Adaptive Dimensional Learning for Multitask Optimization

Ze-Bin Wu[1], Zi-Jia Wang[1*], Hui-Ying Xu[2,3*], Xin-Zhong Zhu[2,3,4], Hong-Bo Li[4]

[1] *School of Computer Science and Cyber Engineering, Guangzhou University, 510006 Guangzhou, China.*

[2] *Zhejiang Key Laboratory of Intelligent Education Technology and Application, Zhejiang Normal University, Jinhua, 321004, Zhejiang, China.*

[3] *School of Computer Science and Technology, Zhejiang Normal University, Jinhua, Zhengjiang, China, 321004.*

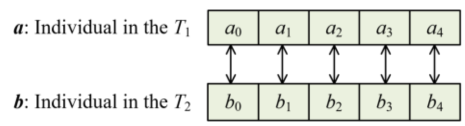
[4] *Beijing Geekplus Technology Co, Ltd, Beijing, 100101, China.*

Abstract*—Evolutionary multitask optimization (EMTO) is an emerging optimization approach that leverages knowledge transfer (KT) across multiple tasks to accelerate convergence and enhance solution quality. However, existing EMTO methods typically focus predominantly on index-aligned learning during self-evolution and KT, overlooking the potential of index-unaligned dimensions. Since the index-aligned dimensions differ in their representation or significance across tasks, strictly index-aligned learning may not be suitable, because it may lead to ineffective or even negative transfer. To address this and fully harness the potential of index-unaligned learning, this paper proposes an adaptive dimensional learning (ADL) for multitask optimization (ADLMTO), which utilizes both index-aligned and index-unaligned dimensional information to enhance optimization effectiveness by implementing evolutionary learning at the dimensional level. ADLMTO employs the adaptive feedback-driven evolutionary strategy (AFDES) to dynamically adjust evolutionary modes, including index-aligned and index-unaligned learning during both self-evolution and KT. ADL strategy performs effective dimensional-level learning based on evolutionary modes, thereby facilitating enhanced optimization. Experimental results demonstrate that ADLMTO outperforms state-of-the-art EMTO algorithms on the CEC2017 and CEC2022 multitask evolution benchmarks, confirming the effectiveness of ADLMTO.

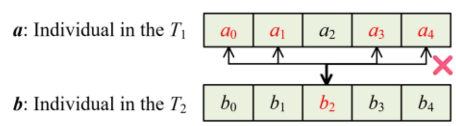
Keywords*—*Evolutionary multitask optimization (EMTO), adaptive dimensional learning (ADL), index-unaligned learning.

I. Introduction

EVOLUTIONARY computation (EC), inspired by natural evolution, include various techniques such as genetic algorithm (GA) [1], differential evolution (DE) [2], particle swarm optimization (PSO) [3], and ant colony optimization (ACO) [4], which have been extensively applied to solve complex optimization problems, including multimodal optimization [5], combinatorial optimization [6], multi-objective optimization [7], and large-scale optimization [8]. In recent years, with the rise of multitask optimization (MTO), evolutionary multitask optimization (EMTO) has become a key research focus. By facilitating knowledge transfer (KT) across multiple related tasks, EMTO significantly enhances convergence speed and solution quality, which demonstrates substantial potential in numerous real-world applications, including engineering design optimization [9] and resource allocation problems [10] , where multiple interconnected tasks frequently arise. As a result, EMTO not only enhances the theoretical foundations of evolutionary computation but also serves as a powerful instrument for tackling intricate issues in real-world applications.



(a) Illustration of the KT method based on index-aligned dimensions



(b) Limitation of the KT method based on index-aligned dimensions

Fig. 1. Illustration and limitation of the KT method based on index-aligned dimensions.

A central challenge in EMTO is enhancing the effectiveness of KT while mitigating ineffective or negative transfer. For instance, EMTO with adaptive intensity of KT (EMTO-AI) [11] and multitask evolutionary algorithm based on anomaly detection (MTEA-AD) [12] improve transfer efficiency by adaptively adjusting transfer frequency and selecting high-potential individuals. However, when implementing KT, these methods usually assume that the dimensions of individuals inter-task are strictly index-aligned, thereby only focus on the optimization characteristics of index-aligned dimensions and ignore the potential value of index-unaligned dimensions. When the alignment of dimensional information between tasks varies significantly, this assumption often breaks down, making it difficult for the algorithm to fully exploit inter-task differences and potentially leading to negative transfer. For example, when the dimensions of individual $\boldsymbol{a}$ from task T_1 and individual $\boldsymbol{b}$

Manuscript received…. revised…. accepted…. This work was supported in part by the National Natural Science Foundations of China (NSFC) under Grant 62106055 and 62376252, in part by the Guangdong Natural Science Foundation under Grants 2025A1515010256, in part by the Guangzhou Science and Technology Planning Project under Grants 2023A04J0388 and 2023A03J0662, and in part by the Tianjin Top Scientist Studio Project under Grant 24JRRCRC00030. *(Corresponding author: Zi-Jia Wang, Hui-Ying Xu.)*

Z. B. Wu and Z. J. Wang are with the School of Computer Science and Cyber Engineering, Guangzhou University, 510006 Guangzhou, China. (Email: zijiawang@gzhu.edu.cn)

H. Y. Xu and X. Z. Zhu are with the Zhejiang Key Laboratory of Intelligent Education Technology and Application, also with the School of Computer Science and Technology, Zhejiang Normal University, Jinhua, 321004, Zhejiang, China. (Email: xhy@zjnu.edu.cn)

X. Z. Zhu and H. B. Li are with the Beijing Geekplus Technology Co, Ltd, Beijing, 100101, China.

from task T_2 are highly dissimilar, index-aligned dimensional learning can only KT based on matching indices, potentially leading to ineffective or negative transfer, as shown in Fig. 1(a). In contrast, even when all dimensions of individual $\boldsymbol{a}$ from task T_1, except for the third dimension, are highly similar to the third dimension of individual $\boldsymbol{b}$ from task T_2, traditional methods fail to utilize this similarity for KT. Since these dimensions are not index-aligned, the opportunity for optimization is missed, as shown in Fig. 1(b).

To overcome this limitation, this paper proposes an adaptive dimensional learning (ADL) for multitask optimization (ADLMTO), which enhances EMTO through dimensional-level evolutionary learning. ADLMTO employs two key strategies: the adaptive feedback-driven evolutionary strategy (AFDES), which dynamically adjusts evolutionary modes and index-sequence based on performance feedback, and ADL strategy, which implements effective index-aligned and index-unaligned dimensional learning within and across tasks to facilitate optimization. Additionally, ADLMTO leverages the success-history based adaptive DE (SHADE) [13] operator to enhance search efficiency. In summary, the contributions of ADLMTO are summarized as follows:

1) AFDES dynamically adjusts evolutionary modes and index-sequence based on performance feedback, enabling more flexible and adaptive evolution, thereby improving optimization efficiency and convergence stability.

2) ADL strategy performs dimensional-level evolutionary learning by integrating index-aligned and index-unaligned dimensional information, reducing ineffective or negative transfer and enhancing MTO performance.

3) Extensive experiments on the CEC2017 and CEC2022 multitask optimization benchmarks demonstrate the superiority of ADLMTO, outperforming most state-of-the-art multitask evolutionary algorithms in terms of convergence speed and solution quality.

The rest of this paper is organized as follows. Section II reviews the progress of related studies in the EMTO domain; Section III elaborates on the design and implementation of the proposed ADLMTO framework. Section IV analyzes the experimental outcomes based on the widely adopted CEC2017 and CEC2022 EMTO benchmarks. Section V concludes the paper with conclusions.

II. Preliminary

A. Evolutionary Search Strategies

DE is a well-established global optimization method, celebrated for its ease of implementation and high performance. However, its performance is heavily influenced by the mutation strategy. Traditional strategies may struggle with complex problems, leading to premature convergence or slow progress, which has led to the development of adaptive DE variants [14]. SHADE introduces the current-to-*p*best/1 mutation operator, Combining the differences between the current individual and a subset of the best individuals with the perturbation from a pair of random individuals to balance convergence speed and population diversity. Additionally, SHADE adapts mutation factors and crossover probabilities, improving both performance and stability. The mutation formula for SHADE is:

$$\boldsymbol{v}_i = \boldsymbol{x}_i + F_i \times (\boldsymbol{x}_{pbest} - \boldsymbol{x}_i) + F_i \times (\boldsymbol{x}_{r1} - \boldsymbol{x}_{r2}) \qquad (1)$$

where $\boldsymbol{x}_i$ is the current individual, $\boldsymbol{x}_{pbest}$ is selected from the top p% individuals, $\boldsymbol{x}_{r1}$ is randomly chosen distinct individuals, and $\boldsymbol{x}_{r2}$ is arbitrarily chosen from the combined set of the current population and the archive. The parameter p is randomized to balance exploration and exploitation. An external archive further enhances diversity.

After the mutation vector $\boldsymbol{v}_i$ is generated, binomial crossover is used to generate the trial vector $\boldsymbol{u}_i$, as follows:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } \mathrm{rand}(0,1) \le CR \text{ or } j == j_{\mathrm{rand}} \\ x_{i,j}, & \text{otherwise} \end{cases} \qquad (2)$$

where CR is the crossover rate and j_{rand} is a random integer to make sure that $\boldsymbol{u}_i$ differs from $\boldsymbol{x}_i$ in at least one dimension.

B. EMTO

The EMTO algorithm is designed to address a multitask optimization problem (MTOP) involving K tasks. The primary objective of this algorithm is to find an optimal solution x_t for each task T_t (t = 1, 2, … , K) by minimizing the objective function $f_t(\boldsymbol{x}_t)$:

$$\boldsymbol{x}_t = \mathrm{argmin}(f_t(\boldsymbol{X}_t)) \text{ , } t = 1, 2, \ldots, K \qquad (3)$$

C. Related Work

EMTO has emerged as a noteworthy approach for addressing multiple tasks concurrently, attracting considerable interest in recent years. Its applications span diverse domains, such as planar kinematic arm control [15], and resource allocation problems [10]. A primary challenge in EMTO is to enhance the effectiveness of KT while mitigating the occurrence of ineffective or negative transfers.

Gupta *et al.* [16] introduced the first EMTO algorithm, known as the multifactorial evolutionary algorithm (MFEA). In MFEA, individuals associated with different skill factors engage in crossover at a random mating probability (*rmp*), enabling implicit KT. Following this work, extensive research has extended the MFEA framework, categorizing these approaches as single-population EMTO algorithms. For example, Zhou *et al.* [11] proposed EMTO-AI, an EMTO algorithm featuring an adaptive transfer intensity adjustment (ATIA) mechanism, which quantifies task relatedness using a similarity matrix and dynamically adjusts transfer frequency based on fitness-based population comparisons, while incorporating a knowledge archive to store successfully transferred individuals to enhance the effectiveness of KT. Liu *et al.* [17] proposed a new MFEA based on diffusion gradient descent (MFEA-DGD), integrating DGD into complementary crossover/mutation operators and a hyperrectangular search strategy to ensure population convergence and theoretically explain KT benefits through local task convexity analysis, thereby accelerating optimization in less-explored regions.

Additionally, several studies [18][19] adopt a multiple-population approach in EMTO, where each task is assigned a distinct population, and inter-task information exchange is achieved through explicit KT. For example, Li *et al.* [20] introduced a meta-knowledge transfer (MKT)-based DE (MKTDE), an algorithm that employs a generalized MKT strategy to dynamically evolve task-specific knowledge, augmented by a multiple-population framework within a unified search space and an elite solution transfer technique, to facilitate effective KT in multitask optimization. Wei *et al.* [21] proposed an adaptive transfer strategy based on the decision tree (EMT-ADT), leveraging a method that quantifies transfer ability to evaluate individuals and utilizes a decision tree to predict transferred individuals, thereby enhancing algorithm performance. Li *et al.* [22] proposed the knowledge-guided external sampling (KGxS) algorithm, which enhances KT in multitask evolution strategies (MTESs). By guiding distribution evolution through adaptive external samples, KGxS combines domain and shape knowledge with a universal boundary constraint handling method to address multitask optimization problems (MTOPs), effectively reducing negative transfer effects.

However, existing methods predominantly rely on strict index-aligned dimensional assumptions and overlook the potential of index-unaligned dimensions. To address these limitations, ADLMTO introduces AFDES for dynamically adjusting evolutionary modes and ADL strategy for adaptive dimensional-level knowledge learning, effectively enhancing KT while mitigating negative transfer.

III. ADLMTO

This section provides a comprehensive overview of the proposed ADLMTO. First, the motivation behind ADLMTO is presented. Next, we outline the ADL strategy and AFDES, while providing a detailed description of the search operators employed, respectively. Finally, we present the complete ADLMTO algorithm.

A. Motivation

A key challenge in EMTO is designing an effective KT mechanism for information exchange across tasks. Traditional methods often rely on index-aligned dimensions, which can lead to issues. For example, when dimensions from different tasks, like "joint torque" in robot control and "logistics cost" in supply chain optimization, are forcibly index-aligned, irrelevant knowledge is transferred, causing optimization bias. Additionally, when tasks evolve, such as a supply chain problem transitioning from static cost to dynamic pricing, index-aligned methods fail to adapt to changing dimensional semantics, hindering their ability to handle dynamic tasks. In contrast, index-unaligned dimensional learning can uncover hidden connections. For instance, even if the task dimensions have different indices, like "obstacle distance" in robot path planning and "battery consumption rate" in drone endurance optimization, both reflect "resource constraints" Such index-unaligned dimensions can share energy-saving strategies, enabling KT that traditional methods may overlook. However, reliance on index-aligned dimensional learning limits the ability to recognize these semantic relationships, leading to missed opportunities and reduced KT efficiency.

B. Adaptive Feedback-Driven Evolutionary Strategy

Conventional EMTO methods often rely on fixed index-aligned learning and manually tuned parameters (e.g., *rmp*) to control KT intensity, which struggle to handle inter-task heterogeneity and semantic dimensional discrepancies dynamically. Such static strategies may cause ineffective transfer or convergence stagnation. To establish a dynamically adjustable optimization mechanism, we formalize four key concepts.

Definition 1 (*Evolutionary Model*): The evolutionary model $\boldsymbol{M}$ is formulated as the Cartesian product of task-mode and dimensional-mode. Specifically, $\boldsymbol{M}$={(self-evolution, index-aligned), (self-evolution, index-unaligned), (KT, index-aligned), (KT, index-unaligned)}. Herein, task modes encompass self-evolution and KT. Meanwhile, dimensional modes are composed of index-aligned learning and index-unaligned learning.

Algorithm 1: AFDES

Input: The offspring population.
Output: Offspring with updated evolutionary mode and index sequence.
Begin
For each offspring $\boldsymbol{c}$ in offspring population:
  Get the Immediate Parent $\boldsymbol{p}$ of $\boldsymbol{c}$.
  Calculate the *IR* of $\boldsymbol{c}$ according to Eq. (3).
  If *IR* < 0:
    Inherit the evolutionary model and index sequence of $\boldsymbol{p}$ for $\boldsymbol{c}$.
  Else:
    Generate a new other evolutionary model and corresponding index sequence for $\boldsymbol{c}$.
  End If
End For
End

Definition 2 (*Index Sequence*): The index sequence is used as the operation carrier of dimensional learning, and its generation rules change dynamically with the alignment mode. In index-aligned learning, the sequence is strictly constrained to be a monotonically increasing arrangement of dimension indices. For example, when the total number of dimensions D=5, the sequence is constant to [1,2,3,4,5] to ensure the consistency of physical meaning between dimensions. In index-unaligned learning, the sequence is generated through repeatable random sampling, which is expressed as an unordered or repeated index combination, such as [1,5,4,3,2] or [1,5,5,3,3], so that the algorithm can capture the potential semantic associations between dimensions across tasks.

Definition 3 (*Immediate Parent*): In the SHADE mutation operator, the direct parent refers to the base individual $\boldsymbol{x}_i$.

Definition 4 (*Improvement Rate* (*IR*)): The *IR* quantifies the performance improvement of an offspring compared to its immediate parent, calculated as follows:

$$IR = \frac{f_o - f_p}{\left| f_p \right|} \tag{4}$$

where f_o refers to the fitness of the offspring generated and f_p refers to the fitness of its immediate parent.

Based on this, this paper proposes AFDES, which dynamically adjusts the evolution mode and index sequence through real-time feedback of offspring fitness, thereby adaptively controlling the evolutionary modes.

The core mechanism of AFDES is based on the *IR*, which dynamically adjusts the evolutionary mode and index sequence. For minimization problems, *IR* is calculated using Eq. (3). If *IR* < 0, it indicates that the offspring outperforms the parent, so the current evolutionary model and index sequence should be retained to continue searching in the promising direction. Conversely, if *IR* ⩾ 0, it suggests that the current strategy is ineffective, prompting the offspring to regenerate using a different evolutionary model and index sequence from the parent to enhance search exploration. For example, in a human hand control task, if the KT (index-aligned) mode adopts the index sequence [1,2,3,4,5], and the offspring's *IR* is -0.2 (indicating a 20% improvement), the offspring inherits the mode and sequence, maintaining the KT effect. However, if *IR* is 0.2 (indicating a 20% decline), the algorithm may switch to the self-

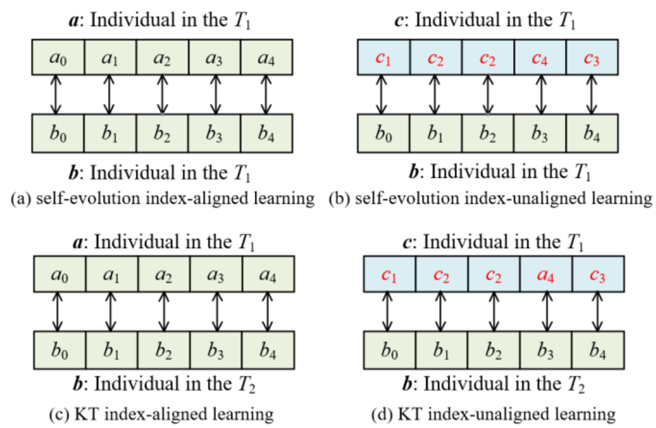


Fig. 2. Illustration of four evolutionary models.

evolution (index-unaligned) mode, generating a new sequence [3,1,5,2,4] to explore a more diverse search path through perturbation in high-degree dimensions.

The pseudocode of AFDES is shown in **Algorithm 1**. First, for every individual $\boldsymbol{c}$ within the offspring population, its immediate parent $\boldsymbol{p}$ is identified (lines 1-2). Second, the IR of $\boldsymbol{c}$ is calculated according to Eq. (3) (line 3). Finally, if $IR < 0$, indicating that $\boldsymbol{c}$ outperforms $\boldsymbol{p}$, it inherits the evolutionary model and index sequence from $\boldsymbol{p}$ to maintain a promising search direction (lines 4-5). Otherwise, if $IR \geqslant 0$, meaning the current strategy is ineffective, a new evolutionary model and corresponding index sequence are generated to enhance search exploration (lines 6-7).

C. Adaptive Dimensional Learning Strategy

Following the adaptive adjustment of the evolutionary model and index sequence by the AFDES, ADL strategy implements dimension-level learning through four distinct modes: index-aligned and index-unaligned learning for self-evolution, as well as index-aligned and index-unaligned learning for KT, as illustrated in Fig. 2. This dual-mode approach not only facilitates efficient intra-task evolution but also enhances inter-task KT, ultimately enhancing the overall efficacy of evolutionary algorithms within the EMTO framework.

For self-evolution, within the same task, the index-aligned of different individuals usually share the same semantics, as shown in Fig. 2(a). Using index-aligned learning can ensure that these dimensions evolve along a consistent optimization direction, thereby maintaining the stability of the evolutionary process. However, in practical problems, even if the dimension indexes are different (for example, the 1th dimension and the 2th dimension), they may contain similar semantic information. Imagine that in a certain evolutionary process, the 2th dimension of individual $\boldsymbol{c}$ first obtains a better fitness value, while the 1th dimension of individual $\boldsymbol{b}$ performs poorly due to the poor initial search direction. In this case, the index-unaligned learning strategy shown in Fig. 2(b) can flexibly match the potential semantic similarity between the two dimensions, and pass the high-quality optimization information in the 2th dimension to the poorly performing 1th dimension, thereby accelerating its performance improvement, and ultimately enhancing the overall convergence speed and solution accuracy. For KT, which main goal is to accelerate convergence and improve optimization quality by sharing useful knowledge across tasks. Tasks may have highly similar dimensions, whether or not they are index-aligned. For instance, Task T_1's 2th dimension is highly similar to Task T_2's 1th and 2th dimensions. Index-aligned KT (as shown in Fig. 2(c)) ensures stable KT for the 2th dimension, enhancing the optimization. However, for the 1th and 2th dimensions across tasks, index-unaligned KT (as shown in Fig. 2(d)) can capture the semantic similarity and further improve knowledge sharing and optimization. In summary, ADL allows the algorithm to adaptively switch between index-aligned and index-unaligned learning, fully exploiting the potential of both intra-task and inter-task KT to accelerate convergence and improve solution quality.

Algorithm 2: ADLMTO

```
Input: NP: The population size.
       p: Greedy coefficient.
Output: The optimal solution of each task.
Begin
1:  Randomly initialize k populations for k tasks.
2:  Randomly initialize the evolution model and corresponding index
    sequence of each individual.
3:  FEs = 0. g = 0.
4:  Evaluate each individual of k populations and update FEs.
5:  While FEs ≤ MaxFEs:
6:    For m in {1, 2, …, k}:
7:      For each individual i_m in pop_m:
8:        If Evolutionary model is self-evolution:
9:          Randomly select an individual from the top p% of the
            target population based on fitness as x_pbest.
10:       Else:
11:         Randomly select an individual from the top p% of the
            auxiliary population based on fitness as x_pbest.
12:       End If
13:       Generate offspring c according to Eq. (5) and Eq. (2).
14:       Evaluate c, FEs += 1.
15:     End For
16:     Update the offspring evolution model and index sequence
        according to Algorithm 1.
17:     Execute elite selection strategy.
18:   End For
19:   g += 1.
20: End While
21: Return the optimal solution of each task.
End
```

In addition, in order to use dimension-level knowledge learning in the EMTO problem, we improve the mutation operator in SHADE, and is redefined as follows:

$$\boldsymbol{v}_i = \boldsymbol{x}_i + F_i \times (\boldsymbol{x}_{pbest}[\boldsymbol{I}] - \boldsymbol{x}_i) + F_i \times (\boldsymbol{x}_{r1} - \boldsymbol{x}_{r2}) \quad (5)$$

where $\boldsymbol{I}$ represents the index sequence of $\boldsymbol{x}_{pbest}$ individuals, when the evolutionary model is self-evolution, $\boldsymbol{x}_{pbest}$ is randomly selected from the top p% individuals in the target population based on fitness. When the evolutionary model is KT, $\boldsymbol{x}_{pbest}$ is randomly selected from the top p% individuals in the auxiliary population based on fitness. $\boldsymbol{x}_{pbest}[\boldsymbol{I}] - \boldsymbol{x}_i$ accelerates local exploitation via elite individuals and fosters dimensional learning, while $\boldsymbol{x}_{r1} - \boldsymbol{x}_{r2}$ boosts global search diversity.

D. Framework of ADLMTO

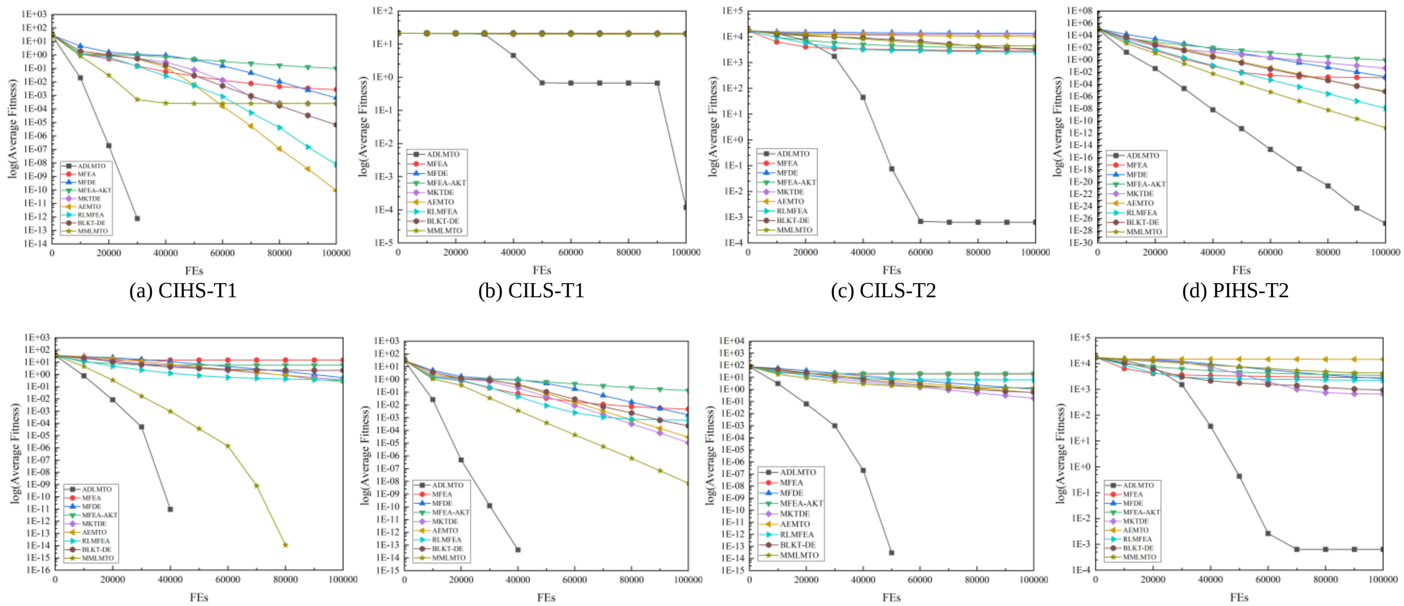
The overall framework of ADLMTO is outlined in **Algorithm 2**. Specifically:

Step 1: Initialize populations for all tasks with random evolutionary models and index sequences, evaluate initial fitness, and set parameters , as shown in lines 1-4 of **Algorithm 2**.

Step 2: Select $\boldsymbol{x}_{pbest}$ according to the evolutionary model, as shown in lines 8-12 of **Algorithm 2**.

TABLE I EXPERIMENTAL RESULTS ON CEC2017 BENCHMARK BETWEEN ADLMTO AND OTHER EMTO ALGORITHMS

Problem	ADLMTO	MFEA	MFDE	MFEA-AKT	MKTDE	AEMTO	RLMFEA	BLKT-DE	MMLMTO
CIHS-T1	**0**	2.68E−3(+)	6.53E−4(+)	9.80E−2(+)	2.47E−4(+)	8.93E−11(+)	7.45E−9(+)	6.53E−6(+)	2.47E−4(≈)
CIHS-T2	**0**	9.17E+1(+)	2.77E+0(+)	1.78E+2(+)	1.46E+0(+)	1.60E−7(+)	1.06E−5(+)	8.55E+1(+)	8.29E−1(≈)
CIMS-T1	**4.44E−16**	1.12E+0(+)	1.44E−2(+)	4.85E+0(+)	1.77E−8(+)	8.24E−8(+)	2.02E−6(+)	4.93E−4(+)	2.27E−12(+)
CIMS-T2	1.26E+0	6.41E+1(+)	1.25E−1(+)	2.30E+2(+)	8.37E−13(+)	4.75E−12(+)	4.71E−9(+)	9.17E+1(+)	**0(≈)**
CILS-T1	**1.18E−4**	2.00E+1(+)	2.12E+1(+)	2.01E+1(+)	2.12E+1(+)	2.12E+1(+)	2.00E+1(+)	2.07E+1(+)	2.01E+1(+)
CILS-T2	**6.36E−4**	2.78E+3(+)	1.35E+4(+)	3.46E+3(+)	1.18E+4(+)	1.02E+4(+)	2.53E+3(+)	3.21E+3(+)	4.40E+3(+)
PIHS-T1	3.91E+1	2.35E+2(+)	7.32E+1(+)	5.08E+2(+)	3.41E+2(+)	3.87E+2(+)	1.23E+2(+)	8.40E+1(+)	**3.52E−12(−)**
PIHS-T2	**1.63E−27**	1.19E−3(+)	1.72E−3(+)	8.59E−1(+)	4.12E−2(+)	5.00E−6(+)	1.14E−8(+)	6.48E−6(+)	7.40E−12(+)
PIMS-T1	**3.85E−2**	4.76E−1(+)	2.23E−2(+)	2.95E+0(+)	8.61E−4(+)	1.96E−4(+)	4.70E−2(+)	5.26E−4(+)	2.44E−7(+)
PIMS-T2	**4.05E+1**	9.50E+1(+)	5.08E+1(+)	2.28E+2(+)	5.77E+1(+)	8.41E+1(+)	8.16E+1(+)	4.83E+1(+)	4.80E+1(+)
PILS-T1	**4.44E−16**	1.57E+1(+)	5.64E−1(+)	5.28E+0(+)	1.69E+0(+)	4.20E−2(+)	2.57E−1(+)	1.04E−2(+)	3.64E−15(≈)
PILS-T2	**0**	1.47E+1(+)	5.47E−1(+)	5.62E+0(+)	3.56E−1(+)	2.35E−1(+)	3.24E−1(+)	2.14E+0(+)	0(≈)
NIHS-T1	**4.07E+1**	1.18E+2(+)	4.77E+1(+)	3.03E+2(+)	4.69E+1(+)	4.59E+1(+)	8.29E+1(+)	5.00E+1(+)	4.63E+1(+)
NIHS-T2	**6.87E+0**	1.16E+2(+)	1.03E+0(+)	2.34E+2(+)	6.75E−2(+)	7.88E−5(+)	5.64E+1(+)	9.06E+1(+)	1.43E+0(+)
NIMS-T1	**0**	4.60E−3(+)	1.61E−3(+)	1.32E−1(+)	1.09E−5(+)	2.87E−5(+)	6.35E−4(+)	2.26E−4(+)	7.18E−9(+)
NIMS-T2	**0**	1.92E+1(+)	1.06E+0(+)	2.06E+1(+)	1.87E−1(+)	5.40E−1(+)	6.36E+0(+)	5.61E−1(+)	1.35E+0(+)
NILS-T1	**4.05E+1**	2.39E+2(+)	4.00E+2(+)	6.75E+2(+)	3.73E+2(+)	3.91E+2(+)	1.50E+2(+)	1.10E+2(+)	8.15E+1(+)
NILS-T2	**6.36E−4**	2.82E+3(+)	2.64E+3(+)	3.46E+3(+)	6.54E+2(+)	1.45E+4(+)	2.25E+3(+)	9.40E+2(+)	4.32E+3(+)
Number of + / ≈ / −		18 / 0 / 0	18 / 0 / 0	18 / 0 / 0	18 / 0 / 0	18 / 0 / 0	18 / 0 / 0	18 / 0 / 0	12 / 5 / 1



(a) CIHS-T1 (b) CILS-T1 (c) CILS-T2 (d) PIHS-T2
(e) PILS-T2 (f) NIMS-T1 (g) NIMS-T2 (h) NILS-T2

Fig. 3. Convergence curves of the average fitness on several tasks from the CEC2017 benchmark.

Step 3: Generate offspring based on ADL strategy, as shown in line 13 of **Algorithm 2**.

Step 4: Evaluate offspring fitness and update the function evaluation counter, as shown in line 14 of **Algorithm 2**.

Step 5: Update the offspring evolution model and index sequence based on AFDES, and update the population through elite selection, as shown in lines 16-17 of **Algorithm 2**.

The whole process repeats until the maximum number of function evaluations (*MaxFEs*) is used up.

IV. EXPERIMENTAL STUDIES

In this study, the performance of ADLMTO is evaluated on two widely recognized EMTO benchmarks, CEC2017 [23] and CEC2022 [24]. To ensure a comprehensive and rigorous comparison, eight state-of-the-art EMTO algorithms are selected: MFEA [16] (2016), MFDE [25] (2017), MFEA-AKT [26] (2021), MKTDE [27] (2022), AEMTO [28] (2022), RLMFEA [29] (2024), BLKT-DE [29] (2024), and MMLMTO [30] (2024). Covering a span from 2016 to 2024, these algorithms provide a solid foundation for assessing the effectiveness and competitiveness of ADLMTO.

A. Parameter Settings

The parameter settings for the ADLMTO are shown as follows:

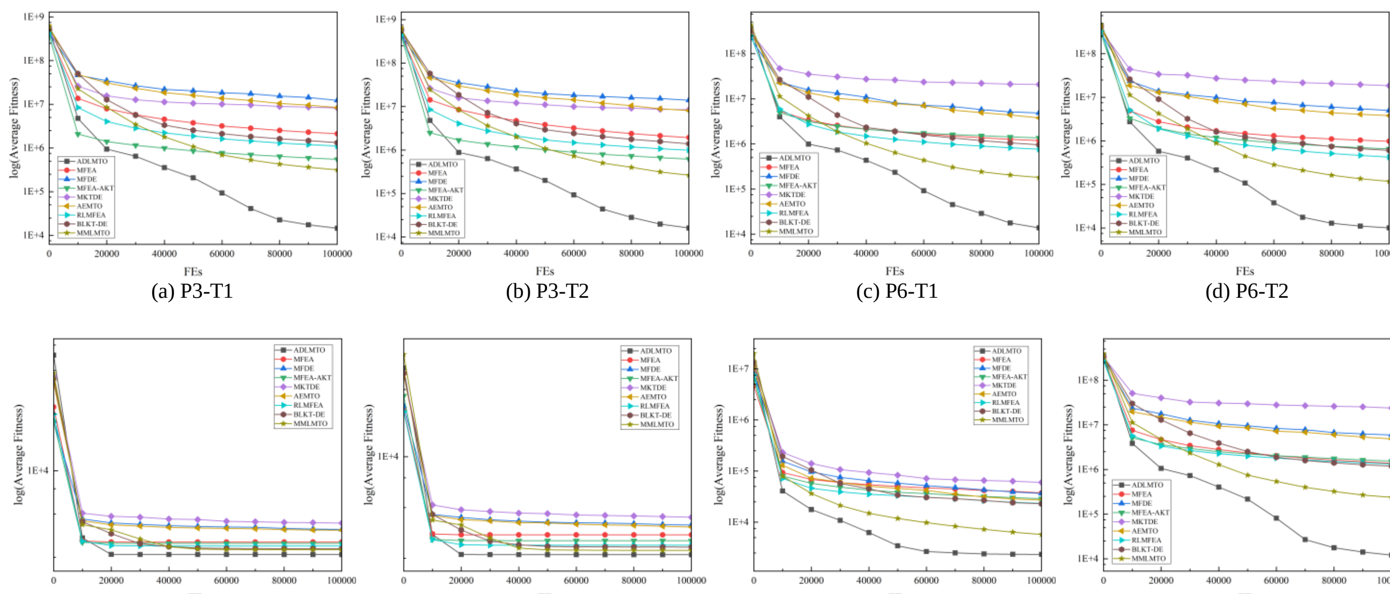
1) Population size in ADLMTO: $NP = 100$ for each task.

2) Greedy coefficient in SHADE: $p = 0.2$ for each task.

To ensure a fair evaluation, the *MaxFEs* is uniformly set to 200,000 for all algorithms, while other parameters follow their original configurations. Every algorithm is run separately 30 times, with the average outcomes documented. The Wilcoxon rank-sum test ($\alpha = 0.05$) is applied to evaluate the statistical significance of the results. The symbols "+ / ≈ / −" indicate whether ADLMTO outperforms, performs comparably to, or

TABLE II EXPERIMENTAL RESULTS ON CEC2022 BENCHMARK BETWEEN ADLMTO AND OTHER EMTO ALGORITHMS

Problem	**ADLMTO**	**MFEA**	**MFDE**	**MFEA-AKT**	**MKTDE**	**AEMTO**	**RLMFEA**	**BLKT-DE**	**MMLMTO**
P1-T1	6.05E+2	6.45E+2(+)	6.10E+2(+)	6.26E+2(+)	6.02E+2(−)	6.02E+2(−)	6.15E+2(+)	6.07E+2(+)	**6.01E+2(−)**
P1-T2	6.05E+2	6.45E+2(+)	6.10E+2(+)	6.27E+2(+)	6.02E+2(−)	6.02E+2(−)	6.15E+2(+)	6.09E+2(+)	**6.02E+2(−)**
P2-T1	7.00E+2	7.00E+2(+)	7.00E+2(+)	7.01E+2(+)	7.00E+2(≈)	7.00E+2(≈)	7.00E+2(≈)	7.00E+2(≈)	**7.00E+2(≈)**
P2-T2	7.00E+2	7.00E+2(≈)	7.00E+2(+)	7.01E+2(+)	7.00E+2(≈)	7.00E+2(≈)	7.00E+2(≈)	7.00E+2(≈)	**7.00E+2(≈)**
P3-T1	**1.46E+4**	2.12E+6(+)	1.23E+7(+)	5.48E+5(+)	8.24E+6(+)	8.42E+6(+)	1.11E+6(+)	1.32E+6(+)	3.15E+5(+)
P3-T2	**1.61E+4**	1.93E+6(+)	1.40E+7(+)	6.18E+5(+)	8.45E+6(+)	8.07E+6(+)	9.94E+5(+)	1.39E+6(+)	2.64E+5(+)
P4-T1	**1.30E+3**	1.30E+3(+)	1.30E+3(+)	1.30E+3(+)	1.30E+3(+)	1.30E+3(+)	1.30E+3(+)	1.30E+3(+)	1.30E+3(+)
P4-T2	**1.30E+3**	1.30E+3(+)	1.30E+3(+)	1.30E+3(+)	1.30E+3(+)	1.30E+3(+)	1.30E+3(+)	1.30E+3(+)	1.30E+3(+)
P5-T1	**1.51E+3**	1.53E+3(+)	1.53E+3(+)	1.55E+3(+)	1.53E+3(+)	1.53E+3(+)	1.51E+3(+)	1.53E+3(+)	1.51E+3(≈)
P5-T2	**1.51E+3**	1.53E+3(+)	1.53E+3(+)	1.56E+3(+)	1.53E+3(+)	1.53E+3(+)	1.51E+3(+)	1.53E+3(+)	1.51E+3(≈)
P6-T1	**1.39E+4**	1.19E+6(+)	4.86E+6(+)	1.36E+6(+)	2.08E+7(+)	3.79E+6(+)	7.65E+5(+)	9.60E+5(+)	1.81E+5(+)
P6-T2	**1.01E+4**	9.78E+5(+)	4.98E+6(+)	6.57E+5(+)	1.86E+7(+)	3.82E+6(+)	4.25E+5(+)	6.60E+5(+)	1.17E+5(+)
P7-T1	**2.61E+3**	3.19E+3(+)	3.90E+3(+)	3.12E+3(+)	4.32E+3(+)	3.86E+3(+)	2.99E+3(+)	2.87E+3(+)	2.82E+3(+)
P7-T2	**2.63E+3**	3.44E+3(+)	3.94E+3(+)	3.18E+3(+)	4.38E+3(+)	3.83E+3(+)	2.99E+3(+)	2.92E+3(+)	2.79E+3(+)
P8-T1	5.20E+2	5.20E+2(−)	5.21E+2(+)	5.20E+2(−)	5.21E+2(+)	5.21E+2(+)	5.21E+2(+)	5.21E+2(+)	**5.20E+2(−)**
P8-T2	5.20E+2	5.20E+2(−)	5.21E+2(+)	5.20E+2(−)	5.21E+2(+)	5.21E+2(+)	5.21E+2(+)	5.21E+2(+)	**5.20E+2(−)**
P9-T1	7.35E+3	8.27E+3(+)	1.47E+4(+)	8.09E+3(+)	1.47E+4(+)	1.47E+4(+)	**7.24E+3(≈)**	7.95E+3(+)	7.92E+3(+)
P9-T2	**1.62E+3**	1.62E+3(+)	1.62E+3(+)	1.62E+3(+)	1.62E+3(+)	1.62E+3(+)	1.62E+3(+)	1.62E+3(+)	1.62E+3(+)
P10-T1	**2.34E+3**	3.73E+4(+)	3.65E+4(+)	2.88E+4(+)	6.01E+4(+)	2.71E+4(+)	2.33E+4(+)	2.29E+4(+)	5.75E+3(+)
P10-T2	**1.21E+4**	1.38E+6(+)	5.93E+6(+)	1.54E+6(+)	2.38E+7(+)	4.85E+6(+)	1.32E+6(+)	1.19E+6(+)	2.38E+5(+)
Number of + / ≈ / −		17 / 1 / 2	20 / 0 / 0	18 / 0 / 2	16 / 2 / 2	16 / 2 / 2	17 / 3 / 0	18 / 2 / 0	12 / 4 / 4



(a) P3-T1 (b) P3-T2 (c) P6-T1 (d) P6-T2

(e) P7-T1 (f) P7-T2 (g) P10-T1 (h) P10-T2

Fig. 4. Convergence curves of the average fitness on several tasks from CEC2022 benchmark.

underperforms compared algorithms, respectively. The optimal outcome for each task is emphasized in bold text.

B. Experimental Results on CEC2017 EMTO Benchmark

Table I presents the experimental results of ADLMTO alongside state-of-the-art EMTO algorithms on the CEC2017 benchmark suite. ADLMTO achieves the best performance in 16 out of 18 tasks, underscoring its exceptional efficacy. Notably, it consistently outperforms MFEA, MFDE, MFEA-AKT, MKTDE, AEMTO, and RLMFEA. These competing algorithms predominantly rely on index-aligned dimension learning, ADLMTO harnesses index-unaligned dimension learning to fully exploit the diverse dimensional information within and across tasks, thereby ensuring the full utilization of high-quality dimensional information. Furthermore, when compared to BLKT-DE and MMLMTO, which also leverage index-unaligned dimension learning, ADLMTO consistently outperforms BLKT-DE across all 18 tasks and surpasses MMLMTO in 12 tasks. In five tasks—namely, CIHS-T1, CIHS-T2, CIMS-T2, PILS-T1, and PILS-T2—ADLMTO and MMLMTO exhibit comparable performance, both approaching the global optimum, with ADLMTO showing only a marginal underperformance on the PIHS-T1 problem. This pronounced advantage is attributed to ADLMTO's AFDES and ADL strategy, further enhanced by the exceptional capability of the SHADE operator, which optimizes search trajectories using historical success experiences, enabling efficient balancing of exploration and exploitation in complex task scenarios.

To gain deeper insights into the evolutionary dynamics of ADLMTO and other EMTO algorithms on the CEC2017 benchmark, their convergence trajectories are visualized in Fig. 3. Firstly, as illustrated in Figs. 3(a), 3(f), and 3(g), ADLMTO is the only algorithm that attains the global optimum (i.e., 0) in the CIHS-T1, NIMS-T1, and NIMS-T2 tasks. Moreover, ADLMTO demonstrates remarkably fast convergence, requiring approximately 30% of the MaxFEs in CIHS-T1, 40% in NIMS-T1, and 50% in NIMS-T2, substantially outperforming other algorithms. Secondly, Figs 3(b), 3(c), and 3(h) present the convergence curves for the CILS-T1, CILS-T2, and NIHS-T2 tasks. As observed, during the early optimization phase, ADLMTO performs comparably to other algorithms (e.g., MFEA, MFDE), exhibiting a similar convergence trend. However, in the later phase, ADLMTO demonstrates a clear advantage by successfully escaping local optima and achieving superior results, whereas other algorithms remain trapped in local optima, resulting in suboptimal convergence. Finally, on the PIHS-T2 and PILS-T2 tasks, as shown in Figs. 3(d) and (e), ADLMTO demonstrates significantly superior solution quality compared to other algorithms, ultimately achieving satisfactory results. Specifically, in the PILS-T2 task, although MMLMTO also attains the global optimum (i.e., 0), ADLMTO exhibits a faster convergence rate, approaching the optimal solution noticeably earlier, while other algorithms perform relatively poorly.

In summary, ADLMTO outperforms other leading EMTO algorithms on the CEC2017 benchmark, exhibiting superior performance and highlighting the effectiveness of dimension learning with index-unaligned.

C. Experimental Results on CEC2022 EMTO Benchmark

Table II presents the results of ADLMTO and other EMTO algorithms on the CEC2022 benchmark. From Table II, ADLMTO achieves the best results on 13 out of 20 tasks, while other algorithms can only achieve the best results in a maximum of 6 tasks, which demonstrates the superiority of ADLMTO. Compared to the EMTO algorithms that use KT methods based on index-aligned dimensions (including MFEA, MFDE, MFEA-AKT, MKTDE, AEMTO, and RLMFEA), ADLMTO outperforms them on 17, 20, 18, 16, 16, and 17 tasks, respectively, and only underperforms them on 2, 0, 2, 2, 2, and 0 tasks. Although BLKT-DE and MMLMTO utilize index-unaligned dimensional learning, ADLMTO consistently demonstrates superior performance. Specifically, it outperforms BLKT-DE on 18 tasks, matches performance on 2 tasks, and underperforms on none. Similarly, it surpasses MMLMTO on 12 tasks, achieves comparable results on 4 tasks, and underperforms on only 4 tasks. These results further highlight the effectiveness of AFDES in adaptively adjusting evolutionary modes and the ADL strategy in efficiently leveraging index-unaligned dimensional information, ensuring more effective KT and improved optimization performance.

To gain deeper insights into the evolutionary dynamics of ADLMTO and other EMTO algorithms on the CEC2022 benchmark, their convergence trajectories are visualized in Fig. 4. Firstly, as shown in Figs. 4(a) and (b), although ADLMTO exhibits a slightly slower convergence rate than MFEA-AKT in the early stages, it progressively demonstrates a substantial advantage in the mid-to-late evolutionary phases, ultimately achieving superior results compared to MFEA-AKT and other algorithms. Secondly, the P6-T1, P6-T2, P10-T1 and P10-T2 tasks, as shown in Figs. 4(c), (d), (g), and (h), ADLMTO consistently outperforms other algorithms throughout the optimization process, with its convergence curve steadily declining, reflecting superior optimization capability, and ultimately achieving satisfactory results across these tasks. Finally, on the P7-T1 and P7-T2 tasks, as shown in Figs. 3(d) and (e), while the performance of most algorithms remains similar, ADLMTO reliably delivers more precise results compared to other EMTO algorithms.

In summary, ADLMTO outperforms other leading EMTO algorithms on the CEC2022 benchmark, exhibiting superior performance and highlighting the effectiveness of dimension learning with index-unaligned.

V. Conclusion

Existing EMTO methods focus on index-aligned dimensional learning, assuming shared semantic meanings across tasks, which overlooks index-unaligned dimensions, leading to ineffective or negative transfer when task semantics vary. To address this and fully harness the potential of index-unaligned learning, we propose ADLMTO, which incorporates two key strategies. AFDES adaptively adjusts evolutionary modes based on performance feedback, enhancing flexibility and stability in the optimization process. ADL strategy performs dimensional-level learning by integrating both index-aligned and index-unaligned dimensions, effectively reducing negative transfer and improving overall optimization performance. Extensive experiments on the CEC2017 and CEC2022 multitask optimization benchmarks demonstrate the superiority of ADLMTO, outperforming most state-of-the-art multitask evolutionary algorithms in terms of convergence speed and solution quality.

ADLMTO shows promising results, but it may face challenges in large-scale problems where the ADL strategy may not perform as effectively. Future research could focus on improving the strategy's scalability in high-dimensional spaces, such as in large-scale feature selection [31]. Additionally, enhancing computational efficiency for larger task sets, like in multi-objective optimization [32], could further optimize its practical applications.

References

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

[2] 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.

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

[4] H. Heng and W. Rahiman, “ACO-GA-Based optimization to enhance global path planning for autonomous navigation in grid environments,” in *IEEE Trans. Evol. Comput*., doi: 10.1109/TEVC.2025.

[5] G. -Y. Lin et al., “A landscape-aware differential evolution for multimodal optimization problems,” in *IEEE Trans. Evol. Comput*., doi: 10.1109/TEVC.2025.

[6] D. Liang, Z. H. Zhan, Y. Zhang, and J. Zhang, “An efficient ant colony system approach for new energy vehicle dispatch problem,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4784–4797, Nov. 2020.

[7] J. J. Lin, S. C. Huang, and M. K. Jiau, “An evolutionary multiobjective Carpool algorithm using set-based operator based on simulated binary crossover,” *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3432–3442, Sep. 2019.

[8] W. Liu, Y. Zhang, K. Liu, B. Quinn, X. Yang and Q. Peng, “Evolutionary multiobjective optimization for large-scale portfolio selection with both

random and uncertain returns," in *IEEE Trans. Evol. Comput*., vol. 29, no. 1, pp. 76-90, Feb. 2025.

[9] J. Liang *et al.*, "Evolutionary multi-task optimization for parameters extraction of photovoltaic models," *Energy Convers. Manag*., vol. 207, pp. 112509–112514, Mar. 2020.

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

[11] X. Zhou, N. Mei, M. Zhong and M. Wang, "Evolutionary multi-task optimization with adaptive intensity of knowledge transfer," in *IEEE Trans. Emerg. Top. Comput. Intell*., doi: 10.1109/TETCI.2024.

[12] 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.

[13] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," *Proc. IEEE Congr. Evol. Comput. (CEC), Cancun*, Mexico, 2013.

[14] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Tran. Evol. Comput*., vol. 15, no. 1, pp. 55–66, 2011.

[15] J. B. Mouret and G. Maguire, "Quality diversity for multitask optimization," *Proc. Genet. Evol. Comput. Conf.*, 2020, pp. 121–129.

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

[17] Z. Liu, G. Li, H. Zhang, Z. Liang, and Z. Zhu, "Multifactorial evolutionary algorithm based on diffusion gradient descent," *IEEE Trans. Cybern.*, pp. 1–13, 2024.

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

[19] X. Wang, Q. Kang, M. Zhou, S. Yao and A. Abusorrah, "Domain adaptation multitask optimization," *IEEE Trans. Cybern.*, vol. 53, no. 7, pp. 4567-4578, July 2023.

[20] 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.

[21] Li, W., Gao, X. & Wang, L. "Multifactorial evolutionary algorithm with adaptive transfer strategy based on decision tree," *Complex Intell. Syst*. 9, 6697–6728 (2023).

[22] Y. Li, W. Gong and S. Li, "Multitask Evolution Strategy With Knowledge-Guided External Sampling," in *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 6, pp. 1733-1745, Dec. 2024.

[23] B. S. Da *et al.*, "Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metrics and baseline results," *School Comput. Sci. Eng.*, Nanyang Technol. Univ., Singapore, Rep., 2016.

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

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

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

[27] 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.

[28] S. Li, W. Gong, L. Wang and Q. Gu, "Evolutionary multitasking via reinforcement learning," *IEEE Trans. Emerg. Topics Comput.*, vol. 8, no. 1, pp. 762-775, Feb. 2024.

[29] Y. Jiang, Z. H. Zhan, K. C. Tan and J. Zhang, "Block-level knowledge transfer for evolutionary multitask optimization," *IEEE Trans. Cybern.*, vol. 54, no. 1, pp. 558-571, Jan. 2024.

[30] Z. -F. Xue, Z. -J. Wang, Y. Jiang, Z. -H. Zhan, S. Kwong and J. Zhang, "Multi-level and multi-segment learning multitask optimization via a niching method," *IEEE Trans. Evol. Comput*., doi: 10.1109/TEVC.2024.

[31] J. -Q. Yang *et al.*, "Bi-directional feature fixation-based particle swarm optimization for large-scale feature selection," *IEEE Trans. Big Data*, vol. 9, no. 3, pp. 1004-1017, 1 June 2023

[32] Q. Gong, Y. Xia, J. Zou, Z. Hou and Y. Liu, "Enhancing dynamic constrained multi-objective optimization with multi-centers based prediction," in *IEEE Trans. Evol. Comput*., doi: 10.1109/TEVC.2025.