Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.086077

ARTICLE

A Hybrid Knowledge Transfer for Multitask Optimization

Hai-Xiang Wang[1], **Chu-Xiang Li**[2] **and Zi-Jia Wang**[2,*]

[1]School of Information Engineering, Kaifeng University, Kaifeng, China
[2]School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China

*Corresponding Author: Zi-Jia Wang. Email: zijiawang@gzhu.edu.cn

Received: 23 May 2026; Accepted: 17 August 2026; Published: 28 September 2026

ABSTRACT: Evolutionary multitasking optimization (EMTO) is an emerging research direction in evolutionary computation (EC), with its core objective being the collaborative solution of multiple problems through inter-task knowledge transfer (KT). In classical EMTO algorithms, KT typically relies on the direct exchange or crossover of individuals between populations. However, such transfer strategies often follow singular rules or direct transplantation, which struggle to adequately adapt to the dynamically evolving distributional differences between tasks, and may lead to inefficient transfer or even negative transfer. To tackle this issue, this study presents HKTMTO, a multitask differential evolution algorithm built upon two complementary transfer patterns. The algorithm combines estimation of distribution transfer and adaptive elite transfer to construct a hybrid knowledge transfer framework, overcoming the limitations of traditional single individual-level transfer and improving transfer efficiency without triggering negative transfer. Extensive experiments on the CEC2017 multitask benchmark demonstrate the competitive performance of HKTMTO against mainstream EMTO methods.

KEYWORDS: Evolutionary multitask optimization (EMTO); knowledge transfer; elite transfer; distribution-based transfer; adaptive perturbation

1 Introduction

Evolutionary computation (EC) is a class of computational methods based on natural evolutionary processes, with its core idea originating from the simulation of biological evolution in nature. This category of algorithms encompasses several typical implementations, including genetic algorithms (GA) [1,2], differential evolution (DE) [3,4], particle swarm optimization (PSO) [5,6], ant colony optimization (ACO) [7,8], and emerging philosophy-inspired metaheuristics Yin–Yang-pair Optimization (YYPO) [9]. Leveraging their inherent population diversity and stochastic search operators, EC algorithms are capable of efficient exploration and exploitation within complex solution spaces. This enables them to demonstrate significant advantages in both solving efficiency and solution quality across various domains, such as large-scale optimization [10,11], multi-objective optimization [12,13], expensive optimization [14,15], and multimodal optimization [16,17] problems. Nevertheless, many complex problems in practice often require handling multiple optimization tasks simultaneously, and these tasks usually possess a certain degree of correlation or share some common characteristics such as overlapping search spaces, consistent problem characteristics, and aligned distribution patterns of global optima. Such similarity suggests that if effective knowledge transfer (KT) can be achieved between different tasks, it may accelerate the convergence process of each task. In this background, evolutionary multitask optimization (EMTO) has garnered increasing attention in recent years.

However, while existing EMTO algorithms have explored various designs in knowledge transfer mechanisms, most tend to focus on only a single dimension of knowledge transfer. For instance, methods such as evolutionary multitasking via explicit autoencoding (EMEA) [18] and Block-Level Knowledge Transfer for EMTO (BLKT-DE) [19] primarily rely on a single transfer paradigm—such as encoding-based mapping or block-level transfer. Their operations largely remain at the level of direct individual exchange and fail to conduct knowledge transfer from the deeper, probabilistic distribution structure of the population. On the other hand, algorithms like Evolutionary Multitasking via Reinforcement Learning (RLMFEA) [20] and (EMTO-AI) [21] concentrate their innovation mainly on adaptive adjustments to transfer frequency or the selection of transferring individuals, whereas MFEA with adaptive KT (MFEA-AKT) [22] focuses primarily on the selection and switching of different crossover operators. Although these strategies optimize the timing or manner of transfer, the content being transferred still originates from the direct utilization or simple recombination of existing individuals themselves. They do not actively model and convey the global distribution features inherent in elite solutions. This widespread neglect of the synergistic effect of knowledge transfer at both the macro-distribution and micro-individual levels presents a fundamental challenge to the depth, adaptability, and robustness of knowledge transfer when algorithms face scenarios with significant inter-task distribution differences or complex solution spaces.

To address these aforementioned challenges, this study presents a novel hybrid knowledge transfer for multitask optimization (HKTMTO). This method employs two distinct yet synergistic knowledge transfer mechanisms: First, the algorithm adopts a periodic estimation of distribution transfer strategy, which at fixed intervals probabilistically models the elite individuals selected from the source population to generate new individuals conforming to their statistical distribution characteristics. This facilitates knowledge transfer at the macro-distribution level of the population, effectively overcoming the limitations of direct individual transfer. Second, an elite transfer strategy with adaptive perturbation is introduced. In each generation, the optimal individual from the source task is subjected to a dynamic Gaussian perturbation correlated with the generation count before being migrated to the target task. This ensures the directional transfer of high-quality genes while maintaining the necessary population diversity to avoid premature convergence. These two strategies, operating from the dual dimensions of long-term "population statistical distribution" learning and immediate "elite individual" transfer, jointly construct a knowledge transfer framework that balances exploratory breadth and exploitative precision.

In conclusion, the major contributions of HKTMTO are listed below:

(1) An innovative periodic estimation of distribution transfer strategy is proposed. By intermittently employing estimation of distribution algorithms to model and sample the elite solution space, it achieves a distribution-based knowledge transfer, enhancing the adaptability of the transferred content to the target population's environment.

(2) An adaptive elite transfer perturbation mechanism integrated with the evolutionary stage is designed. This mechanism automatically adjusts the perturbation intensity based on the iteration count, allowing elite individuals to retain their core advantageous traits during transfer while exploring potential improvements through controlled variation, thereby decreasing the likelihood of negative transfer.

(3) Comprehensive experimental evaluations verify that HKTMTO consistently and significantly outperforms several existing EMTO algorithms with respect to convergence speed and solution accuracy on the widely adopted CEC2017 multitask optimization benchmark. This fully proves the effectiveness and robustness of the presented hybrid transfer strategy in enhancing multitask optimization performance.

The remainder of this study is structured as below. Section 2 surveys related work in the field of EMTO. Section 3 presents the details of the proposed hybrid multitask optimization algorithm HKTMTO. Section 4 describes the experimental validation and comparative analysis conducted on the widely used CEC2017 multitask benchmark. Finally, Section 5 summarizes this paper and outlines several promising future research directions.

2 Problem Formulation

2.1 Evolutionary Search Operators

Differential Evolution (DE) is a representative population-based evolutionary optimization algorithm. It has received significant attention in evolutionary computation for its excellent exploration performance and easy implementation. Given its notable advantages in global optimization, this study selects DE as the foundational search operator to construct the multitask optimization framework. DE achieves the iterative evolution of the population through three meticulously designed operators, which work in concert during each generation to advance the search process until termination criteria are met. These core operational stages—mutation, crossover, and selection—are described in detail as below.

(1) Mutation: In this phase, a corresponding mutant vector is generated for every individual according to the distribution of other individuals within the population. For this paper, this study employs the "DE/rand/1" strategy as the mutation operator, which is formulated as follows:

$$v_i = x_{r1} + F \times (x_{r2} - x_{r3}), \tag{1}$$

here $r1$, $r2$, and $r3$ are different random indices within the population, where F is the scaling factor controlling the amplification of differential variations.

(2) Crossover: Following the operation of mutation, the DE algorithm constructs a corresponding trial vector u_i for every target vector x_i using binomial crossover, as described below:

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

here CR_i denotes the crossover rate, and j_{rand} is a randomly selected integer index, which ensures that the trial vector $\boldsymbol{u}_i$ differs from its parent $\boldsymbol{x}_i$ in at least one dimension.

(3) Selection: In the selection phase, an elite retention strategy is executed to filter out more competitive individuals to form the next generation population. Specifically, the current population and the trial population are merged to form a candidate pool of size $2 \times NP$. Subsequently, through precise fitness evaluation and comparison, the top NP individuals, which have the best fitness values, are chosen to constitute the offspring population for the subsequent evolutionary generation.

2.2 EMTO

The core objective of evolutionary multitask optimization algorithms is to significantly enhance the solution efficiency of each individual task through effective collaborative mechanisms. Specifically, considering an optimization problem with K tasks, the mathematical essence of EMTO can be formulated as finding a series of optimal solutions $\{\boldsymbol{x}_1{}^*, \boldsymbol{x}_2{}^*, \ldots, \boldsymbol{x}_k{}^*\}$ as follows:

$$\{x_1{}^*, x_2{}^*, \ldots, x_k{}^*\} = \text{argmin}\{f_1(x_1), f_2(x_2), \ldots, f_k(x_k)\} \tag{3}$$

here each $\boldsymbol{x}_i{}^*$ represents the optimal global solution of the corresponding task T_i.

2.3 Related Work

Evolutionary Multitask Optimization (EMTO), as an emerging computational paradigm, has demonstrated significant potential in recent years for addressing complex multitask problems using evolutionary algorithms. A substantial body of research has accumulated in this field. This section systematically reviews representative EMTO methods.

The first category of methods employs a unified encoding space with an implicit knowledge transfer mechanism. Here, the Multifactorial Evolutionary Algorithm (MFEA) [23] and its variants serve as typical representatives. MFEA innovatively utilizes a single population to address multiple optimization tasks simultaneously in a unified search space, where every individual is associated with a specific task based on a skill factor. This framework facilitates implicit Knowledge Transfer (KT) by performing crossover operations between individuals from distinct tasks, providing an efficient solution for multitask optimization. Building upon this foundation, researchers have proposed various improvements. For instance, Bali et al. [24] developed MFEA-II, which introduced an adaptive transfer parameter learning mechanism. Li et al. [20] proposed RLMFEA, employing reinforcement learning to adaptively adjust the transfer frequency. Zhou et al. [22] proposed MFEA-AKT, which identifies appropriate crossover operators for knowledge transfer across different optimization problems and stages, effectively promoting KT between tasks of varying dimensions. From the perspective of transfer level, these implicit transfer methods predominantly rely on individual-level transfer, where knowledge is exchanged through direct crossover of individuals between tasks. Although variants like MFEA-AKT and RLMFEA introduce adaptive mechanisms to adjust transfer timing or operator selection, the transferred content remains at the individual level, lacking the ability to capture population-level distributional structures.

The second category addresses multitask problems through explicit knowledge interaction among parallel populations. These methods typically maintain multiple independent populations and establish explicit KT channels. For instance, Feng et al. [18] designed an explicit gene transfer framework, Evolutionary Multitasking via Explicit Autoencoding (EMEA), which integrates autoencoder technology to enable individual transfer between the search spaces for different tasks and supports the hybrid use of optimization algorithms like GA and DE. Jiang et al. [19] partitioned population individuals into blocks, achieving block-level knowledge transfer (BLKT-DE). These studies collectively enrich the methodological system of multitask optimization, offering diverse solutions for problems with different characteristics. Similar to the first category, these explicit transfer methods also operate at the individual level, whether through direct individual injection or block-wise exchange. While they provide greater flexibility by maintaining separate populations, they do not model the statistical distribution of elite solutions, leaving the potential for distribution-level knowledge transfer largely unexplored.

However, the aforementioned algorithms predominantly rely on a single transfer strategy rather than a combined approach. To illustrate the differences between existing approaches and the proposed HKTMTO, the core knowledge transfer characteristics of representative state-of-the-art EMTO algorithms are summarized in Table 1. As listed in the table, all comparative methods only execute knowledge transfer at the individual level with fixed or adaptive single transfer strategies, and their transfer operations are triggered in each generation or self-adaptively within individual-level crossover mechanisms. None of these works integrate distribution-level information sharing and elite individual transfer simultaneously, nor do they adopt a periodic transfer schedule matched with multi-scale transfer patterns.

To address this research gap, this paper proposes HKTMTO. This algorithm periodically employs Estimation of Distribution Algorithms (EDA) [25] to construct a probabilistic model from superior individuals in the current population and generates new individuals for transfer to the other population, thereby transferring the overall distribution information of the task. Concurrently, elite solution transfer is performed every generation to rapidly transmit current optimal solution information. Furthermore, inspired by APSO [6], random-dimensional perturbation is adopted on elite solutions to improve the diversity of migrated solutions. By combining these two transfer strategies, the algorithm can more comprehensively and efficiently utilize complementary information between tasks, thereby accelerating convergence and avoiding local optima.

Table 1: Comparison of HKTMTO with representative EMTO algorithms.

Algorithm	Knowledge Transfer Type	Knowledge Transfer Strategy	Transfer Frequency
MFEA	Individual level	Assortative mating rule and vertical knowledge transfer via crossover	Every generation
MFEA-II	Individual level	Crossover incorporated with online estimation of transfer parameters	Every generation
RLMFEA	Individual level	Crossover guided by RL-based transfer policy	Adaptive
MFEA-AKT	Individual level	Adaptive selection of multiple crossover operators for knowledge transfer	Every generation
EMEA	Individual level	Explicit task space mapping constructed via autoencoder	Every generation
EMTO-AI	Individual level	Crossover with self-adaptive knowledge transfer intensity regulation	Adaptive
BLKT-DE	Individual level	Block-level variable exchange across different task populations	Every generation
HKTMTO	Distribution and Individual level	EDA-based distribution transfer and adaptive elite transfer	Periodic and Every generation

3 HKTMTO

In this section, this study presents a detailed summary of the proposed HKTMTO. First, the motivation behind HKTMTO is introduced. Next, the EDA strategy and the AET strategy are elaborated separately. Finally, this study presents the complete HKTMTO algorithm.

3.1 Motivation

In multitask optimization, effective Knowledge Transfer (KT) is crucial for enhancing algorithmic performance. However, most existing transfer methods rely on a single transfer strategy, such as utilizing only elite solution transfer or solely distribution-based transfer. This may result in incomplete knowledge transfer and even induce negative transfer. To more fully leverage inter-task correlations and improve transfer efficiency, this paper proposes a differential evolution algorithm that integrates a dual transfer strategy combining estimation of distribution and elite solution transfer, thereby achieving superior knowledge transfer. Fig. 1 presents the schematic of the hybrid knowledge transfer architecture to illustrate the above transfer design. These two typical single-transfer strategies have complementary strengths and limitations. Distribution-based transfer provides global population guidance and favors high solution accuracy but lacks rapid local optimization capability. In contrast, elite transfer accelerates early convergence via fine-grained individual transfer but easily falls into suboptimal regions without global perspective. Combining the two mechanisms can achieve better search performance.

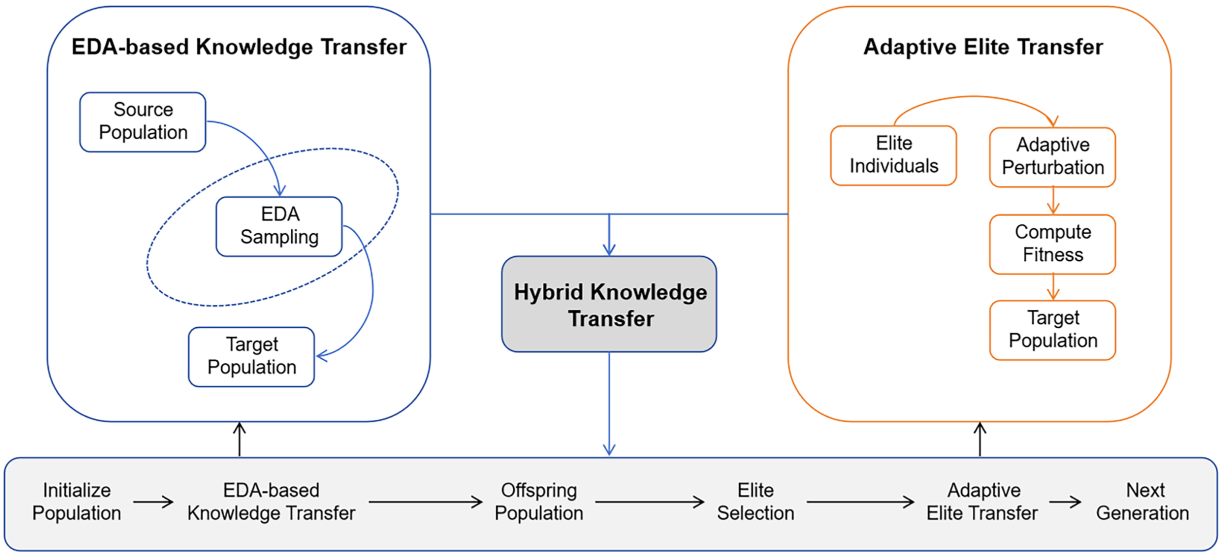


Figure 1: Schematic of the hybrid knowledge transfer architecture based on multilayer optimization.

3.2 Estimation of Distribution Algorithm

In traditional EMTO, knowledge transfer is typically achieved through direct crossover operations between individuals from the source task and the target task. However, this direct transfer mechanism exhibits significant limitations. When individuals from the source task are directly migrated into the evolutionary process of the target population, differences in the search space distribution between tasks often cause the migrated individuals to integrate poorly into the evolutionary trajectory of the target group. This can lead to inefficient knowledge transfer or even induce negative transfer. To tackle this key challenge, an Estimation of Distribution Algorithm (EDA)-based knowledge transfer mechanism is innovatively introduced. The proposed method addresses the shortcomings of traditional direct crossover-based transfer by establishing a dynamic probabilistic model of the source population, thereby facilitating knowledge transfer at the level of population distribution. The main computational steps are formally given in Algorithm 1.

Specifically, this method constructs a probabilistic model that accurately reflects the knowledge structure of the source task by sampling the distribution of the top ρ elite individuals (where ρ is defined as a proportion of the source population, e.g., the top 20%). This model captures both the central tendency represented by the mean vector and the inter-dimensional correlations described by the covariance matrix.

To maintain the stability of cross-task knowledge transfer and prevent the target population from being overwhelmed by foreign solutions, the number of sampled individuals m is kept consistent with the size of the elite set used for modeling, i.e., $m = \lfloor \rho \times NP \rfloor$. This coupled design intuitively controls the transfer scale and reduces the number of defined parameters.

Let the selected population be S, containing n individuals, each with d dimensions. The mean vector μ is then calculated as:

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} s_i \tag{4}$$

where s_i is the ith individual in S. Next, the covariance matrix $\boldsymbol{\Sigma}$ is calculated as:

$$\sum = \frac{1}{n-1}\sum_{i=1}^{n} (s_i - \mu)(s_i - \mu)^T \tag{5}$$

In which $\boldsymbol{\mu}$ denotes the mean vector of S. The term $(\boldsymbol{s}_i - \boldsymbol{\mu})$ is the deviation of $\boldsymbol{s}_i$ from $\boldsymbol{\mu}$, and $n-1$ ensures an unbiased estimate.

Algorithm 1: EDA-based knowledge transfer

Input: Source population P_{src}.
  NP: The population size.
  m: The number of generated individuals.
  ρ: The number of transferred individual ratio.
Output: Migrant population M.
Begin
1: Sort P_{src} in ascending order according to fitness.
2: Select the top $\rho \times NP$ individuals as elite set S.
3: Compute the mean vector $\boldsymbol{\mu}$ of S using Eq. (4).
4: Compute the covariance matrix $\boldsymbol{\Sigma}$ of S using Eq. (5).
5: Regularize $\boldsymbol{\Sigma}$ to ensure positive definiteness using Eq. (7).
6: Initialize empty migrant set M.
7: **for** $i = 1$ to m **do**
8: Sample x_i from multivariate normal distribution N $(\boldsymbol{\mu}, \boldsymbol{\Sigma}_{\mathrm{reg}})$ base on Eq. (6).
9: Add x_i to M.
10: **end for**
End

Subsequently, based on the calculated mean and covariance, m samples are drawn from the multivariate normal distribution N as:

$$x_i \sim N\left(\mu, \sum_{reg}\right), i = 1, 2, \ldots, N \tag{6}$$

with N denoting the number of generated individuals, and x_i denotes the ith individual generated. N follows the multivariate normal distribution, where $\boldsymbol{\mu}$ as its mean vector and $\boldsymbol{\Sigma}_{\mathrm{reg}}$ as its covariance matrix.

What's more, in high-dimensional optimization problems, the covariance matrix may become ill-conditioned, as some eigenvalues are close to zero, leading to over-constrained sampling directions. To address this issue, this study introduces a regularization technique to modify the covariance matrix as follows:

$$\sum_{reg} = \sum + \varepsilon \cdot I \tag{7}$$

where ε is the regularization coefficient and $\mathbf{I}$ denote the identity matrix. By doing so, this operation guarantees the positive definiteness of the covariance matrix and preserves the essential information of its original correlation structure.

Based on this, the individuals generated by EDA faithfully follow the distribution characteristics of the optimal solutions obtained from the source task. Such well-distributed population individuals can better adapt to and match the complex evolutionary landscape of the target task. Consequently, they effectively reduce the mismatch degree of cross-task knowledge transfer, further enhance the environmental adaptability of transferred knowledge, and ultimately improve the overall robustness, generalization ability and optimization performance of the knowledge transfer process.

3.3 Adaptive Elite Transfer

In addition to periodic knowledge transfer via EDA, an elite solution perturbation strategy is integrated into the framework. The core idea of AET is to transfer elite individuals from the source task to the target task after random one-dimensional Gaussian perturbation. This perturbation mechanism helps the algorithm escape local optima.

Specifically, directly copying elite solutions easily triggers premature convergence when elites fall into local basins. Instead, a randomly selected dimension of each elite individual is perturbed by adding Gaussian noise. The perturbation intensity adapts along with evolutionary iterations: larger perturbation magnitudes are adopted in early iterations to explore unknown search regions, while smaller magnitudes are used in later iterations to refine candidate solutions. Restricting perturbation to a single dimension preserves most inherent structural information of elites and introduces controllable population diversity, balancing exploitation of high-quality elite genes and neighborhood exploration.

The following details the design of adaptive perturbation intensity and single-dimensional perturbation.

Linear decay is utilized to adjust perturbation intensity due to its straightforward formulation and reliable performance. Unlike exponential and logarithmic decay schemes that rely on sensitive hyperparameters, this strategy achieves smooth search-range adaptation with no extra tuning parameters.

Moreover, single-dimensional perturbation is essential for high-dimensional optimization. Transferred elites inherit optimized variable correlations and structural features from source tasks. Multi-dimensional perturbation would cause excessive random variations and damage valuable cross-task transfer knowledge. By perturbing only one random dimension, the algorithm implements coordinate-wise local search. This lightweight perturbation adjusts elites on the target fitness landscape while preserving the core inherited structural information from unmodified dimensions.

However, blindly transferring even a perturbed elite individual may still cause negative transfer if the source task's optimum differs significantly from that of the target task. To avoid this risk and ensure that every transfer event actually benefits the target population, a simple yet effective acceptance criterion is adopted in AET. The perturbed elite individual is evaluated on the target task, and it will replace the worst individual in the target population only if its fitness is better than that of the current worst individual. Otherwise, no replacement occurs. This criterion guarantees that the transfer never degrades the average fitness of the target population, as the worst individual is always replaced by a fitter one whenever a transfer happens. In other words, the acceptance rule provides a safety guard against negative transfer while still allowing potentially beneficial elites to enter the population. Through the synergy of random one-dimensional perturbation and fitness-based replacement, AET effectively enhances the algorithm's ability to escape local optima and maintains a positive transfer direction throughout the evolutionary process.

Specifically, the perturbation magnitude for the optimal solution of the source task population in each generation is set according to the following formula:

$$\sigma(g) = \sigma_{\max} - (\sigma_{\max} - \sigma_{\min}) \cdot \frac{g}{G_{\max}} \tag{8}$$

in which σ denotes the current mutation intensity, $\sigma_{\max}$ the maximum mutation intensity, and $\sigma_{\min}$ the minimum mutation intensity; g represents the current evolutionary generation, with $G_{\max}$ denoting the maximum evolutionary generation.

Based on the dynamic perturbation magnitude above, elite solutions are perturbed according to the following formula:

$$e' = e_{best}, e'_j = e_{best,j} + \sigma(g) \cdot \xi \tag{9}$$

where j is a randomly selected dimension from $\{1, \ldots, D\}$, $\xi \sim N(0, 1)$ is a standard normal random variable, and the other dimensions remain unchanged.

The perturbed elite solution is then transferred to the target population for comparison. If its fitness surpasses that of the target population's worst solution, it replaces the original; otherwise, no replacement occurs.

The essential operations of AET are encoded in Algorithm 2. Based on this strategy, a balance between exploration and exploitation is achieved. In the early stage of evolution, a large perturbation strength is adopted to enhance the diversity of transferred individuals, promote the exploration of unknown regions, and avoid premature convergence to local optima. In the later stage, the perturbation strength is gradually reduced, so that the transferred individuals are closer to the original optimal solution of the source task, facilitating the fine-grained search of the target task.

Algorithm 2: Adaptive elite transfer

Input: Elite individual e from source task
g: Current generation.
G: Maximum generation.
perturbation intensity range $[\sigma_{\min}, \sigma_{\max}]$.
target population P_{tar}, its worst individual w_{tar} and fitness $f(w_{tar})$.
Output: The target population P_{tar} after transfer.
Begin
1: Compute adaptive perturbation intensity by Eq. (8).
2: Randomly select a dimension j.
3: Construct perturbed individual es using Eq. (9).
4: Compute fitness = $f(es)$.
5: **if** fitness is better than the fitness $f(w_{tar})$ **then**
6: Replace w_{tar} with es.
7: **end if**
End

3.4 Framework of HKTMTO

The complete procedure in the presented multitask optimization algorithm is outlined in Algorithm 3. Specifically, the complete procedures are given as follows:

Step 1: To begin with, the population is randomly initialized for each task, and the fitness values of all individuals are evaluated, which is described in lines 1–5 of Algorithm 3.

Step 2: In each evolutionary generation, offspring populations are first generated using the Differential Evolution (DE) strategy, as depicted in lines 7–10 of Algorithm 3.

Step 3: The knowledge transfer mechanism is triggered at fixed intervals. For each population, a certain proportion of elite individuals with the best fitness are selected. A probabilistic model is then constructed based on these elite individuals using an Estimation of Distribution Algorithm (EDA), from which an equal proportion of new individuals are sampled. These newly generated individuals are subsequently migrated to the counterpart population, as illustrated in lines 11–15 of Algorithm 3.

Step 4: Merge the parent and offspring populations, and an elite selection strategy is applied, as described in lines 17–18 of Algorithm 3.

Step 5: Random perturbation, based on the Adaptive Elite Transfer (AET), is applied to the optimal solution (elite individual) of each population. The perturbed solution is then compared with the worst individual in the counterpart population: if the fitness of the perturbed solution is better, it replaces the worst individual in that population; otherwise, the counterpart population remains unchanged. This process is shown in lines 19–21 of Algorithm 3.

Steps 2 to 5 are executed iteratively until the maximum number of function evaluations (MaxFEs) is reached.

Algorithm 3: HKTMTO

Input: Two tasks T_1 and T_2.
NP: The population size.
ρ: The selection ratio for EDA modeling.
m: The number of generated EDA migrant individuals.
G: The maximum evolutionary generation.
$\sigma_{\min}$, $\sigma_{\max}$: Adaptive perturbation intensity range
Output: $\boldsymbol{x}^*$: The global best solutions of all tasks.
Begin
1: $FEs = 0$, $g = 0$, $G_{eda} = 5$.
2: **for** t in $\{1, 2\}$ **do**
3: Randomly initialize the population P_t for Task $\boldsymbol{T}_t$.
4: Evaluate all individuals in P_t and update *FEs*.
5: Sort P_t in ascending order according to fitness.
6: **end for**
7: **while** $FEs \leq MaxFEs$:
8: **for** t in $\{1, 2\}$ **do**
9: Generate offspring population $\boldsymbol{C}_t$ for $\boldsymbol{P}_t$ using DE/rand/1.
10: Evaluate $\boldsymbol{C}_t$ and update *FEs*.
11: //EDA-based Knowledge Transfer (Algorithm 1)
12: **if** (g% G_{eda}) == 0 **then**
13: Generate m migrant individuals m_{3-t} by *Algorithm1*(P_{3-t}, m, ρ)
14: Evaluate m_{3-t} and update *FEs*.
15: Add m_{3-t} to the offspring set $\boldsymbol{C}_t$.
16: **end if**

(Continued)

Algorithm 3 (continued)

Merge $\boldsymbol{C}_t$, and $\boldsymbol{P}_t$ to form a candidate pool.
Select the top NP individuals from the candidate pool as the next generation $\boldsymbol{P}_t$ using an elite selection strategy.
//Adaptive Elite Transfer (Algorithm 2)
Select the fittest individual $\boldsymbol{e}_{3-t}$ from Task $\boldsymbol{T}_{3-t}$. Extract worst individual w_{tar} and its fitness $f(w_{tar})$ from $\boldsymbol{P}_t$.
$\boldsymbol{P}_t$ = *Algorithm 2*($\boldsymbol{e}_{3-t}$, g, G, $[\sigma_{\min}, \sigma_{\max}]$, $\boldsymbol{P}_t$, w_{tar}, $f(w_{tar})$), update *FEs*.
end for
$g = g + 1$
end while
End

4 Parameter Settings and Experimental Studies

To comprehensively evaluate the performance of HKTMTO, this study conducted experiments on the widely used EMTO benchmark CEC2017 [26], selecting seven representative EMTO algorithms for comparison: MFEA [23] (2016), EMEA [18] (2019), MTGA [27] (2020), MFEA-AKT [22] (2021), RLMFEA [20] (2024), EMTO-AI [21] (2024) and BLKT-DE [19] (2024). These comparison algorithms span the period from 2016 to 2024 and incorporate various optimization methods, including multi-factor-based approaches, adaptive mechanisms, and reinforcement learning strategies, thereby providing stronger evidence for the effectiveness of HKTMTO.

4.1 Experiment Setup of CEC2017

The CEC2017 [26] evolutionary multi-task optimization benchmark is a widely recognized platform for evaluating the performance of evolutionary multi-task optimization algorithms. The benchmark consists of nine multi-task problem instances, each composed of two basic optimization problems, covering classic functions such as Sphere, Rosenbrock, Ackley, Rastrigin, Griewank, Weierstrass, and Schwefel, with dimensionalities ranging from 25 to 50 D. All these benchmark problems are briefly summarized in Table 2.

Based on the degree of intersection of global optima in the unified search space and the inter-task similarity measured by Spearman's rank correlation, the CEC2017 multitask benchmark problems are organized into nine categories. Table 2 in the original report summarizes these problem types. In terms of the degree of intersection, tasks are grouped into complete-intersection, partial-intersection, and no-intersection cases. Within each group, three levels of inter-task similarity are provided: high, medium, and low similarity.

For complete-intersection tasks, the global optima of the component tasks coincide in the unified search space with respect to all variables. This group includes three types: Complete-Intersection High Similarity (CI + HS), Complete-Intersection Medium Similarity (CI + MS), and Complete-Intersection Low Similarity (CI + LS). CI + HS tasks (e.g., CI + HS-T1 and CI + HS-T2) have highly overlapping search spaces and very similar fitness rankings across the unified domain, which makes them particularly suitable for effective knowledge transfer with relatively low risk of negative transfer. CI + MS tasks (such as CI + MS-T1 and CI + MS-T2) exhibit complete intersection of global optima but only moderate rank correlation. In such cases, the potential benefits of multitasking remain, but transfer mechanisms need to be more selective to avoid propagating misleading genetic material. CI + LS tasks (e.g., CI + LS-T1 and CI + LS-T2) still share a common global optimum in the unified space but have very low rank correlation; here, knowledge transfer becomes much more challenging, as improvements on one task may not systematically translate into improvements on the other.

Table 2: Nine benchmark problems for multitask optimization and their properties.

Problem	Containing Tasks	Intersection Degree	Inter-Task Similarity R_s
CI + HS	Griewank (T_1) / Rastrigin (T_2)	Complete Intersection	1.0000
CI + MS	Ackley (T_1) / Rastrigin (T_2)	Complete Intersection	0.2261
CI + LS	Ackley (T_1) / Schwefel (T_2)	Complete Intersection	0.0002
PI + HS	Rastrigin (T_1) / Sphere (T_2)	Partial Intersection	0.8670
PI + MS	Ackley (T_1) / Rosenbrock (T_2)	Partial Intersection	0.2154
PI + LS	Ackley (T_1) / Weierstrass (T_2)	Partial Intersection	0.0725
NI + HS	Rosenbrock (T_1) / Rastrigin (T_2)	No Intersection	0.9434
NI + MS	Griewank (T_1) / Weierstrass (T_2)	No Intersection	0.3669
NI + LS	Rastrigin (T_1) / Schwefel (T_2)	No Intersection	0.0016

Partial-intersection tasks are characterized by global optima that coincide only on a subset of variables, while differing on the remaining dimensions. This class also consists of three subtypes: Partial-Intersection High Similarity (PI + HS), Partial-Intersection Medium Similarity (PI + MS), and Partial-Intersection Low Similarity (PI + LS). In PI + HS tasks (e.g., PI + HS-T1 and PI + HS-T2), the overlap in decision variables at the optimum, together with high overall rank correlation, allows substantial performance gains from multitasking, provided that the algorithm is capable of exploiting the shared dimensions while filtering out noise from the non-overlapping parts. PI + MS tasks (such as PI + MS-T1 and PI + MS-T2) have partial overlap and moderate similarity in their fitness landscapes; the correlation is not strong enough to support naive knowledge reuse, yet not so weak as to render transfer useless. This setting naturally calls for adaptive transfer strategies that can adjust the intensity and form of knowledge sharing based on ongoing search feedback. PI + LS tasks (e.g., PI + LS-T1 and PI + LS-T2) possess only limited overlap in the global optima and low overall similarity, so even small amounts of careless transfer may induce negative transfer by pushing the search toward regions that are good for one task but detrimental for the other.

In no-intersection tasks, the global optima of the component problems are distinct in the unified search space for all variables. Again, three similarity levels are considered: No-Intersection High Similarity (NI + HS), No-Intersection Medium Similarity (NI + MS), and No-Intersection Low Similarity (NI + LS). NI + HS tasks (e.g., NI + HS-T1 and NI + HS-T2) have disjoint global optima but exhibit highly similar landscape characteristics in terms of rank correlation. In this case, effective knowledge transfer cannot rely on direct overlap in the decision space, but must instead exploit more abstract, generalizable patterns captured in the unified representation, providing a stringent test of an algorithm's ability to extract and reuse such patterns. NI + MS tasks (such as NI + MS-T1 and NI + MS-T2) feature disjoint optima with only moderate similarity; algorithms must carefully balance the extraction of transferable information and the suppression of harmful interference. Finally, NI + LS tasks (e.g., NI + LS-T1 and NI + LS-T2) have completely distinct global optima and very low inter-task similarity. Empirical results indicate that knowledge transfer in such settings is usually not beneficial and can even degrade performance, so these tasks are especially useful for assessing the robustness of multitask algorithms and their capability to detect when transfer should be limited or avoided.

4.2 Parameter Settings

The core parameter configurations for HKTMTO are as follows:

(1) Scaling factor and crossover rate in DE: F = 0.5, CR = 0.6;
(2) The population size NP for each independent task is fixed at 50 individuals;
(3) The knowledge transfer interval G_{eda} for EDA is set to 5 generations;
(4) The EDA sampling proportion ρ is set to 0.2, the regularization coefficient epsilon ε is set to 1e−20, the transfer scale m is governed by the selection ratio ρ via $m = \lfloor \rho \times NP \rfloor$, where the population size NP = 50, leading to $m = 10$;
(5) The $\sigma_{\min}$ and $\sigma_{\max}$ of Adaptive Elite Transfer is set to 0.01 and 0.1, respectively.

To ensure the fairness and reliability of the experimental comparisons, the maximum number of function evaluations is uniformly set to 100,000 for all compared algorithms, and the specific parameters of each compared algorithm are strictly configured according to the values recommended in their original publications. Each experiment is independently repeated 30 times to mitigate the effects of randomness, and the quality of the best solution found in each generation during each run is recorded. For statistical analysis, the Wilcoxon rank-sum test [28] is employed to analyze the significance of performance differences, with the significance level set at $\alpha = 0.05$. The experimental results are annotated with specific symbols: "+" indicates that HKTMTO performs significantly better than the compared algorithm, "≈" indicates comparable performance, and "−" indicates that HKTMTO performs significantly worse. Furthermore, the best statistical result achieved for each test task is highlighted in **bold** font in the tables to facilitate an intuitive comparison of the algorithms' strengths and weaknesses.

4.3 Experimental Results on CEC2017 EMTO Benchmark

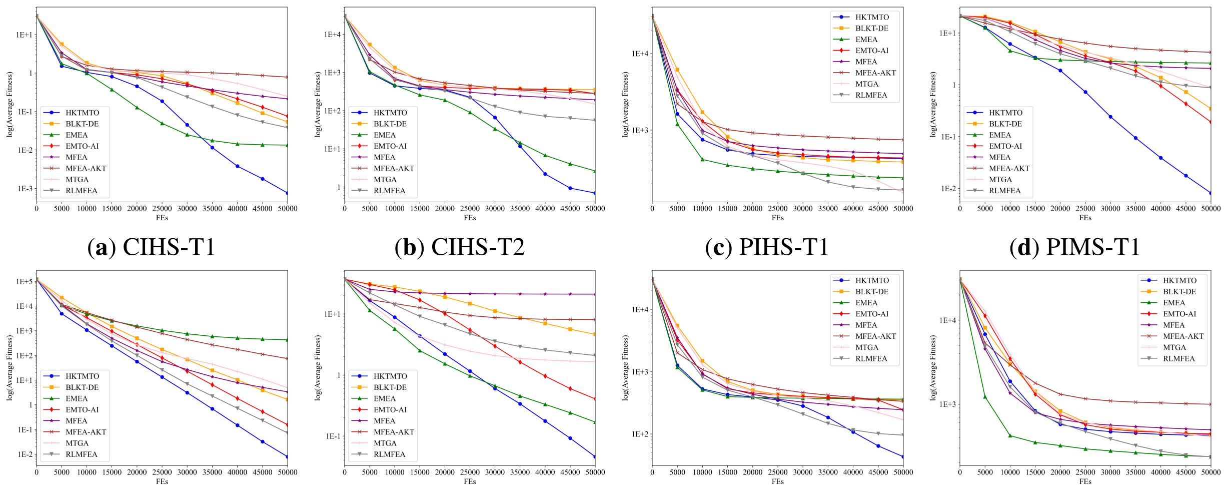
Table 3 presents the experimental results of HKTMTO and other state-of-the-art EMTO algorithms on the CEC2017 benchmark. As shown in Table 3, HKTMTO achieved optimal results in 9 out of 18 tasks, demonstrating its competitive performance. Additionally, HKTMTO outperformed the comparison algorithms MFEA, EMEA, MTGA, MFEA-AKT, EMTO-AI, RLMFEA, and BLKT-DE in 15, 14, 12, 15, 13, 13, and 10 tasks, respectively, while underperforming them in only 3, 2, 4, 3, 2, 5, and 3 tasks. Despite slightly inferior performance in some tasks, HKTMTO significantly outperformed most comparison algorithms overall, validating the effectiveness of its hybrid strategy in solving complex multi-task optimization problems.

Table 3: The CEC17 experimental results of HKTMTO and other EMTO algorithms.

Problem	HKTMTO	MFEA	EMEA	MTGA	MFEA-AKT	EMTO-AI	RLMFEA	BLKT-DE
CIHS-T1	**3.33E−13**	9.73E−2(+)	9.47E−3(+)	2.48E−2(+)	3.06E−1(+)	8.20E−8(+)	6.90E−3(+)	1.43E−4(+)
CIHS-T2	**6.16E−10**	1.25E+2(+)	3.09E−1(+)	5.55E+1(+)	2.33E+2(+)	1.41E−4(+)	1.69E+1(+)	4.14E+1(+)
CIMS-T1	**2.06E−9**	2.94E+0(+)	2.02E+0(+)	1.08E+0(+)	7.10E+0(+)	3.34E−6(+)	8.44E−1(+)	1.57E−3(+)
CIMS-T2	**1.54E−15**	1.27E+2(+)	2.35E+2(+)	5.48E+1(+)	3.48E+2(+)	1.08E−8(+)	3.92E+1(+)	5.76E+1(+)
CILS-T1	2.12E+1	2.02E+1(−)	2.12E+1(≈)	2.12E+1(≈)	2.02E+1(−)	2.12E+1(≈)	**2.00E+1(−)**	2.12E+1(≈)
CILS-T2	1.02E+4	3.69E+3(−)	1.25E+4(+)	5.71E+3(−)	3.93E+3(−)	1.14E+4(+)	**3.33E+3(−)**	1.25E+4(+)
PIHS-T1	3.95E+2	4.41E+2(+)	2.00E+2(−)	**5.17E+1(−)**	6.38E+2(+)	4.04E+2(+)	1.63E+2(−)	3.56E+2(−)
PIHS-T2	**2.73E−9**	5.60E−1(+)	4.00E+2(+)	1.76E−3(+)	4.53E+0(+)	6.21E−7(+)	3.92E−3(+)	7.94E−4(+)
PIMS-T1	**5.32E−6**	1.96E+0(+)	2.46E+0(+)	4.56E−1(+)	3.28E+0(+)	2.56E−4(+)	6.29E−1(+)	1.58E−3(+)
PIMS-T2	8.42E+1	2.05E+2(+)	8.04E+1(+)	1.62E+2(≈)	4.07E+2(+)	7.64E+1(−)	9.70E+1(+)	**4.66E+1(−)**
PILS-T1	**3.44E−5**	1.90E+1(+)	2.32E−2(≈)	1.82E+0(+)	7.40E+0(+)	1.41E+1(+)	1.71E+0(+)	1.00E−1(+)
PILS-T2	**1.78E−5**	1.99E+1(+)	2.04E−2(+)	2.10E+0(+)	8.03E+0(+)	2.05E−1(+)	1.86E+0(+)	2.05E+0(+)
NIHS-T1	4.78E+1	3.17E+2(+)	2.94E+2(+)	2.39E+2(+)	7.19E+2(+)	**4.63E+1(−)**	1.46E+2(+)	4.80E+1(≈)
NIHS-T2	3.44E+1	1.73E+2(+)	3.51E+2(+)	5.20E+1(+)	2.73E+2(+)	**2.00E−4(≈)**	9.22E+1(+)	7.91E+1(≈)
NIMS-T1	**6.99E−6**	1.31E−1(+)	1.72E+0(+)	7.12E−3(+)	3.47E−1(+)	8.50E−4(+)	1.68E−2(+)	2.11E−5(+)
NIMS-T2	5.86E−1	2.74E+1(+)	2.08E+0(+)	7.08E+0(+)	2.55E+1(+)	5.63E−1(≈)	1.37E+1(+)	**2.92E−1(≈)**
NILS-T1	3.97E+2	4.48E+2(+)	**1.92E+2(−)**	2.06E+2(−)	8.98E+2(+)	4.08E+2(+)	2.29E+2(−)	3.99E+2(≈)
NILS-T2	1.05E+4	3.71E+3(−)	1.22E+4(+)	4.46E+3(−)	4.18E+3(−)	1.17E+4(+)	**3.16E+3(−)**	7.80E+3(−)
Number of +/≈/−		15/0/3	14/2/2	12/2/4	15/0/3	13/3/2	13/0/5	10/5/3

To provide a more detailed analysis of the convergence behaviors of HKTMTO and other EMTO algorithms on the benchmark tasks, convergence curves of these algorithms are plotted. As illustrated in Fig. 2 (a) CIHS-T1 and (b) CIHS-T2, HKTMTO shows a sharp downward trend in the early evolutionary stages, indicating rapid convergence, and then gradually stabilizes in the later stages, which helps it quickly approach a high-quality solution. For the task PIHS-T1 in Fig. 2c, after maintaining a steady convergence rate in the mid phase, HKTMTO further accelerates its convergence and quickly narrows the gap with other algorithms. From Fig. 2 (d) PIMS-T1, (e) PIHS-T2, and (f) PILS-T2, it can be seen that HKTMTO maintains a leading convergence speed throughout the evolutionary process: it quickly pulls ahead in the early stages and maintains a stable downward trend, eventually achieving the optimal log average fitness among all algorithms. In Fig. 2g NIHS-T2, HKTMTO demonstrates a fast convergence rate in the initial phase and then stabilizes at a low fitness level. Finally, as shown in Fig. 2h NILS-T1, although HKTMTO has a relatively moderate convergence rate in the early evolutionary stage, it significantly accelerates the optimization process in the later stage, and ultimately outperforms other competing algorithms in the final fitness value.

However, experimental results also reveal two clear limitations of the proposed HKTMTO. Firstly, HKTMTO's performance heavily relies on the similarity between tasks. The algorithm delivers fast convergence and high-quality solutions for medium and high-similarity task pairs with overlapping search spaces, while showing noticeable performance drops on low-similarity benchmarks such as NILS tasks with unrelated fitness landscapes. Secondly, HKTMTO employs fixed dual transfer mechanisms lacking real-time adaptive tuning. Periodic EDA distribution transfer and per-generation elite transfer fail to dynamically adapt to changing task correlations. For task pairs with little shared structural features, rigid cross-task knowledge transfer introduces redundant evolutionary information, wastes additional function evaluations, and leads to mild performance decline of a single task in the later search stage. This limitation restricts the general applicability of HKTMTO to various multitask optimization scenarios.



(**a**) CIHS-T1 (**b**) CIHS-T2 (**c**) PIHS-T1 (**d**) PIMS-T1

(**e**) PIHS-T2 (**f**) PILS-T2 (**g**) NIHS-T2 (**h**) NILS-T1

Figure 2: Convergence curves of the average fitness on several tasks from the CEC2017 benchmark.

Overall, HKTMTO achieves rapid convergence during the early evolutionary stages on the majority of CEC2017 tasks and outperforms other comparative algorithms in terms of solution quality, fully demonstrating its superiority and effectiveness.

4.4 Ablation Experiments

To quantitatively evaluate the individual contributions of the two key knowledge transfer components, namely the EDA-based transfer and the Adaptive Elite Transfer (AET), a comprehensive ablation study is conducted with three variants: HKTMTO-w/o-AET, HKTMTO-w/o-EDA, and HKTMTO-w/o-AET-EDA. HKTMTO-w/o-AET removes the AET module, HKTMTO-w/o-EDA discards the EDA-based transfer module, and HKTMTO-w/o-AET-EDA excludes both components simultaneously. The experimental results are summarized in Table 4.

Table 4: Comparison among HKTMTO variants with or without AET and EDA.

Problem	**HKTMTO**	**HKTMTO-w/o-AET**	**HKTMTO-w/o-EDA**	**HKTMTO-w/o-AET-EDA**
CIHS-T1	**3.33E−13**	3.25E−12(≈)	2.48E−4(+)	9.80E−6(+)
CIHS-T2	**6.16E−10**	4.76E−9(≈)	2.99E−1(+)	3.97E+2(+)
CIMS-T1	**2.06E−9**	2.11E−9(≈)	1.13E−5(+)	4.48E+0(+)
CIMS-T2	**1.54E−15**	2.19E−15(≈)	8.68E−8(+)	3.94E+2(+)
CILS-T1	**2.12E+1**	**2.12E+1(≈)**	**2.12E+1(≈)**	**2.12E+1(≈)**
CILS-T2	1.02E+4	1.01E+4(≈)	**9.97E+3(≈)**	1.02E+4(≈)
PIHS-T1	3.95E+2	**3.94E+2(≈)**	3.99E+2(≈)	3.95E+2(≈)
PIHS-T2	2.73E−9	2.35E−9(≈)	**2.10E−9(≈)**	3.15E−9(≈)
PIMS-T1	5.32E−6	**4.36E−6(≈)**	1.16E−5(+)	3.78E+0(+)
PIMS-T2	8.42E+1	8.53E+1(≈)	**6.00E+1(−)**	1.95E+2(−)

(Continued)

Table 4 (continued)

Problem	HKTMTO	HKTMTO-w/o-AET	HKTMTO-w/o-EDA	HKTMTO-w/o-AET-EDA
PILS-T1	**3.44E−5**	4.95E−5(+)	2.01E−2(+)	5.20E+0(+)
PILS-T2	1.78E−5	**1.76E−6(≈)**	5.93E−3(+)	4.71E−1(+)
NIHS-T1	**4.78E+1**	5.14E+1(≈)	5.61E+1(≈)	6.47E+1(≈)
NIHS-T2	3.44E+1	**1.23E+1(−)**	2.90E+1(≈)	3.97E+2(+)
NIMS-T1	6.99E−6	6.62E−6(≈)	**5.42E−6(−)**	8.72E−6(≈)
NIMS-T2	5.86E−1	**2.28E−1(≈)**	3.03E−1(−)	2.18E+0(+)
NILS-T1	3.97E+2	4.02E+2(≈)	**3.91E+2(≈)**	3.97E+2(≈)
NILS-T2	1.05E+4	1.00E+4(−)	**9.71E+3(−)**	1.02E+4(≈)
Number of +/≈/−		1/15/2	7/7/4	9/8/1

Several profound insights can be derived from the comparisons:

First, the comparison between HKTMTO and HKTMTO-w/o-AET yields a large number of tie cases (1/15/2). This result does not imply that the AET component is redundant. Instead, it verifies the adaptive protection behavior of AET during knowledge transfer. The AET module filters transferred elites strictly according to fitness quality. For low-similarity task pairs such as CILS, imported elite solutions exhibit poor fitness in the target search space. In this case, the fitness-based selection naturally suppresses the elite transfer behavior. This adaptive suppression avoids negative transfer, leading to comparable performance between HKTMTO and HKTMTO-w/o-AET on dissimilar tasks.

Second, comparative results between HKTMTO and HKTMTO-w/o-EDA demonstrate the practical characteristics of distribution-level transfer. Removing the EDA component greatly weakens the optimization performance on highly correlated tasks such as CIHS, which indicates that EDA transfer effectively assists cross-task knowledge sharing when tasks share similar search characteristics.

Finally, the full HKTMTO algorithm outperforms the baseline variant without both EDA and AET in 9 of 18 test cases, resulting in a win-tie-loss balance of 9/8/1. This result indicates that the cooperative use of macroscopic distribution transfer and microscopic elite perturbation can effectively improve overall multitask optimization performance.

Furthermore, generation-level statistical analysis is performed to intuitively observe the iterative performance differences. As shown in Fig. 3, green blocks represent generations where HKTMTO is statistically superior to HKTMTO-w/o-AET, red blocks denote generations where the variant without AET performs better, and white blocks indicate no significant difference. All comparisons are conducted via the Wilcoxon rank-sum test at the significance level of $\alpha < 0.05$, where CI, PI, and NI correspond to completely intersecting, partially intersecting, and non-intersecting task pairs, respectively. The results show that the AET strategy steadily improves solution precision throughout the evolutionary process, validating its effectiveness.

Figure 3: Statistical significance heatmap of performance comparison between HKTMTO and HKTMTO w/o-AET on CEC2017 benchmark.

4.5 Parameter Sensitivity Analysis

This section further analyzes the influence of three key parameters in HKTMTO: the EDA transfer interval G_{eda}, elite selection ratio ρ, and regularization coefficient epsilon ε. To evaluate their effects, the performance of HKTMTO under different parameter configurations is compared on the CEC2017 benchmark. For each configuration, the algorithm is run 30 times independently, and the average fitness values of all tasks are recorded. For the transfer interval G_{eda}, summarized sensitivity results are presented in Table 5, while detailed task-level comparison results are placed in the Supplementary Materials as shown in Table S1. For the selection ratio ρ, summarized outcomes are shown in Table 6, and full detailed results are available in the Supplementary Materials as shown in Table S2. For the regularization coefficient epsilon ε, summarized outcomes are listed in Table 7, and full detailed results are available in the Supplementary Materials as shown in Table S3. "+", "≈", and "−" denote that the baseline is statistically better than, similar to, or worse than the compared setting, respectively.

Table 5: Sensitivity analysis of the migration interval G_{eda}, (Fixed $\rho = 0.2$).

G_{eda}	Number of +/≈/− (Baseline: $G_{\text{eda}} = 5$)
5	–
10	6/12/0
15	6/11/1
20	7/9/2

Table 6: Sensitivity analysis of the selection ratio ρ, (Fixed $G_{\mathrm{eda}} = 5$).

ρ	m	Number of +/≈/− (Baseline: $\rho = 0.2$)
0.05	2	6/10/2
0.1	5	4/12/2
0.15	7	3/15/0
0.2	10	–

Table 7: Sensitivity analysis of the regularization coefficient epsilon ratio ε.

ε	Number of +/≈/− (Baseline: ε = 1e−20)
1e−5	5/13/0
1e−10	6/12/0
1e−15	4/14/0
1e−20	–

To investigate the influence of two key parameters: the EDA transfer interval G_{eda} and the elite selection ratio ρ, the study conducted additional experiments on the CEC2017 benchmark. For G_{eda}, values of 5, 10, 15, and 20 were tested while fixing $\rho = 0.2$. For ρ, values of 0.05, 0.1, 0.15, and 0.2 were tested while fixing $G_{\mathrm{eda}} = 5$. For ε, candidate values of 1e−5,1e−10,1e−15, and 1e−20 defined as the baseline. Each configuration was independently run 30 times, and the Wilcoxon rank-sum test ($\alpha = 0.05$) was employed for performance comparison. Specifically, the reference configuration was $G_{\mathrm{eda}} = 5$, $\rho = 0.2$.

The experimental results indicate that the performance of HKTMTO is sensitive to the choice of G_{eda}. Compared to the baseline ($G_{\mathrm{eda}} = 5$), increasing the interval generally leads to performance degradation on several tasks. Specifically, when $G_{\mathrm{eda}} = 10$, the baseline significantly outperforms this setting on 6 out of 18 tasks, while no task shows improvement. As G_{eda} further increases to 15 and 20, performance continues to decline, with the baseline achieving superior results on 6 and 7 tasks. These results demonstrate that maintaining a smaller interval such as $G_{\mathrm{eda}} = 5$ is crucial for preserving the algorithm's search efficiency, whereas larger intervals reduce performance.

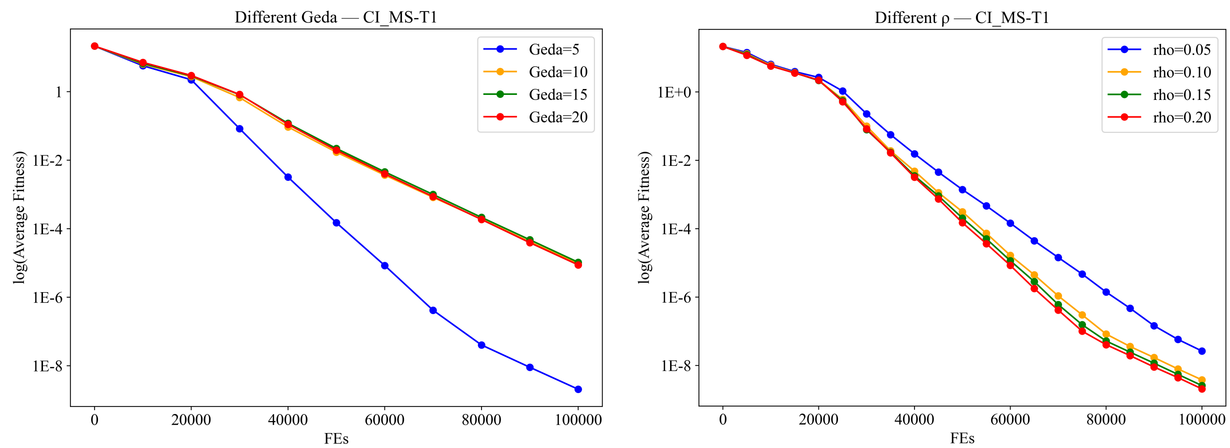
When varying ρ from 0.05 to 0.2, most tasks (10–15 out of 18) show no significant change in performance. However, excessively small ρ (e.g., 0.05) increases the number of tasks with degraded performance. This indicates that the hybrid transfer mechanism does not rely on precise tuning of the elite ratio within a reasonable range. Consistent optimization performance across a wide parameter range demonstrates that the proposed hybrid transfer framework is robust.

For ε, all tested alternative values only bring minor performance fluctuations. As shown in Table 7, configurations ε = 1e−5, 1e−10 and 1e−15 generate 13, 12 and 14 tie results, respectively, and no task achieves statistically superior results compared with the baseline ε = 1e−20. This indicates the regularization coefficient epsilon has limited influence on overall search quality.

4.6 Convergence Behavior under Different Parameter Settings

Although the final fitness values reported in the previous section show no significant differences across the tested parameter ranges, the study further examines the convergence curves to see whether the transient behavior is affected. Fig. 4 plots the average convergence curves of HKTMTO on CIMS-T1 with different G_{eda}

values (5, 10, 15, 20) and different ρ values (0.05, 0.1, 0.15, 0.2). The curves are nearly overlapping, indicating that both the early convergence speed and the final stagnation level are insensitive to these parameters. This is a desirable property, as it means practitioners can safely use the default values (G_{eda} = 5, ρ = 0.2) without risking performance loss due to suboptimal tuning. The robustness of convergence behavior also suggests that the hybrid transfer framework is inherently stable: the periodic EDA injection and the per-generation AET jointly maintain a consistent search dynamic regardless of minor variations in the transfer frequency or model precision.



(**a**) different G_{eda} (**b**) different ρ

Figure 4: Convergence curves under different parameter settings from the CEC2017 benchmark.

4.7 Time and Space Complexity Analysis

To further explore the internal mechanism and performance of the proposed algorithm, the study conducts an asymptotic time complexity analysis of HKTMTO. Analyzing the asymptotic time complexity reveals how an algorithm scales with problem dimension and population size, which is critical for practical applications involving high-dimensional optimization.

Let D denote the problem dimension, NP the population size per task, K the number of tasks (here K = 2), and G_{max} the maximum number of generations. In each generation, HKTMTO performs DE operations (mutation, crossover, selection) for each task, costing $O(K{\cdot}NP{\cdot}D)$. The adaptive elite transfer (AET) operates in $O(K{\cdot}D)$ per generation due to the one-dimensional perturbation. The EDA-based distribution transfer is triggered every G_{eda} = 5 generations. Its dominant cost is the Cholesky decomposition of the covariance matrix, which is $O(D^3)$, and the construction of the covariance matrix from the elite set costs $O(m{\cdot}D^2)$, where $m = \rho{\cdot}NP$ is the number of selected elites. Since $m \ll NP$ and G_{eda} is constant, the amortized per-generation cost of EDA is $O\ (D^3/G_{eda} + \rho{\cdot}NP{\cdot}D^2/G_{eda})$.

Thus, the overall per-generation time complexity of HKTMTO is:

$$o\left(\mathrm{K}\cdot\mathrm{NP}\cdot\mathrm{D}+\mathrm{K}\cdot\mathrm{D}+\frac{\mathrm{D}^3}{\mathrm{G_{eda}}}+\frac{\mathrm{m}\cdot\mathrm{D}^2}{\mathrm{G_{eda}}}\right) \tag{10}$$

In addition to the time complexity, the space complexity of HKTMTO is economical. The algorithm requires storing K populations of size NP with dimension D, consuming $O(K{\cdot}NP{\cdot}D)$ memory space. The EDA mechanism additionally stores a $D \times D$ covariance matrix and a mean vector, requiring $O(D^2)$ space.

Therefore, the overall space complexity is O($K \cdot NP \cdot D + D^2$). For typical high-dimensional optimization problems, this translates to tens of kilobytes of RAM, demonstrating that HKTMTO is lightweight.

Regarding the execution time, empirical wall-clock time comparisons are often biased by hardware configurations, operating systems, and varying levels of code optimization. Therefore, following standard practices in evolutionary computation, the algorithmic computational cost is strictly aligned by the maximum number of Function Evaluations (FEs). Since the asymptotic time and space complexities of HKTMTO are comparable to conventional algorithms, and all methods are restricted to the exact same FEs, HKTMTO achieves improvements without incurring prohibitive computational burdens.

5 Conclusion

The key of EMTO lies in KT. However, most KT strategies rely on a single pattern or timescale for knowledge exchange, which may inadequately balance convergence speed with the risk of negative transfer. Even though some strategies incorporate multiple operators, they often lack a coordinated mechanism to synergize different types of knowledge effectively. To address this issue, this paper proposes a novel HKTMTO, which introduces a hybrid KT framework. This framework coordinates periodic EDA-based distribution transfer for robust global exploration with continuous elite transfer for rapid convergence, and further incorporates adaptive elite perturbation to maintain diversity. Thereby, it significantly enhances the synergy and overall efficiency of cross-task optimization. The experimental results indicate that HKTMTO significantly outperforms other state-of-the-art EMTO algorithms on one widely-used multitask benchmark, CEC2017, demonstrating the effectiveness of HKTMTO.

Since the current HKTMTO framework has demonstrated the effectiveness of coordinated knowledge transfer, future work will explore deeper integration of the periodic distribution estimation transfer and the adaptive perturbation mechanism within the multitask optimization paradigm to further enhance the performance and stability of the algorithm. Additionally, this article plans to extend the core mechanisms of HKTMTO to other challenging optimization domains, such as large-scale and dynamic optimization [10,11], to broaden its application potential. Moreover, although the CEC2017 benchmark provides a comprehensive testbed covering diverse task similarities and landscape types, this article recognizes that further evaluation on additional benchmarks and real-world engineering optimization cases would further establish the generalizability of HKTMTO. This is left as an important direction for future work.

Acknowledgement: Not applicable.

Funding Statement: This work was supported in part by the National Natural Science Foundations of China (NSFC) under Grant 62106055, in part by the Guangdong Natural Science Foundation under Grants 2025A1515010256, and 2026A1515011523, and 2026A1515060004, in part by the Guangzhou Science and Technology Planning Project under Grants 2023A04J0388 and 2023A03J0662.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Hai-Xiang Wang and Chu-Xiang Li; methodology, Hai-Xiang Wang and Chu-Xiang Li; software, Chu-Xiang Li and Zi-Jia Wang; validation, Hai-Xiang Wang and Zi-Jia Wang; formal analysis, Hai-Xiang Wang and Chu-Xiang Li; investigation, Hai-Xiang Wang; resources, Hai-Xiang Wang; data curation, Hai-Xiang Wang and Chu-Xiang Li; writing—original draft preparation, Hai-Xiang Wang; writing—review and editing, Chu-Xiang Li and Zi-Jia Wang; visualization, Hai-Xiang Wang and Chu-Xiang Li; supervision, Zi-Jia Wang; project administration, Zi-Jia Wang; funding acquisition, Hai-Xiang Wang and Zi-Jia Wang. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article or its Supplementary Materials.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Supplementary Materials: The supplementary material is available online at https://www.techscience.com/doi/10.32604/cmes.2026.086077/s1.

Abbreviations

The following abbreviations are used in this manuscript:

EMTO	Evolutionary multitasking optimization
EC	Evolutionary computation
KT	Knowledge transfer
HKTMTO	A Hybrid Knowledge Transfer for Multitask Optimization
DE	Differential evolution
PSO	Particle swarm optimization
ACO	Ant colony optimization
VRP	Vehicle routing problems
EMEA	Evolutionary multitasking via explicit autoencoding
BLKT-DE	Block-level knowledge transfer for evolutionary multitask optimization
RLMFEA	Evolutionary multitasking via reinforcement learning
EMTO-AI	Evolutionary multi-task optimization with adaptive intensity of knowledge transfer
MFEA-AKT	Toward adaptive knowledge transfer in multifactorial evolutionary computation
MFEA	Multifactorial evolution: toward evolutionary multitasking
MFEA-II	Multifactorial Evolutionary Algorithm with Online Transfer Parameter Estimation: MFEA-II
MTGA	Multitasking genetic algorithm (MTGA) for fuzzy system optimization
APSO	Adaptive Particle Swarm Optimization

References

1. El-Shorbagy MA. A review of genetic algorithms: principles, procedures, and applications in optimization. Comput Model Eng Sci. 2026;147(1):6. doi:10.32604/cmes.2026.079859.
2. Pholdee N, SujinBureerat, Nuantong W. Kriging surrogate-based genetic algorithm optimization for blade design of a horizontal AxisWind turbine. Comput Model Eng Sci. 2021;126(1):261–73. doi:10.32604/cmes.2021.012349.
3. Wu C, Kong W. Interior point method-assisted differential evolution for expensive optimization of secondary source deployment. Comput Model Eng Sci. 2026;146(2):14. doi:10.32604/cmes.2026.077613.
4. Zhu Y, Zhao T. Deep structure optimization for incremental hierarchical fuzzy systems using improved differential evolution algorithm. Comput Model Eng Sci. 2024;138(2):1139–58. doi:10.32604/cmes.2023.030178.
5. Binbusayyis A, Sha M. Stability prediction in smart grid using PSO optimized XGBoost algorithm with dynamic inertia weight updation. Comput Model Eng Sci. 2025;142(1):909–31. doi:10.32604/cmes.2024.058202.
6. Zhan ZH, Zhang J, Li Y, Chung HS. Adaptive particle swarm optimization. IEEE Trans Syst Man Cybern Part B Cybern. 2009;39(6):1362–81. doi:10.1109/TSMCB.2009.2015956.
7. Wang C, Zhu R, Jiang Y, Liu W, Jeon SW, Sun L, et al. A scheme library-based ant colony optimization with 2-opt local search for dynamic traveling salesman problem. Comput Model Eng Sci. 2023;135(2):1209–28. doi:10.32604/cmes.2022.022807.
8. Shao C, Heng C, Miao Z, Chen Y, Li H, Tang Z. An adaptive ant colony optimization-based obstacle-avoidance routing algorithm for Network-on-Chip. Microprocess Microsyst. 2026;121(5):105256. doi:10.1016/j.micpro.2026.105256.

9. Wang WC, Xu L, Chau KW, Zhao Y, Xu DM. An orthogonal opposition-based-learning Yin–Yang-pair optimization algorithm for engineering optimization. Eng Comput. 2022;38(2):1149–83. doi:10.1007/s00366-020-01248-9.
10. Si L, Wang Z, Zhang X, Tian Y. Information bottleneck theory-guided dimension reduction for large-scale multi-objective optimization. IEEE Trans Evol Comput. 2026;(99):1. doi:10.1109/TEVC.2026.3669468.
11. Li C, Yen GG, He Z. Diffusion learning-guided evolution for large-scale dynamic multi-objective optimization. IEEE Trans Evol Comput. 2026. doi:10.1109/TEVC.2026.3659883.
12. Zhang L, Li W, Zheng Y, Yao X, Wang R, Zhang T. High-dimensional multimodal multi-objective optimization via subspace projection and angular diversity management. Expert Syst Appl. 2026;316(22):131814. doi:10.1016/j.eswa.2026.131814.
13. Yin L, Zhao B, Gao F. Transformer-accelerated multi-objective high-dimensional multi-fractional-order optimization algorithm for proportional-integral controller parameters tuning of doubly-fed induction generators. Swarm Evol Comput. 2026;101:102325. doi:10.1016/j.swevo.2026.102325.
14. Zhen H, Li X, Gong W, Hu X. Composite indicator-guided infilling sampling for expensive multi-objective optimization. Swarm Evol Comput. 2026;101(2):102312. doi:10.1016/j.swevo.2026.102312.
15. Chen S. Conditional diffusion with gradient guidance for high-dimensional expensive multi-objective optimization. Swarm Evol Comput. 2026;102(2):102340. doi:10.1016/j.swevo.2026.102340.
16. Fan J, Qu L. Multimodal multi-objective particle swarm optimization algorithm based on adaptive selection for multiple unmanned surface vehicle path planning. Robot Auton Syst. 2026;200(4):105393. doi:10.1016/j.robot.2026.105393.
17. Ghosh AK, Bhattacharjee S, Garai G. Advanced self-adaptive distributed differential evolution with hierarchical learning algorithm for multimodal optimization in communication system. Swarm Evol Comput. 2026;106(15):102420. doi:10.1016/j.swevo.2026.102420.
18. Feng L, Zhou L, Zhong J, Gupta A, Ong YS, Tan KC, et al. Evolutionary multitasking via explicit autoencoding. IEEE Trans Cybern. 2019;49(9):3457–70. doi:10.1109/TCYB.2018.2845361.
19. Jiang Y, Zhan ZH, Tan KC, Zhang J. Block-level knowledge transfer for evolutionary multitask optimization. IEEE Trans Cybern. 2024;54(1):558–71. doi:10.1109/TCYB.2023.3273625.
20. Li S, Gong W, Wang L, Gu Q. Evolutionary multitasking via reinforcement learning. IEEE Trans Emerg Top Comput Intell. 2024;8(1):762–75. doi:10.1109/TETCI.2023.3281876.
21. Zhou X, Mei N, Zhong M, Wang M. Evolutionary multi-task optimization with adaptive intensity of knowledge transfer. IEEE Trans Emerg Top Comput Intell. 2024;(99):1–13. doi:10.1109/TETCI.2024.3418810.
22. Zhou L, Feng L, Tan KC, Zhong J, Zhu Z, Liu K, et al. Toward adaptive knowledge transfer in multifactorial evolutionary computation. IEEE Trans Cybern. 2021;51(5):2563–76. doi:10.1109/TCYB.2020.2974100.
23. Gupta A, Ong YS, Feng L. Multifactorial evolution: toward evolutionary multitasking. IEEE Trans Evol Comput. 2016;20(3):343–57. doi:10.1109/TEVC.2015.2458037.
24. Bali KK, Ong YS, Gupta A, Tan PS. Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II. IEEE Trans Evol Comput. 2020;24(1):69–83. doi:10.1109/TEVC.2019.2906927.
25. Larrañaga P, Lozano JA. Estimation of distribution algorithms: a new tool for evolutionary computation. Boston, MA, USA: Kluwer Academic Publishers; 2002.
26. Yuan Y, Ong YS, Feng L, Qin A, Gupta A, Da B, et al. Evolutionary multitasking for multiobjective continuous optimization: benchmark problems, performance metrics and baseline results. arXiv:1706.02766. 2017. doi:10.48550/arxiv.1706.02766.
27. Wu D, Tan X. Multitasking genetic algorithm (MTGA) for fuzzy system optimization. IEEE Trans Fuzzy Syst. 2020;28(6):1050–61. doi:10.1109/TFUZZ.2020.2968863.
28. Li JY, Zhan ZH, Wang C, Jin H, Zhang J. Boosting data-driven evolutionary algorithm with localized data generation. IEEE Trans Evol Comput. 2020;24(5):923–37. doi:10.1109/TEVC.2020.2979740.