobjective optimization problems (CMOPs), the optimization of the CMOP is transformed into two related tasks: one task is for the original CMOP, and the other task is only for the objectives by ignoring all constraints. Li et al. [34] devised a new EMT algorithm for feature selection in high-dimensional classification, which first adopts different filtering methods to produce multiple tasks and modifies a competitive swarm optimizer to efficiently solve these tasks via knowledge transfer.

Inspired by the success of EMT in solving complex optimization problems by leveraging the experience of simple optimization problems, Yang et al. [35] developed a novel multitasking framework whose effectiveness is demonstrated in solving the costly task offloading problem. By implementing a knowledge-sharing mechanism among multiple search processes, Zhou et al. [36] presented an evolutionary multitask convolutional neural architecture search framework to enable efficient architecture searches in multitask scenarios by incorporating architectural similarities. Liaw and Wen [37] proposed an innovative framework for ensemble learning through EMT and formulated a classification problem as a dynamic MTOP. Huang et al. [38] presented a novel EMT algorithm with centralized learning for solving the large-scale and multiobjective combinatorial optimization in many-task manner, in which knowledge transfer across tasks is conducted based on a centralized learning model. By discovering and utilizing local knowledge across modalities of different tasks, Gao et al. [39] designed a distributed knowledge transfer-based evolutionary multitask multimodal optimization approach for solving multiple MMOPs simultaneously, which deals with multitask multimodal optimization problems that aims to simultaneously handle multiple MMOPs using a multitask approach. In contrast, our proposed method focuses on a single MMOP, with the goal of finding multiple optimal solutions within that single problem using a multitask approach.

Given the successful experience of EMT in handling complex optimization problems and practical issues, this article employs EMT to address MMOPs. By designing two effective knowledge transfer strategies (i.e., E-KT and I-SCKT), the approach assists the population in concurrently locating multiple optima and avoids local optima.

III. $\text{MTBKT}_{\text{MMOP}}$

In this section, we first introduce the framework of $\text{MTBKT}_{\text{MMOP}}$; Then, two components of $\text{MTBKT}_{\text{MMOP}}$: E-KT and I-SCKT are introduced, respectively; Finally, the complete $\text{MTBKT}_{\text{MMOP}}$ is presented.

A. Framework of $MTBKT_{MMOP}$

To effectively transform an MMOP into an MTOP, this article proposes a multitask-for-multimodal framework. Fig. 2 provides a flowchart of the multitask-for-multimodal framework. In the context of MMOPs, the population is first divided into multiple species, assuming that each species contains an optimum to be found. The process of seeking optima within different species can then be treated as independent tasks. Next, knowledge transfer between species—such as between the best-performing and worst-performing species—is employed to assist in exploring optimal solution regions across different species. This article introduces two strategies for knowledge transfer: E-KT and I-SCKT. Finally, after each generation completes the knowledge transfer, the population is redivided, and the species are updated until all optima are located or the termination criteria are met. By adopting this approach, not only can the convergence of the population be accelerated effectively, but also the diversity of the population can be maintained, thereby enhancing the performance of the algorithm.

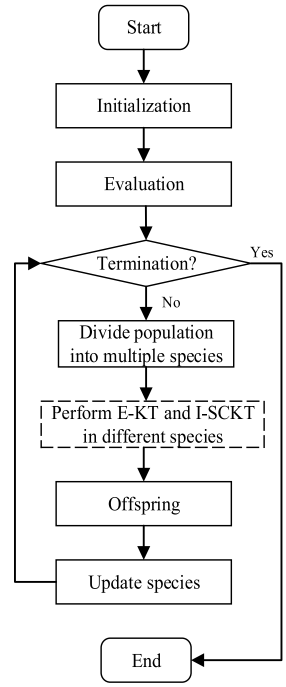

Fig. 2. Framework of $\text{MTBKT}_{\text{MMOP}}$.

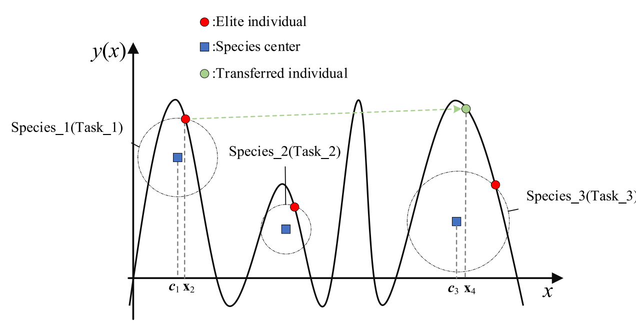

Fig. 3. Diagram of E-KT.

B. E-KT

As depicted in Fig. 3, the species distribution of the population at a specific evolutionary stage consists of three species: 1) species_1; 2) species_2; and 3) species_3. For the sake of brevity, only the species center and the optimal individual (i.e., the fittest individual within a species) of each species are indicated. At this stage, it can be seen that species_1 exhibits the fastest convergence (as indicated by its elite individual's superior fitness). The species centers, denoted as c_1 and c_3, correspond to the center positions of species_1 and species_3, respectively. We define the species center as the convergence point toward which all species or all elite individuals within the species converge. The elite individuals of a species are closer to the optima, and the center of these elite individuals can reflect the center of this species more precisely. Consequently, the individuals within a species can be ranked in descending order of their fitness values, and the species center can be computed based on the distribution of the top 50% of individuals with the highest fitness. To ensure the number of baseline individuals is an integer, the number (M_k) of species S_k can be calculated as

$$M_k = \left\lceil \frac{\text{size of } S_k}{2} \right\rceil \tag{10}$$

where *size of* S_k is the number of individuals of species S_k. Then the center of S_k that marked $\mathbf{C}_k$ can be calculated as

$$C_k = \left[\frac{\sum_{m=1}^{M_k} x_{m,1}}{M_k}, \frac{\sum_{m=1}^{M_k} x_{m,2}}{M_k}, \ldots, \frac{\sum_{m=1}^{M_k} x_{m,D}}{M_k} \right] \tag{11}$$

where m means the mth individual in M_k.

The E-KT between species_1 and species_3 in the decision space is performed as follows.

1) The distance vector $\boldsymbol{L}$ between the $\mathbf{C}_1$ and $\mathbf{C}_3$ in the decision space is calculated as: $\boldsymbol{L} = C_3 - C_1$.

Algorithm 1 E-KT

Input: S_{best} and S_{target}
Output: S_{target}

Calculate M_{best} and M_{target} by Eq. (10);
Calculate $\mathbf{C}_{best}$ and $\mathbf{C}_{target}$ by Eq. (11);
$\boldsymbol{L} = \mathbf{C}_{target}$ - $\mathbf{C}_{best}$;
Select the best individual $\mathbf{x}_{best}$ from S_{best}.
$\mathbf{x}_{transfer} = \mathbf{x}_{best} + \boldsymbol{L}$;
if $f(\mathbf{x}_{transfer})$ > the optima fitness value of S_{target} **then**
    Replace the best individual of S_{target} with $\mathbf{x}_{transfer}$;
end if
: Output S_{target}.

2) The decision vector $\mathbf{x}_4$ of the transferred individual is calculated as: $\mathbf{x}_4 = \mathbf{x}_2 + \boldsymbol{L}$.

The species with fastest convergence transfers optimal individual to those species with slow convergence by using E-KT and then promotes the convergence of the whole algorithm.

As illustrated in Algorithm 1, the inputs S_{best} and S_{target} refer to the species with best fitness value and species to receive knowledge, respectively, in the current generation. First, the species sizes M_{best} and M_{target} for S_{best} and S_{target} are calculated using (10). Next, the species centers $\mathbf{C}_{\text{best}}$ and $\mathbf{C}_{\text{target}}$ for S_{best} and S_{target} are calculated using (11). Then, calculate the distance vector $\boldsymbol{L}$ between the species centers $\mathbf{C}_{\text{best}}$ and $\mathbf{C}_{\text{target}}$. Using the distance vector $\boldsymbol{L}$, the knowledge individual $\mathbf{x}_{\text{transfer}}$ is calculated. Finally, the best individual in S_{target} is replaced with $\mathbf{x}_{\text{transfer}}$ when the fitness value of $\mathbf{x}_{\text{transfer}}$ is bigger than the optimal fitness value of S_{target}.

To conduct a more in-depth analysis of the principles behind E-KT, we can transform the transfer solution $\mathbf{x}_{\text{transfer}}$, i.e., $\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{best}} + (\mathbf{C}_{\text{target}}$ - $\mathbf{C}_{\text{best}})$, into the form of $\mathbf{x}_{\text{new}} = \mathbf{C}_{\text{target}} + (\mathbf{x}_{\text{best}}$ - $\mathbf{C}_{\text{best}}) = \mathbf{C}_{\text{target}} + delta_{\text{best}}$, where $delta_{\text{best}}$ is the difference between the best solution $\mathbf{x}_{\text{best}}$ and its centroid $\mathbf{C}_{\text{best}}$ and $\mathbf{x}_{\text{new}}$ is the $\mathbf{x}_{\text{transfer}}$. In earlier generations, the difference between $\mathbf{x}_{\text{best}}$ and $\mathbf{C}_{\text{best}}$ can bring in suitable difference to enhance the diversity of population, improving the peak exploration ability of the algorithm. As population evolves, we will observe that the centroid of each species converges toward the best solutions, so this $delta_{\text{best}}$ will converge to 0 ($\mathbf{x}_{\text{best}} = \mathbf{C}_{\text{best}}$) in the later generations, meaning that the E-KT in the later generations corresponds to just proposed as a solution the centroid solution of the species, which again is a reasonable solution as the population of the species may be converging around and toward the optimal solution. Additionally, the phenomenon that the center of each species gradually converges near the optimal solution within its species as the population evolves has already been verified in [21], which support the above analysis.

As population evolves, individuals in the population gradually converge toward the neighborhood of the optimal solution, and the targeted individuals for transfer within the species become the optimal solution, meaning $\mathbf{C}_{\text{target}} = \mathbf{C}_{\text{best}}$. Additionally, according to the preceding analysis, as the population evolves, $delta_{\text{best}}$ will converge to 0 ($\mathbf{x}_{\text{best}} = \mathbf{C}_{\text{best}}$). This implies that the newly generated individual $\mathbf{x}_{\text{new}}$ is the optimal solution within their current species in later generations, that is $\mathbf{x}_{\text{new}} = \mathbf{C}_{\text{target}} = \mathbf{x}_{\text{best}}$. Therefore, this demonstrates that the E-KT strategy promotes convergence of the population and moves individuals toward the optimal solution.

Moreover, suppose the proposed solution is the center of the centroids with some random noise, that is $\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{best}} + (\mathbf{C}_{\text{target}}$ - $\mathbf{C}_{\text{best}}) + beta$, where beta means some random noise. Building on the earlier analysis, as the population evolves, we can achieve $\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{best}} + beta$. In fact, in E-KT, to ensure positive transfer, a comparison of the fitness values of $\mathbf{x}_{\text{new}}$ and $\mathbf{x}_{\text{best}}$ in target is made each time; that is, $\mathbf{x}_{\text{new}}$ will only replace $\mathbf{x}_{\text{best}}$ if the fitness value of $\mathbf{x}_{\text{new}}$ is greater than that of $\mathbf{x}_{\text{best}}$. According to this evolutionary mechanism, $\mathbf{x}_{\text{new}}$ will still eventually equal to $\mathbf{x}_{\text{best}}$, meaning that the noise beta gradually tends toward 0. This again demonstrates that the E-KT strategy causes individuals in the population to gradually converge toward the near of the optimal solution. Although it seems that it is possible to add random noise (e.g., the beta mentioned above) to the E-KT, how to determine the suitable noise range for different generations and evolution states is challenging, which is not the focus of this article and could be considered in future studies. Based on the above, as both the original E-KT and its noise version can promote population convergence and move individuals toward the optimal solution, the original E-KT is used in this article.

C. I-SCKT

To enhance the information exchange between species and increase population diversity, an I-SCKT is developed to generate new individuals to facilitate the exploration of the solution space. The mutation operator is a vector-based search method that provides a scalable and versatile approach to conducting the search process. Our I-SCKT is inspired by this principle, utilizing the vector-based operations to transfer information between species and enable effective exploration of the search space. The new individual generated in I-SCKT is calculated as

$$x_{\text{new}} = C_{\text{sk}} + \lambda \times (C_{\text{sa}} - C_{\text{sb}}) \tag{12}$$

where $\mathbf{x}_{\text{new}}$ is a new generated individual, $\mathbf{C}_{sk}$ is the center vector of the species S_k to be evolved in current generation, λ is a control parameter for scaling $\mathbf{x}_{\text{new}}$, $\mathbf{C}_{sa}$ and $\mathbf{C}_{sb}$ are the center vector of the species S_a and S_b in the previous generation, respectively. As shown in Algorithm 2, first, M_k, M_a and M_b for S_k, S_a and S_b are calculated by using (10). Next, the species centers $\mathbf{C}_{sk}$, $\mathbf{C}_{sa}$ and $\mathbf{C}_{sb}$ for S_k, S_a and S_b are calculated by using (11). Then, the new individual $\mathbf{x}_{\text{new}}$ is calculated using (12). Finally, the worst individual in S_k is replaced with $\mathbf{x}_{\text{new}}$.

Similarly inspired by the variation operation, our proposed I-SCKT primarily consists of two parts. In the first part, we utilize the center $\mathbf{C}_{sk}$ of species $\mathbf{S}_k$ as the base direction. In the second part, we randomly select the centers $\mathbf{C}_{sa}$ and $\mathbf{C}_{sb}$ of two other species $\mathbf{S}_a$ and $\mathbf{S}_b$ to perform a linear difference and incorporate a scaling factor λ to control the magnitude of the difference vector. Note that the idea of this

Algorithm 2 I-SCKT

Input: S_k, S_a, and S_b
Output: S_k

Calculate M_k, M_a and M_b by Eq. (10);
Calculate $\mathbf{C}_{sk}$, $\mathbf{C}_{sa}$, and $\mathbf{C}_{sb}$ by Eq. (11);
Calculate $\mathbf{x}_{new}$ by Eq. (12);
Replace the worst individual of S_a with $\mathbf{x}_{new}$;
Output S_k.

approach is somewhat similar to the distance-dependent variance directional Gaussian described in [40], which indicates the effectiveness of the I-SCKT design. However, the proposed I-SCKT is different with the method in [40] from two aspects. First, the method in [40] uses Gaussian-based difference to ensure continual exploration around the promising individual. However, the proposed I-SCKT uses the difference between centroids of different species to transfer useful knowledge, so as to improve the diversity of multiple species; Second, the method in [40] focuses on exploiting the correlations between the elites in population to find the high-performing solutions, whereas our proposed I-SCKT emphasizes utilizing the differences between any two distinct random species centers within the population to prevent individuals from getting trapped in local optima.

Further analysis reveals that when λ equals 0, the newly generated individual $\mathbf{x}_{\text{new}}$ becomes the center $\mathbf{C}_{sk}$ of the current species $\mathbf{S}_k$, at which $\mathbf{C}_{sk}$ directly replaces the worst individual in the current species $\mathbf{S}_k$. According to our previous analysis of the E-KT strategy, as the population evolves, the species center $\mathbf{C}_{sk}$ will become the optimal individual within the current species. In I-SCKT, this means directly replacing the worst-performing or locally-optimal-trapped individuals with the best individual from the population, thereby accelerating the convergence of the population. When λ equals 1, the newly generated individual $\mathbf{x}_{\text{new}}$ is the center $\mathbf{C}_{sk}$ of the current species plus the maximum difference vector produced by $\mathbf{C}_{sa}$ and $\mathbf{C}_{sb}$, i.e., $\boldsymbol{x}_{new}=\boldsymbol{C}_{sk}+\lambda\times(\mathbf{C}_{sa}-\mathbf{C}_{sb})$, which perturbs the center $\mathbf{C}_{sk}$ of the current species. In this way, the proposed I-SCKT strategy increases population diversity and further explores more regions of optimal solutions. In summary, our proposed I-SCKT strategy can effectively balance the diversity and convergence of the population.

D. Two-Stage Mutation

To enhance the searchability and convergence of the algorithm, we design a two-stage mutation (TSM) strategy that employs different mutation operators at different stages of evolution. During the early stage of evolution, the mutant operators with strong exploration capabilities are more suitable to increase population diversity and allow species to explore as many areas as possible. In the later stage of evolution, the mutant operators with strong exploitation capability are more suitable to promote the convergence of species toward the optimal peaks.

In this article, DE/rand/1 and DE/rand/2 are chosen as exploration mutations in the early stage of evolution, and DE/best/1 and DE/best/2 as exploitation mutations in the later stage of evolution. The parameter of mutation-selection probability mp is set as $1-(fes/MaxFEs)^{\alpha}$, where fes represents the current number of evaluations consumed, $MaxFEs$ is the maximum number of evaluations of the current problem, and α is the control parameter for balancing exploration and exploitation. Algorithm 3 outlines the TSM strategy used in this article. First, two random numbers *num_1* and *num_2* are generated between 0 and 1. Next, *num_1* decides which mutation operator style is selected (exploration or exploitation), and *num_2* decides which mutation operator type is selected. Repeat the above steps until the size of mS_k is equal to the size of S_k.

Algorithm 3 TSM

Input: S_k, $MaxFEs$, mp
Output: mS_k

Set $mS_k = \emptyset$;
While the size of $mS_k \leq$ the size of S_k **Do**
  num_1 = rand (0,1);
  num_2 = rand (0,1);
  If $num1 < mp$ **Then**
    If $num2 < 0.5$ **Then**
      Generate $\mathbf{v}_i$ by DE/rand/1;
    else
      Generate $\mathbf{v}_i$ by DE/rand/2;
    End If
  else
    If $num2 < 0.5$ **Then**
      Generate $\mathbf{v}_i$ by DE/best/1;
    else
      Generate $\mathbf{v}_i$ by DE/best/2;
    End If
  End If
  Put $\mathbf{v}_i$ into mS_k;
EndWhile

E. $MTBKT_{\text{MMOP}}$

Based on these above components, the process of $\text{MTBKT}_{\text{MMOP}}$ *is* shown in Algorithm 4. First, the input parameters $MaxFEs$, D, φ, α, F, and CR are set; NP is calculated according (16); The *BS_archive* and *RS_archive* is set to store the best species and two random species in the previous generation, respectively. Next, randomly generate the initial population P_0 and evaluate it. Then use NBC-Minsize to divide P_0 into several species set $\boldsymbol{S}$. Traverse all the species in $\boldsymbol{S}$, if the current generation g is equal to 1, the offspring is generated through mutation and crossover. Otherwise, perform E-KT and I-SCKT for the current species, then the offspring is generated through TSM and crossover. Finally, selection is performed to choose the individuals with better fitness values for the next generation population. This process is repeated until the termination condition is met.

The main concept behind NBC is that the distance between individuals near different optima is larger than the weighted

Algorithm 4 $\text{MTBKT}_{\text{MMOP}}$

Input: *MaxFEs*, D, φ, F, *CR*
Output: Found optimal solutions

```
Calculate the population size NP by Eq. (16);
Randomly generate the initial population P_0;
fes = 0, g = 0;
Evaluate P_0;
fes = fes + NP;
Set BS_archive = Ø; // Store best species
Set RS_archive = Ø; // Store two random species
While fes < MaxFEs Do
   Divide P_0 into species and stores in set S;
   For each species S_k of S Do
      If g > 1 Then
         Get the best species S_best from BS_archive;
         Perform E-KT on S_k according to Algorithm 1;
         fes = fes +1;
         Get species S_a, and S_b from RS_archive;
         Perform I-SCKT on S_k according to Algorithm 2;
      End If
      Generate v^g_i by TSM according to Algorithm 3;
      Generate u^g_i by Eq. (7);
      Select the better individual into P_{g+1} by Eq. (8);
      fes = fes + |S_k|;
   End For
   Update BS_archive with the best species;
   Update RS_archive with two random species;
   g = g + 1;
EndWhile
Output found all optimal solutions.
```

average distance of the rest of the individuals to their nearest-better neighbors. To implement this idea, the first step of NBC is to rank all individuals in the population in descending order based on their fitness values. The second step involves calculating the distance between each pair of individuals. Individuals, except for the best individual, can find their nearest-better neighbor and create an edge to connect themselves and their neighbor, forming a spanning tree. Finally, the mean distance μ of these edges is calculated, and edges longer than $\varphi \times \mu$ are cut off, where φ is the control parameter of NBC used to ensure that the number of species is not too large or too small [10]. A small φ results in more species, while increasing φ leads to fewer species. In most cases, φ is set to 2.

The previous formed spanning tree is then divided into a number of subtrees, each represents a species, and the roots of the subtrees are considered to be the seeds of the corresponding species. NBC-Minsize [41] is an improved NBC, the φ of NBC-Minsize is set to 1 which can increase the number of species and the parameter *minsize* of NBC-Minsize is set as

$$\min\text{size}(g) = 5 + \frac{g}{2} \tag{13}$$

where g is the current generation. To prevent the *minzize* is too large because g might be very large, the upper bound of *minsize* is set no more than max (10, 3*D).

IV. Experimental Results

A. Benchmark Problems and Experimental Settings

In this section, the performance of $\text{MTBKT}_{\text{MMOP}}$ is evaluated on CEC's 2013 niching benchmark suite [42]. This is a standard benchmark test suite used to evaluate algorithms in solving MMOPs. The suite contains 20 test functions with different numbers of global peaks (NKP) and D. Details of the benchmark suite can be found in Table S.I of the supplementary materials, which includes D, NKP, r (niching radius), *MaxFEs* (total number of fitness evaluations), and *NP*.

The performance of $\text{MTBKT}_{\text{MMOP}}$ is evaluated using two commonly used quantified methods, namely, peak ratio (PR) and success rate (SR) [42]. PR represents the average percentage of optimal peaks found by $\text{MTBKT}_{\text{MMOP}}$ across multiple runs and can be calculated as

$$\text{PR} = \frac{\sum_{i=1}^{\text{NR}} \text{NP}F_i}{NKP \times NR} \tag{14}$$

where NPF_i is the number of optimal peaks found by the algorithm at ith run and *NR* is the total number of runs. While SR represents the percentage of successful run which means finding all the optimal peaks in a single run, which can be calculated as

$$SR = \frac{\text{NSR}}{\text{NR}} \tag{15}$$

where NSR is the number of runs that the algorithm finds all optimal peaks, and *NR* is the total number of runs.

$\text{MTBKT}_{\text{MMOP}}$ is run 51 times independently, the PR and SR are computed in five accuracy levels $\epsilon = \{1.0\text{E-}01, 1.0\text{E-}02, 1.0\text{E-}03, 1.0\text{E-}04, 1.0\text{E-}05\}$. The population size *NP* is calculated according to the maximum evolutionary generations *MaxGen* and *MaxFEs*, which can be expressed as

$$NP = \left\lceil \frac{\text{MaxFEs}}{\text{MaxGen}} \right\rceil. \tag{16}$$

Wilcoxon's rank-sum test [43] at significance level $\alpha = 0.05$ in PR is employed to judge whether the $\text{MTBKT}_{\text{MMOP}}$ is significantly better than "+," worse than "−," or similar to "≈" the compared algorithms. In the $\text{MTBKT}_{\text{MMOP}}$, the parameters are set as described in Table S.II of the supplementary materials, the φ of NBC-Minsize is set to 1 to increase the number of species, and the parameter minsize of NBC-Minsize is set to 5 to ensure the mutation can be carried out since DE/rand/2 (i.e., (2)) and DE/best/2 (i.e., (4)) require at least 5 individuals. *CR* is set to 0.9 to increase the global exploration capability of population. In each mutation, the value of F is randomly generated from 0.2 to 0.8 to increase the diversity of population. *MaxGen* is set to 200 when D is greater than 5, and 300 otherwise. The value of α in TSM is set to 2 to achieve the best performance of $\text{MTBKT}_{\text{MMOP}}$ and the sensitivity analysis α of is presented in Section IV-E. In each E-KT, the value of λ is randomly generated from 0.5 to 1 to ensure the algorithm explore the unexplored regions as much as possible.

TABLE I
SUMMARIZED COMPARISON RESULTS ON PR AND SR AT FIVE DIFFERENT ACCURACY LEVELS

ϵ	F_1		F_2		F_3		F_4		F_5	
	PR	*SR*	*PR*	*SR*	*PR*	*SR*	*PR*	*SR*	*PR*	*SR*
1.0E-01	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-02	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-03	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-04	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-05	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
ϵ	F_6		F_7		F_8		F_9		F_{10}	
	PR	*SR*	*PR*	*SR*	*PR*	*SR*	*PR*	*SR*	*PR*	*SR*
1.0E-01	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-02	1.000	1.000	1.000	1.000	0.996	0.961	1.000	1.000	1.000	1.000
1.0E-03	1.000	1.000	1.000	1.000	0.974	0.902	1.000	1.000	1.000	1.000
1.0E-04	1.000	1.000	1.000	1.000	0.920	0.745	1.000	1.000	1.000	1.000
1.0E-05	1.000	1.000	1.000	1.000	0.827	0.588	1.000	1.000	1.000	1.000
ϵ	F_{11}		F_{12}		F_{13}		F_{14}		F_{15}	
	PR	*SR*	*PR*	*SR*	*PR*	*SR*	*PR*	*SR*	*PR*	*SR*
1.0E-01	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-02	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-03	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-04	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-05	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
ϵ	F_{16}		F_{17}		F_{18}		F_{19}		F_{20}	
	PR	*SR*	*PR*	*SR*	*PR*	*SR*	*PR*	*SR*	*PR*	*SR*
1.0E-01	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.716	0.706
1.0E-02	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.716	0.706
1.0E-03	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.713	0.686
1.0E-04	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.711	0.686
1.0E-05	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.706	0.647

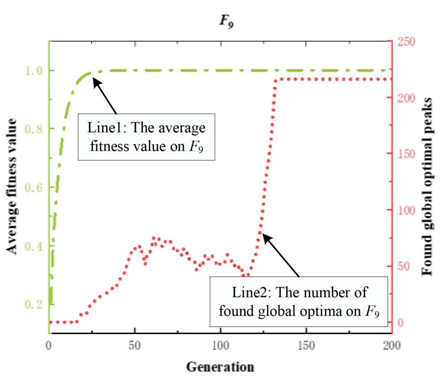

Fig. 4. *Line* 1 represents the average fitness value of $\text{MTBKT}_{\text{MMOP}}$ on F_9 (global optima is 1), *Line* 2 represents the number of global optimal solutions have been found by $\text{MTBKT}_{\text{MMOP}}$ on F_9 (NKP is 216).

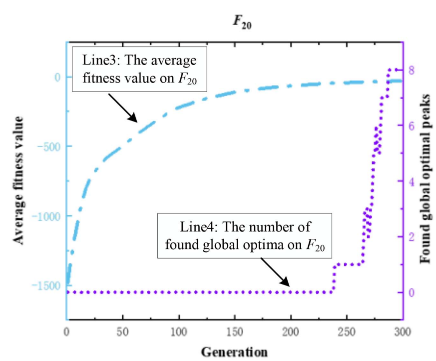

Fig. 5. *Line* 3 represents the average fitness value of $\text{MTBKT}_{\text{MMOP}}$ on F_{20} (global optima is 0), *Line* 4 represents the number of global optimal solutions have been found by $\text{MTBKT}_{\text{MMOP}}$ on F_{20} (NKP is 8).

B. Experimental Results of MTBKTMMOP on CEC'2013 Benchmark

We first give the comparison results of $\text{MTBKT}_{\text{MMOP}}$ in terms of PR and SR in accuracy of 1.0E-01 to 1.0E-05. The performance of $\text{MTBKT}_{\text{MMOP}}$ for all benchmark problems is presented in Table I. The results indicate that $\text{MTBKT}_{\text{MMOP}}$ is able to discover all global optimal solutions in 18 test functions (F_1-F_{18}) and more than 90% of all global optimal solutions in test functions (F_{19}-F_{20}).

Additionally, as demonstrated in Fig. 4, *Line* 1 shows that $\text{MTBKT}_{\text{MMOP}}$ is successful in finding all the 216 global optimal solutions of F_9 in the mid-stage of evolution, indicating good convergence. Similarly, as demonstrated in Fig. 5, *Line* 3 shows that $\text{MTBKT}_{\text{MMOP}}$ is able to find all the global optimal solutions of F_{20}, a 20-D test function, although the convergence is not as fast as in other test functions, it is still better than other comparison algorithms.

C. Compared With the State-of-the-Art Multimodal Algorithms

In this section, the $\text{MTBKT}_{\text{MMOP}}$ is compared with 13 popular comparison algorithms, such as PNPCDE [17], LoICDE [18], the distance-based locally informed particle swarm (LIPS) algorithm [45], which uses multiple local best particles instead of a global best particle to guide the search of each particle. The multiobjective optimization for MMOPs (MOMMOP) algorithm [46], which models MMOP

TABLE II
SUMMARIZED COMPARISON RESULTS IN TERMS OF THE WILCOXON RANK-SUM TEST AMONG $MTBKT_{MMOP}$ AND THE COMPARED ALGORITHMS

$MTBKT_{MMOP}$	LIPS	LoICDE	PNPCDE	MOMMOP	LMCEDA	LMSEDA	LAMC-ACO
+/≈/−	17/3/0	14/6/0	14/6/0	10/9/1	14/6/0	15/5/0	14/6/0
$MTBKT_{MMOP}$	LAMS-ACO	DSDE	NCD-DE	PMODE	NetCDE	TNDE	
+/≈/−	14/6/0	11/9/0	10/9/1	11/9/0	12/7/1	9/10/1	

as a multiobjective optimization problem with the multimodal function itself as the first objective and gradient information or population distance information as the second objective. The local search-based multimodal estimation of distribution (LMEDAs) algorithms, which include LMCEDA and LMSEDA [47], generate offspring using Gaussian distribution and Cauchy distribution alternately. The local search-based adaptive multimodal continuous ant colony optimization (LAM-ACO) algorithms, which include LAMC-ACO and LAMS-ACO [48], use DE mutation operator to construct new solutions for ants. The dual-strategy DE (DSDE) and DSDE with selection operator in CDE (DSDE-C) algorithms [49] use a dual-strategy mutation scheme and an adaptive selection mechanism based on affinity propagation clustering to locate as many peaks as possible by selecting individuals from different optimal regions. Besides, to highlight the novelty of the comparative algorithms, we have included four multimodal optimization algorithms from the past two years, namely NCD-DE [21], PMODE [25], NetCDE [26], and TNDE [27].

As the results at ϵ = 1.0E-01 and ϵ = 1.0E-02 are not precise, for the sake of simplicity, the comparison of results has been carried out at the commonly used accuracy level ϵ = 1.0E-04.

Table II shows the summarized comparison results in terms of the Wilcoxon rank-sum test (α=0.05) among $MTBKT_{MMOP}$ and the compared algorithms, where row "+/≈/−" represents $MTBKT_{MMOP}$ is significantly better/similar/worse with the compared algorithms. The detailed results are shown in Table S.III of the supplementary material. The *NP* of all the algorithms are set to their default values. The analysis of the results in Table II and Table S.III are as follows.

1) For test functions F_1-F_5, $MTBKT_{MMOP}$ and most algorithms can find all the global optimal solutions.
2) For the test functions F_6-F_9 that have numerous global optimal solutions, only MOMMOP is able to find all the global optimal solutions and our $MTBKT_{MMOP}$ also can find all the global optima except for F8. Besides, our $MTBKT_{MMOP}$ consistently finds optimal solutions more than 75% of the time for F8, which outperforms all other compared algorithms except for MOMMOP.
3) For the test functions F_{10}-F_{15}, our $MTBKT_{MMOP}$ is able to find all global optimal solutions. Specially, only $MTBKT_{MMOP}$ can find all global optimal solutions on F_{11}-F_{15}, while half of the comparison algorithms can locate all global optimal solutions on F_{10}, and only DSDE/DSDE-C and $MTBKT_{MMOP}$ can find all global optimal solutions on F_{11}-F_{12}. $MTBKT_{MMOP}$ performs significantly better than other compared algorithms on F_{13}-F_{15}.
4) For the test functions F_{16}-F_{20}, $MTBKT_{MMOP}$ gets the best results and performs significantly better than other compared algorithms on these test functions. It can be observed that $MTBKT_{MMOP}$ performs well on high-dimensional test functions and identifies all the global optimal solutions on F_{16}-F_{18}. Furthermore, $MTBKT_{MMOP}$ can identify 75% of the global optimal solutions of F_{19} and over 90% of the global optimal solutions of F_{20}.

TABLE III
RANK IN PR AND SR OF DIFFERENT ALGORITHMS

Algorithm	PR_{mean}	PR_{std}	*averank*
$MTBKT_{MMOP}$	**0.9815**	**0.0623**	**1.05**
LIPS	0.5028	0.4270	5.95
LoICDE	0.4086	0.4625	8.65
PNPCDE	0.4200	0.5506	6.3
MOMMOP	0.7837	0.3857	3.25
LMCEDA	0.6474	0.3952	4.2
LMSEDA	0.7087	0.3578	4.1
LAMC-ACO	0.6898	0.3519	3.15
LAMS-ACO	0.7352	0.3618	2.7
DSDE/DSDE-C	0.6774	0.3672	2.55
NCD-DE	0.7120	0.3722	4.6
PMODE	0.6671	0.3635	4.55
NetCDE	0.7300	0.3445	3.07
TNDE	0.7334	0.3511	2.33

In all, $MTBKT_{MMOP}$ dominates LIPS, LoICDE, PNPCDE, MOMMOP, LMCEDA, LMSEDA, LAMC-ACO, LAMS-ACO, DSDE/DSDE-C, NCD-DE, PMODE, NetCDE, and TNDE on 17, 14, 14, 10, 14, 15, 14, 14, 11, 10, 11, 12, and 9 test functions, respectively. Only MOMMOP can dominate $MTBKT_{MMOP}$ on F_8. LIPS, LoICDE, PNPCDE, MOMMOP, LMCEDA, LMSEDA, LAMC-ACO, LAMS-ACO, DSDE/DSDE-C, NCD-DE, PMODE, NetCDE, and TNDE can just be similar to $MTBKT_{MMOP}$ on 3, 6, 6, 9, 6, 5,6,6, 9, 9, 9, 7, and 10 test functions, respectively.

Table III presents a comparison between $MTBKT_{MMOP}$ and 13 comparison algorithms, where PR_{mean} and PR_{std} show the average values and the standard deviations of PR on the 20 benchmark problems. For each test function, the results of algorithms are ranked from the best to worst and each algorithm can get a rank number. The *averank* is the average value of ranks of each algorithm, it is evident that the $MTBKT_{MMOP}$ performs the best among these algorithms. As shown in Table III, our $MTBKT_{MMOP}$ achieves the best performance on metrics PR_{mean}, PR_{std}, and *averank*. This indicates that our algorithm not only performs well but also demonstrates robustness in handling MMOPs.

D. Comparisons With Winners of CEC Competitions

According to the experimental results mentioned above, it is shown that our proposed $MTBKT_{MMOP}$ algorithm demonstrates significant advantages for solving MMOPs while

TABLE IV
COMPARISON RESULTS BETWEEN $\text{MTBKT}_{\text{MMOP}}$, NEA2, NMMSO AT ACCURACY $\epsilon = 1.0\text{E-4}$

Func	$\text{MTBKT}_{\text{MMOP}}$		NEA2		NMMSO	
	PR	*SR*	*PR*	*SR*	*PR*	*SR*
F_1	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_2	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_3	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_4	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_5	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_6	**1.000**	1.000	0.950	0.380	0.992	0.880
F_7	**1.000**	1.000	0.914	0.040	**1.000**	1.000
F_8	**0.920**	0.745	0.240	0.000	0.889	0.020
F_9	**1.000**	1.000	0.581	0.000	0.978	0.120
F_{10}	**1.000**	1.000	0.988	0.860	**1.000**	1.000
F_{11}	**1.000**	1.000	0.960	0.760	0.990	0.940
F_{12}	**1.000**	1.000	0.840	0.160	0.993	0.940
F_{13}	**1.000**	1.000	0.957	0.740	0.983	0.900
F_{14}	**1.000**	1.000	0.807	0.060	0.720	0.000
F_{15}	**1.000**	1.000	0.718	0.000	0.632	0.000
F_{16}	**1.000**	1.000	0.673	0.000	0.660	0.000
F_{17}	**1.000**	1.000	0.695	0.000	0.468	0.000
F_{18}	**1.000**	1.000	0.667	0.000	0.650	0.000
F_{19}	**1.000**	1.000	0.667	0.000	0.450	0.000
F_{20}	**0.711**	0.686	0.360	0.000	0.172	0.000
bprs	**20**		5		7	

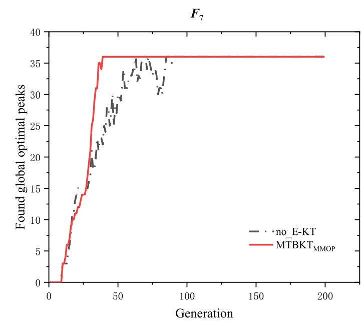

Fig. 6. Number of global optima $\text{MTBKT}_{\text{MMOP}}$ and no_E-KT found on test function F_7 during evolution process.

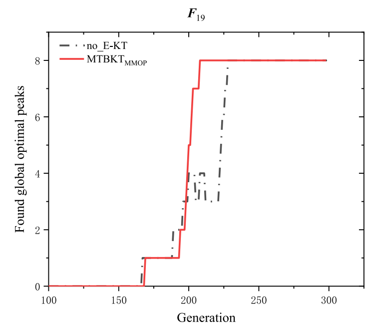

Fig. 7. Number of global optima $\text{MTBKT}_{\text{MMOP}}$ and no_E-KT found on test function F_{19} during evolution process.

compared with other existing methods. In this section, we aim to compare our $\text{MTBKT}_{\text{MMOP}}$ algorithm with the ultimate winner algorithms of CEC'2013 and CEC'2015 competitions, namely the NBC (NEA2) algorithm [9] and the niching migratory multiswarm optimizer (NMMSO) algorithm [50]. The comparison results at the accuracy level $\epsilon = 1.0\text{E-4}$ are shown as Table IV and further validate the effectiveness of our $\text{MTBKT}_{\text{MMOP}}$ on MMOPs.

As shown in Table IV, the comparison results of PR and SR between $\text{MTBKT}_{\text{MMOP}}$ and the winner algorithms are listed in detail, where the best PR is highlighted in *boldface* and the *bprs* represents the number of best PR that each algorithm achieved on the total 20 functions. Overall, our $\text{MTBKT}_{\text{MMOP}}$ performs best among the three algorithms and performs best on all 20 functions, while the NEA2 algorithm performs best on 5 functions and the NMMSO algorithms perform best on seven functions, which obviously illustrates the superiority of the proposed $\text{MTBKT}_{\text{MMOP}}$. Meanwhile, it is worth mentioning that the PR value of our $\text{MTBKT}_{\text{MMOP}}$ algorithm achieves 1.000 on 18 functions, only except for F_8 with 0.920 and F_{20} with 0.711. From Table IV, we can also find that the NMMSO has a better performance than NEA2 on low-dimensional functions, such as F_7 with 1.000, F_8 with 0.889 and F_9 with 0.978. On the other hand, the NEA2 performs better on high-dimensional functions than NMMSO, especially on F_{17} with 0.695, F_{19} with 0.667, and F_{20} with 0.360. Nevertheless, our $\text{MTBKT}_{\text{MMOP}}$ shows a better performance not only on low-dimensional functions but also on high-dimensional functions, which indicates that the great ability of $\text{MTBKT}_{\text{MMOP}}$ in dealing with MMOPs.

In conclusion, it can be observed that the $\text{MTBKT}_{\text{MMOP}}$ is significantly better than and more effective than the winners of CEC'2013 and CEC'2015 competitions for MMOPs.

E. Component Analysis and Parameter Investigation

1) Effectiveness of E-KT: The effectiveness of the E-KT in $\text{MTBKT}_{\text{MMOP}}$ is discussed in this section. The purpose of equipping E-KT for $\text{MTBKT}_{\text{MMOP}}$ is to promote the convergence of the current species by using the optimal individual provided by the best species in the previous generation. Herein, a $\text{MTBKT}_{\text{MMOP}}$ variant (referred to as no_E-KT) is employed as a comparison algorithm which does not have E-KT component while the other components are the same as $\text{MTBKT}_{\text{MMOP}}$. $\text{MTBKT}_{\text{MMOP}}$ and no_E-KT are run on test function F_7 and F_{19} and record the number of global optima they found during evolution process. As shown in Figs. 6 and 7, $\text{MTBKT}_{\text{MMOP}}$ converges faster than no_E-KT on both F_7 and F_{19}, which demonstrates the effectiveness of the E-KT component in $\text{MTBKT}_{\text{MMOP}}$.

To evaluate the impact of E-KT on the $\text{MTBKT}_{\text{MMOP}}$ algorithm, we designed a set of experiments without E-KT, labeled as no_E-KT. Table V presents the experimental results under both the E-KT strategy and the no_E-KT strategy, obtained by running the algorithms 51 times at 1.0E-4 accuracy and averaging the outcomes. We used "bprs" to count the number of optimal experimental results. As shown in Table V, the algorithm with the E-KT strategy outperforms across all 20 test functions, whereas the no_E-KT strategy only performs best on the F1-F5 test functions and shows poorer performance on complex high-dimensional functions. Therefore, it can

TABLE V
EXPERIMENTS RESULTS MTBKT$_{\text{MMOP}}$, NO_E-KT, AND NO_I-SCKT AT ACCURACY $\epsilon = 1.0\text{E-}4$

Func	MTBKT$_{\text{MMOP}}$		no_E-KT		no_I-SCKT	
	PR	*SR*	*PR*	*SR*	*PR*	*SR*
F_1	**1.000**	1.000	**1.000**	1.000	0.990	0.980
F_2	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_3	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_4	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_5	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_6	**1.000**	1.000	0.625	0.000	0.798	0.000
F_7	**1.000**	1.000	0.551	0.000	0.728	0.000
F_8	**0.920**	0.745	0.282	0.000	0.445	0.000
F_9	**1.000**	1.000	0.166	0.000	0.356	0.000
F_{10}	**1.000**	1.000	0.538	0.000	**1.000**	1.000
F_{11}	**1.000**	1.000	0.432	0.000	**1.000**	1.000
F_{12}	**1.000**	1.000	0.667	0.000	0.855	0.118
F_{13}	**1.000**	1.000	0.325	0.000	0.980	0.882
F_{14}	**1.000**	1.000	0.333	0.000	0.758	0.039
F_{15}	**1.000**	1.000	0.211	0.000	0.693	0.000
F_{16}	**1.000**	1.000	0.375	0.000	0.667	0.000
F_{17}	**1.000**	1.000	0.435	0.000	0.662	0.000
F_{18}	**1.000**	1.000	0.225	0.000	0.667	0.000
F_{19}	**1.000**	1.000	0.000	0.000	0.463	0.000
F_{20}	**0.711**	0.686	0.000	0.000	0.091	0.000
bprs	**20**		5		6	

be concluded that E-KT has a significant impact on high-dimensional complex problems, that is the effective knowledge transfer can greatly enhance the ability of algorithm to handle such complex high-dimensional problems.

2) Effectiveness of I-SCKT: The I-SCKT aims to facilitate species information exchange, thereby increasing population diversity. In this section, the effectiveness of the proposed I-SCKT is verified by comparing it with a MTBKT$_{\text{MMOP}}$ variant (referred to as no_I-SCKT) that lacks the I-SCKT component, while keeping the other components the same as in MTBKT$_{\text{MMOP}}$. The results, presented in Table V, clearly demonstrate a significant performance reduction in the I-SCKT-absent MTBKT$_{\text{MMOP}}$.

As can be seen from Table V, without component I-SCKT, the algorithm performs poorly on the F1, F6-F9, and F12-F20 test functions, while it exhibits relatively stable performance on the F2-F5 and F10-F11 test functions. This may be due to the characteristics of the test functions themselves. For instance, the F1, F6-F9, and F12-F20 test functions contain multiple local optima, and without component I-SCKT, some individuals in the population are prone to getting trapped in these local optima during evolution, leading to a decrease in population diversity. Conversely, with component I-SCKT included, except for F8 and F20, our algorithm is able to find all global optima (i.e., both PR and SR are 1.000) across the board.

The test functions F2-F5 and F10-F11 feature multiple global optima, and for these functions, the component I-SCKT has a lesser impact on performance. Therefore, this observation further demonstrates that the component I-SCKT we have designed helps enhance the diversity among individuals in the population and prevents them from being trapped in local optima. Despite the improved convergence of MTBKT$_{\text{MMOP}}$ with the assistance of E-KT, the absence of I-SCKT limits the ability of population to diversify and discover a greater number of global optima.

3) Sensitivity Analysis α: In this section, different values of α are compared according to the results of numerical experiments. α is used to balance the exploration and exploitation capability of MTBKT$_{\text{MMOP}}$. Herein, four different MTBKT$_{\text{MMOP}}$ variants MTBKT$_{\text{MMOP}}$-1/2, MTBKT$_{\text{MMOP}}$-1, MTBKT$_{\text{MMOP}}$-3 and MTBKT$_{\text{MMOP}}$-4 corresponding to the values of α 1/2, 1, 3 and 4, respectively. While MTBKT$_{\text{MMOP}}$ adopts the value of α 2.

As shown in Table S. IV of the supplementary material, MTBKT$_{\text{MMOP}}$ achieves the best *bprs* (i.e., 20). There is no difference among MTBKT$_{\text{MMOP}}$, MTBKT$_{\text{MMOP}}$-1/2, MTBKT$_{\text{MMOP}}$-1, MTBKT$_{\text{MMOP}}$-3 and MTBKT$_{\text{MMOP}}$-4 on test functions F_1-F_{19}. MTBKT$_{\text{MMOP}}$-1/2, MTBKT$_{\text{MMOP}}$-1, MTBKT$_{\text{MMOP}}$-3 and MTBKT$_{\text{MMOP}}$-4 performs not so well as MTBKT$_{\text{MMOP}}$ on high-dimensional (20D) test functions F_{20}. This indicates the performance of the proposed MTBKT$_{\text{MMOP}}$ algorithm is not very sensitive to the value of α and can get good performance on high-dimensional test functions.

V. MTBKT$_{\text{MMOP}}$ IN SOLVING FJSP

In recent years, some multimodal optimization methods have been used to solve many real-world problems, including protein structure prediction [51], electromagnetic design [52], and pedestrian detection [53]. In this study, the proposed MTBKT$_{\text{MMOP}}$ algorithm is applied to solve a classic combinatorial optimization problem: FJSP [54], which involves multiple jobs and multiple machines, and each job has different optional operations on different machines. The objective of FJSP is to schedule all jobs on different machines at the best time to minimize the overall completion time.

In practical FJSP solving, various constraints and limitations often make the search space very large, pursuing only the global optimal solution may not be practical, as the solution may be infeasible or difficult to implement. While FJSP has a multimodal nature, which means that there may exist multiple equivalent local optimal solutions for different scheduling plans. Therefore, MTBKT$_{\text{MMOP}}$ algorithm aims to find as many local optimal solutions as possible. Here is the mathematical description of FJSP.

Let there be a jobs, b machines, each job A_t (t $\in$[1, a]) contains an operations set $K = \{K_{t,1}\ K_{t,2}, \ldots, K_{t,h}\}$, where $K_{t,k}$ is the kth operation of the job A_t, h is the number of operations, and $p_{t,k,y}$ is the processing time of operation $K_{t,k}$ on machine B_y ($y\in[1, b]$). Operations of each job must be processed in a specific sequence. The objective function is to minimize the makespan, which is defined as the time required to complete all the jobs. The mathematical formulation of FJSP can be expressed as:

Each operation is processed only once as

$$\sum_{t=1}^{a}\sum_{k=1}^{h}\sum_{y=1}^{b} q_{t,k,y} = 1 \tag{17}$$

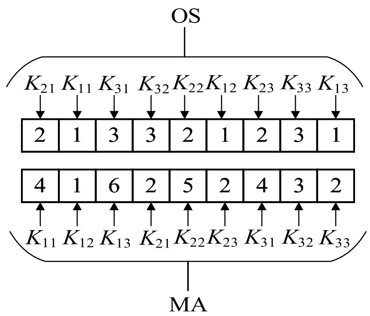

Fig. 8. Structure of chromosome bilayer encoding.

TABLE VI
MACHINES AND CORRESPONDING PROCESSING TIME OF EACH OPERATION, "--" INDICATES THE MACHINE IS NOT AVAILABLE

Job	Operation	$\boldsymbol{B_1}$	$\boldsymbol{B_2}$	$\boldsymbol{B_3}$	$\boldsymbol{B_4}$	$\boldsymbol{B_5}$	$\boldsymbol{B_6}$
$\boldsymbol{A_1}$	K_{11}	--	4	--	3	--	--
	K_{12}	5	--	2	5	--	--
	K_{13}	2	--	--	--	4	3
$\boldsymbol{A_2}$	K_{21}	--	4	5	--	--	--
	K_{22}	--	--	--	6	7	3
	K_{23}	--	3	--	8	--	2
$\boldsymbol{A_3}$	K_{31}	7	2	--	4	--	--
	K_{32}	1		5	--	--	3
	K_{33}	--	4	--	5	6	--

where $q_{t,k,y}$ is a binary variable indicating whether operation $K_{t,k}$ is assigned to machine y.

Each machine processes only one operation at a time as

$$\sum_{t=1}^{a}\sum_{y=1}^{b}\sum_{k_t=1}^{h} q_{t,k,y} \leq 1. \tag{18}$$

The makespan is the time required to complete all the jobs

$$C_{\max} = \text{Max}\left(\sum_{t=1}^{a}\sum_{y=1}^{b}\left(s_{t,k,y} + p_{t,k,y}\right)\right) \tag{19}$$

where $s_{t,k,y}$ denotes the start time of operation $K_{t,k}$ on machine y.

The chromosome utilized in this article is bilayer encoded and comprises machine assignment (MA) and operation sequence (OS). Specifically, there is a problem consists of 3 jobs (A_1, A_2, A_3) and 6 machines (B_1, B_2, B_3, B_4, B_5, B_6), where each job involves 3 operations A_1 (K_{11}, K_{12}, K_{13}); A_2 (K_{21}, K_{22}, K_{23}); A_3 (K_{11}, K_{12}, K_{13}) as listed in Table VI. The order of operations within each job is represented in the OS chromosome part, as illustrated in Fig. 8, where the first occurrence of the job index corresponds to the first operation of the job, the second occurrence to the second operation, and so on. In the MA chromosome part, machines are assigned to operations in ascending order of job index based on the available machine collections listed in Table VI. However, unlike the bilayer-encoded chromosome used in this article, the DE's individual consists of randomly generated numerical values for continuous problems. To represent the information on the individuals in MA, the range of gene values in DE is divided based on the number of optional machines for each operation. For example, for operation K_{11}, two machines (B_2 and B_4) are available. The search space [0, 1] is divided into two equally sized intervals: [0, 0.5] and [0.5, 1]. If the corresponding gene value on the DE chromosome falls within the interval [0, 0.5], select machine B_2; otherwise, select machine B_4.

TABLE VII
COMPARISON RESULTS OF $\text{MTBKT}_{\text{MMOP}}$ AND COMPARISON ALGORITHMS ON MAKESPAN

Problem	$\text{MTBKT}_{\text{MMOP}}$	CDE	SDE	NCDE
	(*mk*, *num*)	(*mk*, *num*)	(*mk*, *num*)	(*mk*, *num*)
Mk01	(**44**,37)	(48,2)	(46,3)	(46,3)
Mk02	(**41**,**3**)	(44,2)	(42,2)	(42,**3**)
Mk03	(**225**,**2**)	(242,**2**)	(251,**2**)	(235,**2**)
Mk04	(**71**,**27**)	(77,2)	(80,2)	(76,2)
Mk05	(**184**,**17**)	(192,2)	(197,2)	(191,2)

The MA chromosome and OS chromosome require different operators for crossover and mutation. For the OS chromosome, the precedence preserving order-based crossover (POX) [55] is utilized to generate feasible offspring without the need for a repair mechanism, which improves the decoding efficiency. Mutation for the OS chromosome involves randomly selecting two genes and exchanging their positions. The same crossover and mutation operators used in $\text{MTBKT}_{\text{MMOP}}$ are employed for the MA chromosome.

Five instances of Braindimarte's data set [56] are used to verify the performance of $\text{MTBKT}_{\text{MMOP}}$. Take the CDE, SDE, and NCDE as the comparison algorithms. The *NP* of each algorithm is set to 100, *MaxFes* of each algorithm is set to 5000, each algorithm is run 30 times independently, makespan (*mk*) and the highest number of the optimal solution (*num*) are calculated as the average over 30 times. Table VII presents a comparison between MTBKT_{MMOP} and three comparison algorithms, $\text{MTBKT}_{\text{MMOP}}$ achieves the best results in five instances, with the smallest *mk* and *num*.

VI. CONCLUSION

In this article, we introduce a novel framework that transforms MMOPs into multitask optimization problems. By applying neighborhood-based clustering, the population is partitioned into multiple species, with the assumption that each species encompasses one or more peaks of interest. Consequently, the search for peaks within each species can be treated as an independent subtask. We then propose two effective knowledge transfer mechanisms to facilitate inter-species information exchange by leveraging the differences between species centers. These mechanisms include explicit and implicit knowledge transfer, ensuring positive transfer through adaptive comparison of the fitness values associated with transferred information. This approach mitigates the risk of negative transfer, thereby enhancing the overall efficiency and robustness of the optimization process. Through an in-depth analysis of the underlying principles governing these knowledge transfer methods, we demonstrate that the cluster centers of each species progressively converge toward the vicinity of optimal solutions within their respective species, eventually stabilizing at these optima. Moreover, the interspecies communication via random species centers significantly enhances population diversity and mitigates the risk of premature convergence to local optima. In summary, our proposed $\mathrm{MTBKT}_{\mathrm{MMOP}}$ method offers a robust and innovative solution for addressing MMOPs within the domain of evolutionary computation. Additionally, it provides new application scenarios for multitask optimization techniques, thereby broadening the scope of their practical utility.

Specially, two information exchange strategies: E-KT and I-SCKT are proposed to enhance information exchange between species. E-KT operates by transferring the optimal individual from species with the fastest convergence to those species with slower convergence, thereby accelerating the convergence of the latter species. On the other hand, I-SCKT is an implicit knowledge transfer strategy that utilizes vector-based operations to facilitate information transfer between species, promoting effective exploration of the search space. Our proposed algorithm, $\mathrm{MTBKT}_{\mathrm{MMOP}}$, is compared with 13 state-of-the-art algorithms, and the results demonstrate that $\mathrm{MTBKT}_{\mathrm{MMOP}}$ performs well in problems with numerous optimal solutions or in high-dimensional problems. Additionally, $\mathrm{MTBKT}_{\mathrm{MMOP}}$ outperforms other comparison algorithms on most of the test functions of the CEC'2013 benchmark suite. In summary, knowledge transfer is an effective way to help the population locate global optimal solutions efficiently. Overall, the knowledge transfer of EMT can help the population effectively locate more global optimal solutions.

In the future, we plan to pursue two main research directions. First, there will be an emphasis on developing advanced solutions for MMOPs, particularly targeting more complex scenarios such as large-scale and computationally expensive MMOPs. Addressing these challenges requires innovative methodologies to efficiently identify multiple optima in high-dimensional and resource-intensive environments. Second, further exploration will be conducted into the application domains of multitask optimization techniques. The objective is to design more sophisticated knowledge transfer strategies to significantly enhance algorithmic performance. By refining these strategies, it is expected that the adaptability and robustness of optimization algorithms across a wide range of challenging problem domains will be markedly improved.

References

[1] D. Beasley, D. R. Bull, and R. R. Martin, "A sequential niche technique for multimodal function optimization," *Evol. Comput.*, vol. 1, no. 2, pp. 101–125, Jun. 1993.

[2] A. E. Eiben and J. E. Smith, "What is an evolutionary algorithm?" in *Introduction to Evolutionary Computing*. Berlin, Germany: Springer, 2015, pp. 25–48.

[3] A. Pétrowski, "A clearing procedure as a niching method for genetic algorithms," in *Proc. IEEE Int. Conf. Evol. Comput.*, 1996, pp. 798–803.

[4] B. Sareni and L. Krahenbuhl, "Fitness sharing and niching methods revisited," *IEEE Trans. Evol. Comput.*, vol. 2, no. 3, pp. 97–106, Sep. 1998.

[5] B. Qu, P. N. Suganthan, and J. J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 601–614, Oct. 2012.

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

[7] T. Wei, S. Wang, J. Zhong, D. Liu, and J. Zhang, "A review on evolutionary multitask optimization: Trends and challenges," *IEEE Trans. Evol. Comput.*, vol.26, no. 5, pp. 941–960, Oct. 2022.

[8] 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.

[9] M. Preuss, "Niching the CMA-ES via nearest-better clustering," in *Proc. ACM 12th Annu. Conf. Companion Genet. Evol. Comput.*, 2010, pp. 1711–1718.

[10] M. Preuss, *Multimodal Optimization by Means of Evolutionary Algorithms*. Cham, Switzerland: Springer, 2015.

[11] R. Storn and K. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, pp. 341–359, Dec. 1997.

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

[13] S. Singh, A. Tiwari, and S. Agrawal, "Differential evolution algorithm for multimodal optimization: A short survey," in *Proc. Int. Conf. Soft Comput. Probl. Solving*, 2021, pp. 745–756.

[14] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. 2nd Int. Conf. Genet. Algorithms*, 1987, pp. 41–49.

[15] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, 2004, pp. 1382–1389.

[16] W. Gao, G. G. Yen, and S. Liu, "A cluster-based differential evolution with self-adaptive strategy for multimodal optimization," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1314–1327, Aug. 2014.

[17] S. Biswas, S. Kundu, and S. Das, "An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1726–1737, Oct. 2014.

[18] S. Biswas, S. Kundu, and S. Das, "Inducing niching behavior in differential evolution through local information sharing," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 246–263, Apr. 2015.

[19] H. Zhao et al., "Local binary pattern-based adaptive differential evolution for multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3343–3357, Jul. 2020.

[20] Z.-J. Wang, Y.-R. Zhou, and J. Zhang, "Adaptive estimation distribution distributed differential evolution for multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6059–6070, Jul. 2022.

[21] Y. Jiang, Z.-H. Zhan, K. C. Tan, and J. Zhang, "Optimizing niche center for multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 53, no. 4, pp. 2544–2557, Apr. 2023.

[22] S. Hui and P. N. Suganthan, "Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 64–74, Jan. 2016.

[23] Y.-Z. Zhou, W.-C. Yi, L. Gao, and X.-Y. Li, "Adaptive differential evolution with sorting crossover rate for continuous optimization problems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2742–2753, Sep. 2017.

[24] 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.

[25] Z. Wei, W. Gao, G. Li, and Q. Zhang, "A penalty-based differential evolution for multimodal optimization," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6024–6033, Jul. 2022.

[26] X.-Y. Chen, H. Zhao, and J. Liu, "A network community-based differential evolution for multimodal optimization problems," *Inf. Sci.*, vol. 645, Oct. 2023, Art. no. 119359.

[27] X. Li, H. Zhao, and J. Liu, "Minimum spanning tree niching-based differential evolution with knowledge-driven update strategy for multimodal optimization problems," *Appl. Soft Comput.*, vol. 145, Sep. 2023, Art. no. 110589.

[28] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[29] L. Torrey and J. Shavlik, *Transfer Learning*. Hershey, PA, USA: IGI Glob., 2010.

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

[31] W. Du, Z. Ren, A. Chen, and H. Liu, "A knowledge transfer-based evolutionary algorithm for multimodal optimization," in *Proc. IEEE Congr. Evol. Comput.*, 2021, pp. 1953–1960.

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

[33] K. Qiao, K. Yu, B. Qu, J. Liang, H. Song, and C. Yue, "An evolutionary multitasking optimization framework for constrained multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 26, no. 2, pp. 263–277, Apr. 2022.

[34] L. Li, M. Xuan, Q. Lin, M. Jiang, Z. Ming, and K. C. Tan, "An evolutionary multitasking algorithm with multiple filtering for high-dimensional feature selection," *IEEE Trans. Evol. Comput.*, vol. 27, no. 4, pp. 802–816, Aug. 2023.

[35] C. Yang, Q. Chen, Z. Zhu, Z.-A. Huang, S. Lan, and L. Zhu, "Evolutionary multitasking for costly task offloading in mobile-edge computing networks," *IEEE Trans. Evol. Comput.*, vol. 28, no. 2, pp. 338–352, Apr. 2024.

[36] X. Zhou, Z. Wang, L. Feng, S. Liu, K.-C. Wong, and K. C. Tan, "Toward evolutionary multitask convolutional neural architecture search," *IEEE Trans. Evol. Comput.*, vol. 28, no. 3, pp. 682–695, Jun. 2024.

[37] R.-T. Liaw and Y.-W. Wen, "Ensemble learning through evolutionary multitasking: A formulation and case study," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 8, no. 4, pp. 3081–3094, Aug. 2024.

[38] Y. Huang, W. Zhou, Y. Wang, M. Li, L. Feng, and K. C. Tan, "Evolutionary multitasking with centralized learning for large-scale combinatorial multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 28, no. 5, pp. 1499–1513, Oct. 2024.

[39] K. Gao, C. Yang, J. Ding, K. C. Tan, and T. Chai, "Distributed knowledge transfer for evolutionary multitask multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 28, no. 4, pp. 1141–1155, Aug. 2024.

[40] V. Vassiiis and J.-B. Mouret. "Discovering the elite hypervolume by leveraging interspecies correlation," in *Proc. Genet. Evol. Comput.*, 2018, pp. 149–156.

[41] X. Lin, W. Luo, and P. Xu, "Differential evolution for multimodal optimization with species by nearest-better clustering," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 970–983, Feb. 2021.

[42] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization," Evol. Comput. Mach. Learn. Group, RMIT Univ., Melbourne, VIC, Australia, 2013. [Online]. Available: https://www.epi-tropakis.co.uk/cec16-niching/competition/

[43] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011.

[44] M. Zámková, M. Prokop, and R. Stolín, "Non-parametric anova methods applied on students' performance development in course of statistics," *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*, vol. 68, no. 1, pp. 281–289, 2020.

[45] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 387–402, Jun. 2013.

[46] Y. Wang, H.-X. Li, G. G. Yen, and W. Song, "MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 830–843, Apr. 2015.

[47] Q. Yang, W.-N. Chen, Y. Li, C. L. P. Chen, X.-M. Xu, and J. Zhang, "Multimodal estimation of distribution algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 636–650, Mar. 2017.

[48] Q. Yang et al., "Adaptive multimodal continuous ant colony optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 191–205, Apr. 2017.

[49] 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.

[50] J. E. Fieldsend, "Running up those hills: Multi-modal search with the niching migratory multi-swarm optimiser," in *Proc. IEEE. Congr. Evol. Comput.*, 2014, pp. 2593–2600.

[51] K.-C. Wong, K.-S. Leung, and M.-H. Wong, "Protein structure prediction on a lattice model via multimodal optimization techniques," in *Proc. 12th Annu. Genet. Evol. Comput. Conf.*, 2014, pp. 155–162.

[52] D.-K. Woo, J.-H. Choi, M. Ali, and H.-K. Jung, "A novel multimodal optimization algorithm applied to electromagnetic optimization," *IEEE Trans. Magn.*, vol. 47, no. 6, pp. 1667–1673, Jun. 2011.

[53] D.-Z. Tan, W.-N. Chen, J. Zhang, and W.-J. Yu, "Fast pedestrian detection using multimodal estimation of distribution algorithms," in *Proc. Genet. Evol. Comput. Conf.*, 2017, pp. 1248–1255.

[54] K. Gao, F. Yang, M. Zhou, Q. Pan, and P. N. Suganthan, "Flexible job-shop rescheduling for new job insertion by using discrete Jaya algorithm," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1944–1955, May 2019.

[55] K.-M. Lee, T. Yamakawa, and K.-M. Lee, "A genetic algorithm for general machine scheduling problems," in *Proc. Int. Conf. Knowl. Intell. Electr. Syst.*, 1998, pp. 60–66.

[56] B. Paolo, "Routing and scheduling in a flexible job shop by tabu search," *Ann. Oper. Res.*, vol. 41, no. 3, pp. 157–183, Sep. 1993.

Hong Zhao (Member, IEEE) received the bachelor's degree from the Kunming University of Science and Technology, Kunming, China, in 2017, and the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 2020.

She is currently an Associate Professor with the Guangzhou Institute of Technology, Xidian University, Guangzhou. Her research interests include artificial intelligence, evolutionary computation, swarm intelligence, and their applications in design and optimization.

Xu-Hui Ning received the bachelor's degree in space science and technology from Xidian University, Xi'an, China, in 2021, where he is currently pursuing the M.S. degree with the Guangzhou Institute of Technology.

His current research interests include machine learning, evolutionary computation, and their applications in real-world problems.

Jian-Yu Li (Member, IEEE) received the bachelor's degree and the Ph.D. degree in computer science and technology from the South China University of Technology, Guangzhou, China, in 2018 and 2022, respectively.

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

Dr. Li has been invited as a Reviewer of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS: SYSTEMS.

Jing Liu (Senior Member, IEEE) received the B.S. degree in computer science and technology and the Ph.D. degree in circuits and systems from Xidian University, Guangzhou, China, in 2000 and 2004, respectively.

In 2005, she joined Xidian University as a Lecturer, and was promoted to a Full Professor in 2009. From April 2007 to April 2008, she worked with The University of Queensland, Brisbane, QLD, Australia, as a Postdoctoral Research Fellow, and from July 2009 to July 2011, she worked with The University of New South Wales at the Australian Defence Force Academy, Canberra, ACT, Australia, as a Research Associate. She is currently a Full Professor with the Guangzhou Institute of Technology, Xidian University. Her research interests include evolutionary computation, complex networks, fuzzy cognitive maps, multiagent systems, and data mining.

Dr. Liu was an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION from 2015 to 2020. She has been the Chair of Emerging Technologies Technical Committee of IEEE Computational Intelligence Society from 2017 to 2018.