

Article

A Coordinated Feedback-Based Hierarchical Triangulation Topology Aggregation Optimizer for Bilevel Optimization Problems

Xiaomei Yu ¹, Yu Li ^{2,*} , Jingsen Liu ³ , Huan Zhou ⁴ and Qian Guo ⁴

¹ School of Business Administration, South China University of Technology, Guangzhou 510641, China; yuxm@henu.edu.cn

² Institute of Management Science and Engineering, School of Business, Henan University, Kaifeng 475004, China

³ Institute of Intelligent Network Systems, Software School, Henan University, Kaifeng 475004, China; ljs@henu.edu.cn

⁴ School of Business, Henan University, Kaifeng 475004, China; zhouhuan@henu.edu.cn (H.Z.); amandaguo_bs@henu.edu.cn (Q.G.)

* Correspondence: leey@henu.edu.cn

Abstract

Bilevel optimization problems (BOPs) are widely encountered in hierarchical decision-making, wherein the embedding of the lower-level problem within the upper-level problem poses significant challenges to the solution process. To solve complex BOPs, this paper proposes a coordinated feedback-based hierarchical triangulation topology aggregation optimizer (CFHTTAO). CFHTTAO is developed by constructing a hierarchical solution framework and designing improvement mechanisms to enhance its performance. Specifically, a knowledge-sharing mechanism is established to promote information exchange among triangular topological units, a coordinated feedback mechanism is designed to adaptively select a suitable lower-level search method and feed the obtained lower-level solution back to the upper level, and an elite learning evolution mechanism is used to reduce the risk of premature convergence. CFHTTAO is evaluated on the SMD test suite under different dimensional settings, where the test problems involve various complex characteristics of BOPs. In addition, the Friedman test results show that CFHTTAO achieves the smallest mean ranks, with upper-level mean ranks of 1.08, 1.50, and 1.50 and lower-level mean ranks of 1.00, 1.17, and 1.25 on the SMD(2 + 3), SMD(5 + 5), and SMD(10 + 10) problems, respectively. Finally, CFHTTAO is applied to the inverse optimal control problem as the practical case study. Overall, experimental results demonstrate that CFHTTAO achieves better solution accuracy, stronger stability, and faster convergence when solving complex BOPs, and exhibits strong applicability to the practical bilevel optimization task.



Academic Editor: Andrea Scozzari

Received: 10 June 2026

Revised: 17 July 2026

Accepted: 18 July 2026

Published: 20 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: bilevel optimization; triangulation topology aggregation optimizer; hierarchical framework; coordinated feedback

MSC: 90C26; 90C59

1. Introduction

Many practical problems involve hierarchical decision-making structures, in which decisions at one level influence decisions at the other level [1,2]. These problems are formulated as bilevel optimization problems (BOPs) and arise in various applications [3], such

as supply chain management [4], renewable energy communities [5], and inverse optimal control [6]. Solving linear BOPs has been proven to be NP-hard, and the solution difficulty is further intensified when BOPs involve complex characteristics such as nonlinearity, nonconvexity, and multimodality, which often limit the applicability of traditional methods [7]. Therefore, it is of great importance to explore more effective algorithms for BOPs, as this can advance methodological developments in bilevel optimization and promote its practical applications.

A bilevel optimization problem consists of an upper-level problem and a lower-level problem, where the lower-level problem is embedded within the upper-level problem. To address the nested structure of BOPs, existing studies can be classified into two main categories: single-level reduction methods and nested methods. Among them, single-level reduction methods transform the bilevel optimization problem into a single-level formulation [8]. However, single-level reduction methods typically rely on specific properties of the problem. When the lower-level problem is nonconvex or fails to satisfy sufficient regularity conditions, deriving an equivalent single-level reformulation becomes difficult, which limits the applicability of single-level reduction methods to complex BOPs [7,9]. Compared with single-level reduction methods, nested methods can preserve the original structure of BOPs and avoid the dependence on specific function properties typically required by single-level reduction methods, which gives nested methods broader applicability to complex BOPs [10]. By embedding the lower-level solution process into the upper-level process, nested methods have been widely applied [7,11]. In recent years, combining nested methods with intelligent optimization algorithms has gradually become an important direction for solving complex BOPs [12,13]. Meanwhile, with the development of artificial intelligence, new intelligent optimization algorithms have continuously emerged. These include triangulation topology aggregation optimizer (TTAO), a new optimization algorithm [14]. Owing to its simple parameter setting and practical applicability, TTAO has attracted increasing attention and shown potential for solving complex optimization problems [15,16].

Solving BOPs is inherently challenging, as even linear BOPs have been proven to be NP-hard [17,18]. In the real world, these problems usually involve more complex characteristics, such as nonlinearity, nonconvexity, and high dimensionality, which further aggravate the solution difficulty. Nevertheless, effective algorithms designed for complex BOPs remain relatively insufficient. Therefore, CFHTTAO is proposed to solve complex BOPs. CFHTTAO is based on TTAO and integrates the nested hierarchical solution framework with the designed improvement mechanisms to enhance its performance. Furthermore, SMD problems under different dimensional settings are used to evaluate the effectiveness of CFHTTAO, as these problems involve various complex characteristics of BOPs. CFHTTAO is compared with other algorithms developed for solving BOPs, and its effectiveness is further verified through the practical application. The key contributions of this paper are as follows.

- (1) A novel hierarchical coordinated method with triangular topological structures is proposed for solving BOPs. It reorganizes information exchange among triangular topological units, constructs a new coordinated feedback method for the lower level, and establishes differentiated methods tailored to the tasks at the upper and lower levels.
- (2) CFHTTAO is proposed to solve complex BOPs. CFHTTAO adopts a nested hierarchical solution framework and avoids the strict reformulation requirements. It shows broader applicability to complex BOPs involving challenging characteristics such as nonlinearity and nonconvexity.

- (3) A knowledge-sharing mechanism is designed to enhance information exchange among triangular topological units. By constructing a knowledge pool, this mechanism strengthens information communication among different units and helps the algorithm search more promising regions.
- (4) A coordinated feedback mechanism is developed to adaptively select a suitable lower-level search method. In particular, the k -nearest neighbors (KNN)-based approximation technique is adopted to estimate the lower-level solution, and dynamic t -distribution search is designed to further uncover the potential value of the approximate lower-level solution.
- (5) An elite learning evolution mechanism is designed to prevent the algorithm from being trapped in local optima. Through diverse learning strategies, the elite learns from other superior individuals, which enhances its guiding ability and reduces the risk of premature convergence.
- (6) CFHTTAO is evaluated on the SMD test suite under different dimensional settings, because these test problems capture various complex characteristics of BOPs. Experimental results show that CFHTTAO achieves better accuracy, stability, and convergence. Moreover, CFHTTAO maintains its competitiveness as the problem dimension increases, and the Friedman test further confirms the statistical significance of its performance advantage. Finally, CFHTTAO is verified by inverse optimal control problem, displaying its practical applicability for real-world BOPs.

The rest of this paper is organized as follows. Section 2 provides a literature review of the related work. Section 3 describes the standard TTAO. Section 4 elaborates on the proposed CFHTTAO, including the improvement mechanisms developed to enhance its optimization performance, as well as the overall framework and implementation procedure of the algorithm. Section 5 reports the simulation experiments and result analysis to evaluate the performance of CFHTTAO. Section 6 applies CFHTTAO to the practical bilevel optimization problem. Section 7 concludes the paper.

2. Literature Review

BOPs can effectively characterize the interaction between upper-level and lower-level decisions. Solving BOPs faces considerable challenges, especially when they involve complex characteristics such as nonlinearity and nonconvexity, which further increase the solution difficulty. Considering the inherent difficulty and practical importance of BOPs, it is necessary to develop algorithms for BOPs.

As discussed in Section 1, the main methods for handling the nested structure of BOPs are single-level reduction methods and nested methods. Single-level reduction methods usually transform a bilevel problem into a single-level problem through mathematical reformulation, but they are limited by function properties of the problem [10]. In contrast, nested methods can preserve the original structure of BOPs by optimizing the upper-level decision in the outer loop and solving the lower-level response in the inner loop, showing greater flexibility. Nested methods have become an important approach to solve bilevel problems, the combination of nested methods and intelligent optimization algorithms has attracted increasing attention. For example, Ma et al. [19] adopted the nested method and developed a hierarchical hybrid particle swarm optimization (PSO) and differential evolution (DE)-based algorithm to solve BOPs, in which modified PSO is used to solve the upper-level problem and modified DE is used to solve the lower-level problem. Wang et al. [20] proposed a hierarchical modified DE algorithm with nested method for solving BOPs. Wang et al. [21] developed a bilevel improved fruit fly optimization algorithm that preserves the structure of BOPs to solve these problems. Wu et al. [22] proposed a bilevel algorithm by combining the nested method with DE to solve BOPs. Lu et al. [12] developed

a nested bilevel whale optimization algorithm to solve a schedule risk management problem formulated as a BOP. Camacho-Vallejo and Dávila [23] formulated a warehouse location problem with inventory decisions as a BOP and proposed nested evolutionary algorithms to solve it. Wang et al. [24] developed a nested genetic algorithm for a BOP of low-carbon product family design and multi-factory collaborative production configuration.

When single-level reduction methods or nested methods are used to handle the hierarchical structure of BOPs, the specific solution process is usually implemented using exact algorithms or intelligent optimization algorithms. Exact algorithms often face difficulties in solving complex problems. Intelligent optimization algorithms exhibit strong applicability and have been widely used [25,26]. For example, Xiao et al. [27] used an artificial lemming algorithm to address some problems including the parameter identification. Guo et al. [28] employed the PSO to study the precision of the mechanical model. Dang et al. [29] improved an evolutionary algorithm to address multimodal multi-objective problems.

TTAO is a recently proposed optimization algorithm that has attracted increasing attention. For example, Siva Subramanian et al. [15] combined TTAO with a hyperbolic graph wavelet neural network for solving intrusion detection and classification. Dahou et al. [16] developed an improved TTAO-based feature selection algorithm for fake news detection. Abd Elaziz et al. [30] employed TTAO to optimize long short-term memory for solar distiller productivity prediction. Khamies et al. [31] used TTAO to solve complex power-system control problems. Chen et al. [32] applied TTAO to enhance the operational performance of the integrated system under variable environmental conditions. Wang et al. [33] used an improved TTAO to optimize short-term load forecasting. Xu et al. [34] introduced TTAO into a short-term photovoltaic power forecasting framework to optimize the hyperparameters. Qu et al. [35] adopted TTAO to support the secondary decomposition of multivariate load sequences in integrated energy systems, aiming to improve decomposition accuracy and facilitate the extraction of trend and periodic components for load forecasting. Zhang et al. [36] enhanced the original TTAO and validated it on benchmark functions, engineering design problems, and constrained optimization, demonstrating the extensibility of TTAO in practical optimization tasks. Vidhyalakshmi et al. [37] applied TTAO to optimize model parameters, extending the application of TTAO to computer vision and pattern recognition.

Overall, existing studies have shown that the nested method combined with the intelligent optimization algorithm can preserve the original structure of BOPs and show broader applicability to complex BOPs. In addition, TTAO has been successfully applied to complex problems. Although progress has been made in solving BOPs, algorithms developed for BOPs with complex characteristics remain relatively insufficient. The existing studies still have limitations. For example, single-level reduction methods usually depend on specific mathematical properties of BOPs, and their applicability may be limited when the lower-level problem is nonconvex or does not satisfy regularity conditions. Moreover, existing TTAO-related studies mainly apply TTAO to forecasting, control optimization, and other single-level optimization tasks, while its potential for solving BOPs has not been sufficiently investigated. Therefore, to better solve complex BOPs, this paper proposes CFHTTAO by integrating TTAO with a nested hierarchical solution framework and designing knowledge-sharing, coordinated feedback, and elite learning evolution mechanisms to improve its performance on complex BOPs.

3. The Original TTAO Algorithm

TTAO is a mathematics-based metaheuristic inspired by triangular topological structures [14]. The triangle is the most fundamental and stable structure. TTAO mainly comprises initialization, triangular topological unit formation, generic aggregation, and

local aggregation. In the initialization stage, the population size N is partitioned into $\lfloor N/3 \rfloor$ triangular topological units, where $\lfloor \cdot \rfloor$ denotes rounding down to the nearest integer. Specifically, if N is divisible by 3, the N is divided into $N/3$ units. Otherwise, the N is divided into $\lfloor N/3 \rfloor$ units, and the remaining one or two individuals are randomly generated within the feasible region.

Each triangular topological unit consists of different vertexes, and the first vertex of each unit is randomly generated during initialization stage by

$$x_{i,1} = r_0 \times (ub - lb) + lb, \quad (1)$$

where $x_{i,1}$ is the first vertex of the i -th unit. r_0 represents a random number in the interval $[0, 1]$. The upper and lower bounds of the search space are denoted by ub and lb , respectively. For a D -dimensional decision vector, they can be expressed as $ub = (ub_1, ub_2, \dots, ub_D)$ and $lb = (lb_1, lb_2, \dots, lb_D)$.

For each triangular topological unit formation, the second and third vertexes are generated from the first vertex and the direction vector. They are formulated as

$$\begin{cases} x_{i,2} = x_{i,1} + l \times \cos(\theta), \\ x_{i,3} = x_{i,1} + l \times \cos\left(\theta + \frac{\pi}{3}\right), \end{cases} \quad (2)$$

where $x_{i,2}$ and $x_{i,3}$ denote the second and third vertexes of the i -th unit, respectively. θ is a random vector where each element is a random number in the range $[0, \pi]$. l represents the size of a unit. l is calculated as

$$l = 9 \times e^{-t/T}, \quad (3)$$

where t denotes the current iteration number and T denotes the maximum number of iterations.

The fourth vertex is generated by using three vertexes information in a single unit. The formula for the fourth vertex is presented as

$$x_{i,4} = r_1 \times x_{i,1} + r_2 \times x_{i,2} + r_3 \times x_{i,3}, \quad (4)$$

where $x_{i,4}$ denotes the fourth vertex of the i -th unit. r_1 , r_2 , and r_3 are randomly generated numbers between $[0, 1]$, and their sum equals one. $x_{i,1}$, $x_{i,2}$, and $x_{i,3}$ are the first, second, and third vertices used to construct the triangular structure. $x_{i,4}$ is the fourth vertex generated from $x_{i,1}$, $x_{i,2}$, and $x_{i,3}$, rather than a part of the triangle itself.

Generic aggregation focuses on exploration by exchanging information between the optimal vertex of the current unit and the optimal vertex of a randomly selected unit to generate the new vertex. Subsequently, the generated new vertex is compared with the optimal or suboptimal vertex of the unit, and the optimal or suboptimal vertex of the unit is updated. The formula for the new vertex in the generic aggregation phase is defined as

$$x_{i,new1} = r_4 \times x_{i,best} + (1 - r_4) \times x_{rand,best}, \quad (5)$$

where $x_{i,new1}$ represents the generated vertex in the generic aggregation phase for the unit i . r_4 is a random number between $[0, 1]$. $x_{i,best}$ and $x_{rand,best}$ refer to the best vertexes for the unit i and a randomly selected unit, respectively.

Local aggregation focuses on the exploitation phase. In this phase, the new vertex is generated by using the positional difference between the best and second-best vertices of the current unit. The generated new vertex in the local aggregation stage is compared with the optimal vertex of the unit, and the superior one is selected as the updated optimal vertex of the unit. The expression for the new vertex in local aggregation phase is written as

$$\mathbf{x}_{i,new2} = \mathbf{x}_{i,best} + a \times (\mathbf{x}_{i,best} - \mathbf{x}_{i,sbest}), \quad (6)$$

where $\mathbf{x}_{i,new2}$ denotes the generated vertex in the local aggregation phase for the unit i . $\mathbf{x}_{i,sbest}$ is the suboptimal vertex for the unit i . a adjusts the aggregation scope size and is calculated as

$$a = \ln\left(\frac{e - e^3}{T - 1}t + e^3 - \frac{e - e^3}{T - 1}\right), \quad (7)$$

where t denotes the current iteration number and T denotes the maximum number of iterations.

4. The Proposed CFHTTAO Algorithm

In this section, improvement mechanisms are designed in CFHTTAO to enhance the optimization performance. Meanwhile, the overall framework and execution procedure of CFHTTAO are presented. For clarity, the upper-level and lower-level solution frameworks of CFHTTAO for BOPs are illustrated separately.

4.1. Knowledge-Sharing Mechanism

Knowledge sharing refers to the process by which knowledge, experience, and other relevant information are communicated, shared, transferred, or coordinated among individuals or units [38,39]. Inspired by [40,41], here, a knowledge-sharing mechanism is designed to enhance information exchange among units and generate new vertexes by sharing superior information across different units.

In the original TTAO, the fourth vertex is generated by Equation (4) based on the three vertexes in a single unit. However, this original generation approach has some limitations. When all three vertexes of a unit are located in a low-quality region, the fourth vertex generated solely from intra-unit information can hardly break away from such a region. As a result, this original approach is not conducive to exploring more promising regions.

The knowledge-sharing mechanism is used to improve the generation approach of the fourth vertex in each unit. Specifically, the best vertex is first extracted from the existing vertexes in each unit to construct a knowledge pool. This knowledge pool carries the superior information of each unit. Subsequently, three vertexes are randomly selected from the pool to help generate the fourth vertex from a combination of well-performing vertexes. Finally, the fourth vertex of each unit is generated using the randomly selected three vertexes from the pool, and its expression is given by Equation (8). Figure 1 shows the process by which four units generate their respective fourth vertexes using the knowledge-sharing mechanism.

$$\mathbf{x}_{i,4} = r_1 \times \mathbf{x}_{\varphi}^p + r_2 \times \mathbf{x}_{\delta}^p + r_3 \times \mathbf{x}_{\eta}^p, \quad (8)$$

where $\mathbf{x}_{\varphi}^p$, $\mathbf{x}_{\delta}^p$, and $\mathbf{x}_{\eta}^p$ denote the three different vertexes randomly selected from the knowledge pool, respectively.

As shown in Figure 1, this mechanism breaks the boundaries among units and enables the sharing and utilization of high-quality information across different units. Each unit first contributes its best vertex to the knowledge pool. Then, three distinct vertexes are randomly selected from the knowledge pool, and the position of the fourth vertex for each unit is calculated according to Equation (8). Unlike the original method that only utilizes the internal information of a single unit to generate the fourth vertex, the fourth vertex generated by the knowledge-sharing mechanism integrates beneficial knowledge from different units, which alleviates the limitations of relying solely on intra-unit information and helps generate competitive vertex positions.

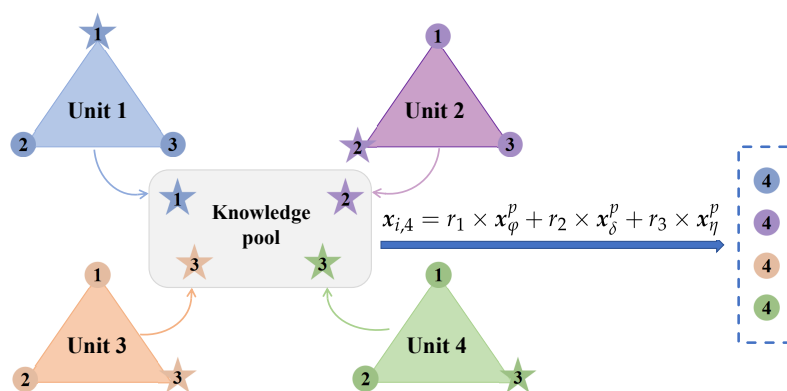


Figure 1. Schematic diagram of knowledge-sharing mechanism.

4.2. KNN-Based Approximation Technique

KNN-based approximation can help solve the lower-level problem of BOPs [22,42]. The KNN-based approximation technique is adopted by exploiting the archive that preserves historical search experience. The obtained approximate lower-level solution is used in the following Section 4.3 to assist the lower-level optimization process of CFHTTAO.

Specifically, the archive records feasible and promising upper-lower solution pairs generated during the optimization process, where each pair consists of an upper-level solution and its corresponding lower-level solution. When an upper-level solution is passed to the lower-level optimization, the algorithm first retrieves relevant historical samples from the archive instead of immediately initiating a new lower-level search. For a given upper-level solution, if the archive is not empty, KNN is employed to approximate its corresponding lower-level solution. Subsequently, based on the obtained approximate lower-level solution, further exploration is conducted in the following Section 4.3 to improve the lower-level optimization.

Based on the above elaboration, when an upper-level solution x^u is passed to the lower level, the process of obtaining its approximate lower-level solution can be formulated as follows [22]. Firstly, let the archive be

$$\mathcal{L} = \left\{ (x_1^u, x_1^l), (x_2^u, x_2^l), \dots, (x_S^u, x_S^l) \right\}, \quad (9)$$

where x_1^u , x_2^u , and x_S^u represent the upper-level solutions for the first, second, and S -th solution pairs in the archive, respectively. x_1^l , x_2^l , and x_S^l represent their corresponding lower-level solutions, respectively.

Subsequently, the k upper-level solutions closest to x^u are selected from the archive, and the corresponding lower-level solutions are used to construct the approximate lower-level solution associated with x^u . Specifically, the distances between x^u and these k upper-level solutions are calculated, and the inverse distance weighting strategy [43] is then used to weight and sum the corresponding k lower-level solutions to obtain the approximate lower-level solution with respect to x^u . When an upper-level solution stored in the archive has zero distance from x^u , that is, when an identical upper-level solution to x^u exists in the archive, the corresponding lower-level solution in the archive is regarded as the approximate lower-level solution. Accordingly, the KNN-based approximate lower-level solution can be expressed as

$$x^l = \begin{cases} \frac{\sum_{k=1}^K w_k x_k^l}{\sum_{k=1}^K w_k}, & \text{if } d(x^u, x_k^u) \neq 0, \quad k = 1, \dots, K, \\ x_k^l, & \text{if } d(x^u, x_k^u) = 0, \end{cases} \quad (10)$$

where $d(x^u, x_k^u)$ denotes the distance between the upper-level solution x^u and the k -th nearest upper-level solution in the archive. w_k denotes the inverse distance weighting and is calculated by $w_k = 1 / [d(x^u, x_k^u)]^p$, where p is the power parameter.

The main implementation of the KNN-based approximation technique is described as follows. When an upper-level solution x^u is passed to the lower level, the archive $\mathcal{L}$ is first checked. If the archive is empty, the KNN-based approximation is not performed. Otherwise, the distances between x^u and the archived upper-level solutions are calculated, and the k nearest archived upper-level solutions are selected. Then, their corresponding lower-level solutions are used to generate the approximate lower-level solution $\hat{x}^l$ through the inverse distance weighting strategy in Equation (10). If an identical upper-level solution exists in the archive, its corresponding lower-level solution is directly used as $\hat{x}^l$. The obtained $\hat{x}^l$ is then used in the coordinated feedback mechanism to assist the lower-level search.

4.3. Coordinated Feedback Mechanism

In a BOP, the lower-level problem is solved according to the upper-level solution, and the resulting lower-level solution is fed back to the upper level [21]. Inspired by [44,45], a coordinated feedback mechanism is designed to improve the flexibility of lower-level optimization. By utilizing the archive, the mechanism integrates different lower-level search methods, adaptively selects a suitable method for lower-level optimization, and feeds the obtained lower-level solution back to the upper level.

The coordinated feedback mechanism divides the lower-level search into two main methods according to the archive. If the archive is empty, the approximate lower-level solution cannot be generated, and the improved TTAO with the knowledge-sharing mechanism described in Section 4.1 is adopted. Otherwise, the archive provides solution pairs for generating the KNN-based approximate lower-level solution, and the population is divided into two parts to enrich search diversity. One part retains the improved TTAO, while the other executes the dynamic t -distribution search strategy combined with the KNN-based approximate lower-level solution. Figure 2 is a schematic diagram of the coordinated feedback mechanism.

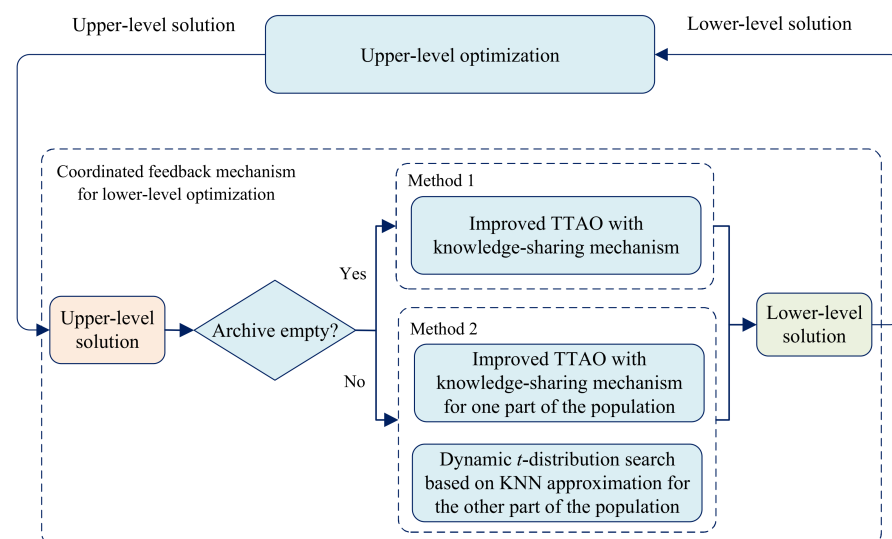


Figure 2. Schematic diagram of the coordinated feedback mechanism.

A dynamic t -distribution search strategy is designed within the coordinated feedback mechanism to generate lower-level solutions, aiming to further exploit the potential of the KNN-based approximate lower-level solution. Existing studies indicate that the Cauchy distribution $C(0, 1)$ is equivalent to the t -distribution with one degree of freedom, whereas the t -distribution gradually approaches the Gaussian distribution $N(0, 1)$ as the degrees of

freedom tend to infinity [46]. Moreover, Cauchy mutation has strong global exploration capability, whereas Gaussian mutation exhibits strong local exploitation capability [46,47]. In this strategy, the iteration number gen is used as the degrees of freedom parameter of the t -distribution, which integrates the advantages of Cauchy mutation and Gaussian mutation. In Figure 3, the probability density curves of the Gaussian distribution $N(0,1)$, the t -distributions with degrees of freedom $gen = 2$ and $gen = 3$, and the Cauchy distribution $C(0,1)$ are illustrated. The t -distribution with $gen = 2$ is closer to the Cauchy distribution, while the t -distribution with $gen = 3$ gradually shifts toward the Gaussian distribution. In the early stage of iteration, the dynamic t -distribution tends to resemble the Cauchy distribution, which helps explore a wider region of the search space. As the iteration number increases, the degrees of freedom of the dynamic t -distribution gradually increase, and its shape evolves toward that of a Gaussian distribution. At this stage, the mutation operator exhibits stronger local perturbation capability, which helps the algorithm perform refined exploitation. The dynamic t -distribution search strategy can be formulated as

$$x_j = \hat{x}^l + t^g \times (\hat{x}^l - r \times x_n^l) \times F, \quad (11)$$

where t^g denotes the dynamic t -distribution search operator with the iteration number gen used as the degree of freedom. $\hat{x}^l$ represents the KNN-based approximate lower-level solution obtained by Equation (10). r is a random number within $[0,1]$. x_n^l represents the archived lower-level solution, which corresponds to the archived upper-level solution closest to the current upper-level solution. F is used to restrict the search within a reasonable range, where $F = c(ub - lb)$, and c is a coefficient.

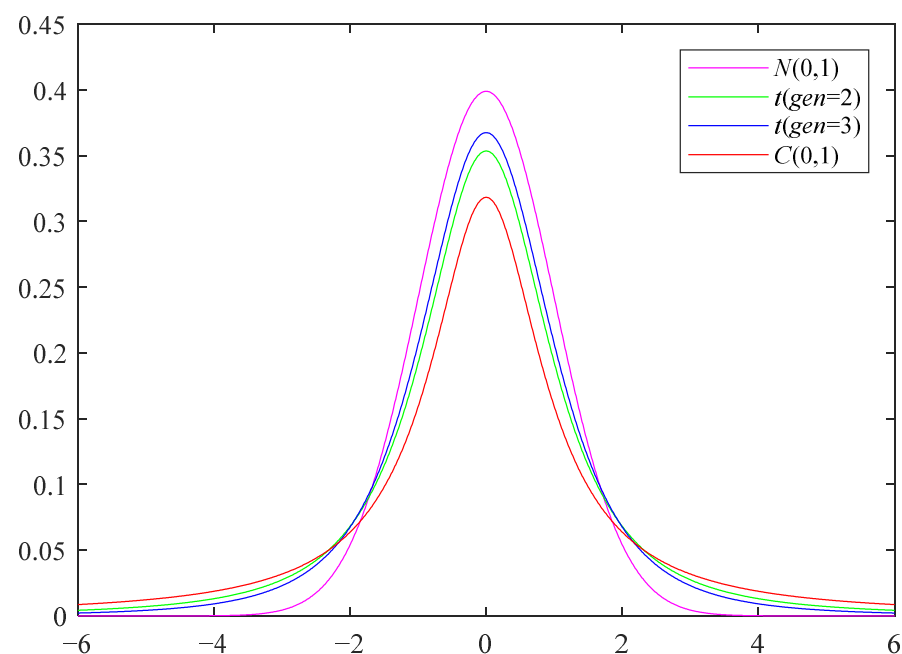


Figure 3. The probability density curves.

In summary, the coordinated feedback mechanism coordinates two main lower-level optimization methods by utilizing archive information. The first method employs improved TTAO, and the second method divides the population into two parts, with one retaining the first method and the other using a dynamic t -distribution search strategy. This mechanism coordinates lower-level search methods to obtain the lower-level solution and feeds it back to the upper-level optimization.

4.4. Elite Learning Evolution Mechanism

In general, elites are regarded as superior individuals that play a guiding role [48,49]. Here, the current optimal solution of BOP obtained by an algorithm is considered an elite. The effectiveness of the elite plays an important role, and a less competitive elite will increase the risk of the algorithm being trapped in a local optimum [50]. Therefore, the elite learning evolution mechanism is designed to drive the learning and evolution of elites by leveraging valuable information, which guides population development and enhances the overall solution efficiency of the algorithm.

The elite learning evolution mechanism is developed based on the knowledge pool constructed in Section 4.1. This pool is constructed from the best vertex selected from each unit and therefore contains representative and competitive information from different units. By selecting vertexes from the knowledge pool to perturb the elite, the mechanism helps the elite absorb valuable information. In addition, to enrich the learning process of elites and reduce the risk of the algorithm being trapped in a local optimum, perturbation strategies are designed from different perspectives in the elite learning evolution mechanism. First, considering the differences between the elite and other individuals, and that the information of the knowledge pool can help compensate for the shortcomings of the elite, the following strategy is constructed.

$$\tilde{g} = w \times g + r^2 \times (g - x_v^p), \quad (12)$$

where g denotes the original elite. x_v^p is the v -th vertex randomly selected from the knowledge pool. w is the inertia weight, decreasing linearly from $w_{\max}$ to $w_{\min}$ [51].

Moreover, to exploit the valuable information among different individuals and enrich the learning mode of the elite, the following strategy is further constructed.

$$\tilde{g} = w \times g + r^2 \times (x_m^p - x_h^p), \quad (13)$$

where x_m^p and x_h^p represent the m -th and h -th vertex randomly selected from the knowledge pool, respectively.

Therefore, according to Equations (12) and (13), the elite is perturbed by randomly selecting one of the strategies, and the formulations can be summarized as follows.

$$\tilde{g} = \begin{cases} w \times g + r^2 \times (g - x_v^p), & \text{if } rand < \rho, \\ w \times g + r^2 \times (x_m^p - x_h^p), & \text{otherwise.} \end{cases} \quad (14)$$

where $rand$ is a random number within $[0, 1]$, which is used to control the selection between the two strategies mentioned above. ρ is a selection probability. It is worth noting that, to ensure that the elite evolves in a favorable direction, the $\tilde{g}$ is further compared with the original elite g , and the better one is selected as the updated elite to guide the population toward more promising regions.

4.5. Description of CFHTTAO

To better solve BOPs, the improvement mechanisms designed in Sections 4.1–4.4 are employed to enhance the performance of CFHTTAO. In addition, the nested method is commonly used for BOPs [7]. Compared with single-level reduction methods, the nested method is not restricted by problem properties and shows stronger applicability to BOPs. Therefore, the designed mechanisms are integrated with a nested hierarchical framework to develop CFHTTAO for complex BOPs.

4.5.1. Framework of CFHTTAO

In the framework of CFHTTAO, the optimization processes of the upper and lower levels are integrated through a nested method, forming a bilevel algorithm with the hierarchical framework. To provide a clearer description of CFHTTAO, its overall framework comprises two interdependent components and interactive components, namely the upper-level framework and the lower-level framework. Within this framework, the lower-level framework is nested inside the upper-level framework, and together they constitute the overall framework of CFHTTAO.

In a BOP, the lower-level problem is embedded within the upper-level problem, where the upper-level problem can be regarded as the outer optimization task and the lower-level problem as the inner optimization task [52]. The overall framework of CFHTTAO for solving the BOP is illustrated in Figure 4. In Figure 4, x^u denotes the upper-level solution generated by the upper-level framework and passed to the lower-level framework. For a given x^u , x^{l*} represents the corresponding optimal lower-level solution obtained by the lower-level framework of CFHTTAO, which is then fed back to the upper level. Based on the lower-level feedback, the upper-level framework of CFHTTAO further determines the optimal upper-level solution x^{u*} . (x^{u*}, x^{l*}) represents the final optimal solution of the BOP obtained by CFHTTAO.

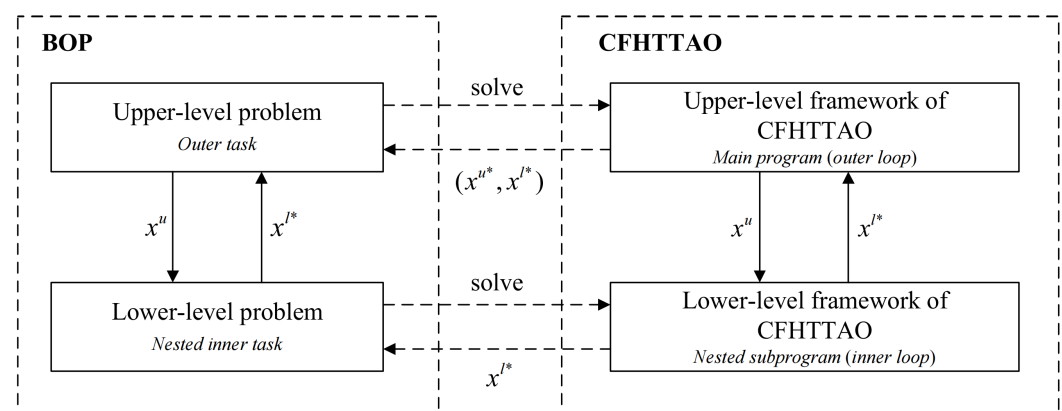


Figure 4. The overall framework of CFHTTAO for solving BOPs.

In addition, CFHTTAO incorporates the improvement mechanisms designed above to enhance its capability for solving BOPs. By combining the nested method with the designed mechanisms, the upper-level framework and lower-level framework of CFHTTAO are further described in Sections 4.5.2 and 4.5.3, respectively. Similar to [20,21], CFHTTAO adopts a nested hierarchical framework for solving BOPs, in which the upper-level framework serves as the main program, while the lower-level framework acts as a subprogram. Accordingly, an upper-level population and a lower-level population are employed to perform searches at the upper and lower levels, respectively, and their iterative processes are controlled by the corresponding maximum numbers of iterations [12].

4.5.2. Upper-Level Framework of CFHTTAO

The upper-level framework of CFHTTAO serves as the main program of the algorithm, and its completion indicates the termination of the entire solution process. The lower-level framework is nested within the upper-level framework as a subprogram and is invoked by the upper-level framework. Furthermore, CFHTTAO integrates a knowledge-sharing mechanism and an elite learning evolution mechanism at the upper level to enhance the performance of the original TTAO. The main steps and pseudocode Algorithm 1 of the upper-level framework of CFHTTAO are as follows.

Step 1: Set parameters: upper-level population size N_u , maximum number of upper-level iterations T_u , lower-level population size N_l , and maximum number of lower-level iterations T_l .

Step 2: Divide the upper-level population into upper-level triangular topological units. Initialize the first vertex $x_{i,1}^u$ in each upper-level unit using Equation (1).

Step 3: Calculate the second vertex $x_{i,2}^u$ and the third vertex $x_{i,3}^u$ in each upper-level unit by Equation (2). Pass the upper-level solutions $x_{i,1}^u$, $x_{i,2}^u$, and $x_{i,3}^u$ separately to the lower-level framework of CFHTTAO to obtain the corresponding optimal lower-level solutions. Then, the corresponding lower-level solutions $x_{i,1}^{l*}$, $x_{i,2}^{l*}$, and $x_{i,3}^{l*}$ are sequentially fed back to the upper-level framework of CFHTTAO.

Step 4: Calculate the fourth vertex $x_{i,4}^u$ of each upper-level unit using Equation (8) of the knowledge-sharing mechanism.

Step 5: Conduct generic aggregation and local aggregation on each upper-level unit using Equation (5) and Equation (6), respectively.

Step 6: Perform random search on the remaining individuals when isolated individuals remain after forming upper-level units.

Step 7: Generate $x_{i,1}^u$ for the next iteration based on the knowledge pool.

Step 8: Determine the current optimal solution for the BOP. Perturb the original best solution using Equation (14), compare the perturbed solution with the original one, and retain the better solution.

Step 9: Check whether the maximum number of upper-level iterations has been reached. If the termination condition is not met, return to Step 3. Otherwise, terminate the procedure and output the final optimal solution, indicating the end of the main program.

Algorithm 1 Main program: Upper-level pseudocode of CFHTTAO

```

1: Initialize  $N_u$ ,  $T_u$ ,  $N_l$ , and  $T_l$ 
2: Divide the upper-level population into  $M_u$  units and initialize  $x_{i,1}^u$  using Equation (1)
3: while  $t_u < T_u$  do
4:   for  $i = 1$  to  $M_u$  do
5:     Generate  $x_{i,2}^u$  and  $x_{i,3}^u$  using Equation (2)
6:     Invoke Algorithm 2 to obtain the corresponding lower-level solutions
7:     Calculate  $x_{i,4}^u$  using Equation (8)
8:     Conduct generic and local aggregation using Equations (5) and (6)
9:   end for
10:  Perform random search on the remaining individuals after forming  $M_u$  units
11:  Generate  $x_{i,1}^u$  for the next iteration
12:  Record the original optimal solution at the current iteration
13:  Perturb the original optimal solution using Equation (14) and update it
14: end while
15: Output the optimal solution  $(x^{u*}, x^{l*})$  for the BOP

```

4.5.3. Lower-Level Framework of CFHTTAO

In CFHTTAO, the lower-level framework is implemented as a nested subprogram within the upper-level framework and is invoked by the main program. The termination of this subprogram indicates that the lower-level optimization has been completed and the corresponding result has been fed back to the upper level for continued execution of the main program. Within the lower-level framework, the archive is used to determine whether a KNN-based approximate solution can be generated. Based on the archive, the coordinated feedback mechanism adaptively selects between the two main lower-level optimization methods and returns the best lower-level solution to the upper level as the response to the given upper-level solution. The main steps and pseudocode Algorithm 2

are presented below. Because the lower-level framework is nested within the upper-level framework, its steps are denoted as sub-steps.

Sub-step 1: For the upper-level solution x^u passed to the lower-level framework of CFHTTAO, the coordinated feedback mechanism is executed to determine which lower-level optimization method should be adopted based on the archive. If an approximate lower-level solution cannot be generated from the archive, Sub-step 2 is executed; otherwise, Sub-step 3 is executed.

Sub-step 2: Execute the first lower-level optimization method. The detailed procedure is described in Sub-steps 2.1–2.7.

Sub-step 2.1: Divide the lower-level population size into lower-level triangular topological units. Initialize the first vertex $x_{i,1}^l$ of each lower-level unit using Equation (1).

Sub-step 2.2: Calculate the second vertex $x_{i,2}^l$ and the third vertex $x_{i,3}^l$ in each lower-level unit using Equation (2).

Sub-step 2.3: Calculate the fourth vertex $x_{i,4}^l$ in each lower-level unit according to Equation (8) in the knowledge-sharing mechanism.

Sub-step 2.4: Conduct generic aggregation and local aggregation on each lower-level unit using Equations (5) and (6), respectively.

Sub-step 2.5: Perform random search on the remaining individuals when isolated individuals remain after forming lower-level units.

Sub-step 2.6: Generate $x_{i,1}^l$ for next iteration based on the knowledge pool.

Sub-step 2.7: Record the optimal lower-level solution for the current lower-level iteration and go to Sub-step 4.

Sub-step 3: Execute the second lower-level optimization method. The detailed procedure is described in Sub-steps 3.1–3.4.

Sub-step 3.1: Divide the lower-level population into two parts. One part conducts Sub-step 3.2, while the other part goes to Sub-step 3.3.

Sub-step 3.2: Calculate the KNN-based approximate lower-level solution using Equation (10). Then, according to Equation (11), perform the lower-level search on a portion of the population.

Sub-step 3.3: Divide the other portion of the population into lower-level units. Initialize the first vertex $x_{i,1}^l$ of each lower-level unit, and then perform the subsequent operations following the same procedure described in Sub-steps 2.2–2.6.

Sub-step 3.4: Compare the most competitive lower-level solutions generated in Sub-step 3.2 and Sub-step 3.3, select the best one as the optimal lower-level solution of the current lower-level iteration, and execute Sub-step 4.

Sub-step 4: Check whether the maximum number of lower-level iterations has been reached. If not, continue with Sub-step 2 or Sub-step 3 according to the archive information. Otherwise, return the optimal lower-level solution x^{l*} corresponding to the given upper-level solution to the upper-level framework of CFHTTAO.

Algorithm 2 Subprogram: Lower-level pseudocode of CFHTTAO

```

1: /* Coordinated feedback mechanism */
2: /* The first method */
3: if archive is empty then
4:   Divide the lower-level population into  $M_l$  units and initialize  $x_{i,1}^l$  using Equation (1)
5:   while  $t_l < T_l$  do
6:     for  $i = 1$  to  $M_l$  do
7:       Generate  $x_{i,2}^l$  and  $x_{i,3}^l$  using Equation (2)
8:       Calculate  $x_{i,4}^l$  using Equation (8)
9:       Conduct generic and local aggregation using Equations (5) and (6)
10:    end for
11:    Perform random search on the remaining individuals after forming  $M_l$  units
12:    Generate  $x_{i,1}^l$  for the next iteration
13:    Record the optimal lower-level solution at the current iteration
14:  end while
15: else
16:   /* The second method */
17:   Calculate the KNN-based approximate lower-level solution  $\hat{x}^l$  using Equation (10)
18:   Equally split the lower-level population into two groups
19:   Use one group as  $N_k$  individuals and construct  $M_k$  units from the other group
20:   while  $t_l < T_l$  do
21:     for  $j = 1$  to  $N_k$  do
22:       Calculate  $x^l$  using Equation (11)
23:     end for
24:     for  $i = 1$  to  $M_k$  do
25:       Generate  $x_{i,2}^l$  and  $x_{i,3}^l$  using Equation (2)
26:       Calculate  $x_{i,4}^l$  using Equation (8)
27:       Conduct generic and local aggregation using Equations (5) and (6)
28:     end for
29:     Perform random search on the remaining individuals after forming  $M_k$  units
30:     Generate  $x_{i,1}^l$  for the next iteration
31:     Record the optimal lower-level solution at the current iteration
32:   end while
33: end if
34: Return the optimal lower-level solution  $x^{l*}$  to the upper-level framework of CFHTTAO

```

5. Simulation Experiment and Result Analysis

To evaluate the performance of CFHTTAO for solving BOPs, the bilevel test problems known as the SMD test suite are employed in the simulation experiments. The SMD test suite consists of 12 problems, denoted SMD1–SMD12, which are designed to capture diverse challenges encountered in BOPs [53]. By introducing various complex characteristics such as nonlinearity, multimodality and nonconvexity, SMD problems pose great difficulties for algorithms seeking optimal solutions [54]. In addition, the SMD test suite has been used to evaluate the performance of algorithms for solving BOPs [22,55].

In this section, the performance of CFHTTAO is evaluated through experimental tests. First, an ablation study is conducted to examine the effectiveness of the designed improvement mechanisms, including the knowledge-sharing mechanism, coordinated feedback mechanism, and elite learning evolution mechanism. Then, CFHTTAO is compared with other algorithms developed for solving BOPs, namely the hierarchical hybrid particle swarm optimization and differential evolution based algorithm (HHPSODE) [19], hierarchical modified differential evolution algorithm (HMDE) [20], bilevel improved fruit fly optimization algorithm (BIFOA) [21], self-adaptive bilevel differential evolution algorithm (SABILDE) [22], and bilevel whale optimization algorithm (BiWOA) [12]. Moreover, the optimal solutions obtained by CFHTTAO for all SMD problems under different dimen-

sional settings are visualized. Furthermore, the convergence performance of CFHTTAO is examined through convergence curves, and the Friedman test is conducted for statistical analysis. Finally, the computational cost of the algorithms is analyzed by recording the number of function evaluations and CPU time. The detailed experimental results are presented as follows.

5.1. Experimental Setting

To ensure the fairness and comparability of the experimental results, all algorithms are independently run 20 times under the same experimental environment [21]. The population size is set to 30, and the maximum numbers of iteration is set to 100 for all algorithms. In addition, for CFHTTAO, the maximum weight $w_{\max}$ and minimum weight $w_{\min}$ are set to 0.9 and 0.4, respectively, [51], the selection probability ρ is set to 0.5 [12], and the power parameter p is set to 2 [22]. The parameter settings of the other algorithms are kept consistent with those reported in their original references, and the corresponding settings are shown in Table 1. All experimental results are rounded to two decimal places.

Table 1. Parameter settings of other algorithms.

Algorithm	Parameter Setting
HHPSODE	Learning factors: $c_1 = c_2 = 2.0$; maximum inertia weight: $\omega_{\max} = 0.9$; minimum inertia weight: $\omega_{\min} = 0.4$; scaling factor: $F = 0.5$; crossover rate: $CR = 0.9$ [19].
HMDE	Scaling factor: $F \in [0.4, 0.9]$; crossover rate: $CR \in [0.4, 0.9]$ [20].
BIFOA	Maximum search radius: $\lambda_j^{\max} = (UB_j - LB_j)/2$; minimum search radius: $\lambda_j^{\min} = 1.0 \times 10^{-5}$ [21].
SABiLDE	Scaling factor for DE/rand/1: $F_r^u = 0.5$; crossover rate for DE/rand/1: $CR_r^u = 0.1$; scaling factor for DE/best/1: $F_b^u = 0.5$; crossover rate for DE/best/1: $CR_b^u = 0.9$; scaling factor for DE/target-to-best/1: $F^l = 0.5$; crossover rate for DE/target-to-best/1: $CR^l = 0.9$; control parameter for lower population size: $c_p = 10$; control parameter for lower search radius: $c_r = 3$; probability: $\tau = 0.7$; power parameter: $p = 2$ [22].
BiWOA	Upper-level coefficient of nonlinear convergence factor: $k_T = 0.3$; lower-level coefficient of nonlinear convergence factor: $k_B = 0.2$ [12].

Computational accuracy is used as an important performance evaluation metric, and represents the absolute deviation between the function value obtained by the algorithm and the actual optimal function value [56]. A smaller absolute deviation indicates higher computational accuracy of the algorithm. Therefore, upper-level accuracy and lower-level accuracy are measured by the absolute deviations between the function values obtained by algorithm and actual optimal function values at the upper and lower levels, respectively. Let UF and LF denote the upper-level and lower-level function values obtained by the algorithm, respectively, and let UF^* and LF^* denote the corresponding actual optimal upper-level and lower-level function values, respectively. Similar to [56], the upper-level accuracy metric and lower-level accuracy metric are defined as $UA = |UF - UF^*|$ and $LA = |LF - LF^*|$, respectively. UA and LA provide an intuitive measure of the solution performance of the algorithm. Smaller values of UA and LA indicate smaller deviations between the function values obtained by the algorithm and the actual optimal function values at the upper and lower levels, respectively, implying better algorithmic performance.

All experiments were conducted under the same experimental conditions in MATLAB R2017b on a computer equipped with an Intel(R) Core(TM) i7-12650H CPU @ 2.30 GHz processor, 16.00 GB of RAM, and Windows 11 64-bit as the Operating System. All parameters in MATLAB R2017b were set to their default values.

5.2. Ablation Study of Mechanisms

To evaluate the effectiveness of the improvement mechanisms in CFHTTAO, an ablation study is conducted. Since the original TTAO is designed for single-level optimization problems, it cannot be directly applied to bilevel optimization problems without considering the hierarchical interaction between the upper and lower levels. Therefore, in this ablation study, HTTAO is constructed based on the original TTAO and the nested hierarchical solution framework to address the upper–lower structure of BOPs, which serves as the baseline algorithm in the ablation study. On this basis, three variants are further constructed by independently incorporating the designed improvement mechanisms. Specifically, the CFHTTAO and its related ablation variants are defined in Table 2.

Table 2. Descriptions of CFHTTAO and its related ablation variants.

Algorithm	Description
HTTAO	TTAO with the nested hierarchical solution framework
HTTAO-KS	HTTAO with the knowledge-sharing mechanism
HTTAO-CF	HTTAO with the coordinated feedback mechanism
HTTAO-EL	HTTAO with the elite learning evolution mechanism
CFHTTAO	HTTAO with all three mechanisms

To ensure experimental fairness, HTTAO, HTTAO-KS, HTTAO-CF, HTTAO-EL, and CFHTTAO are tested under the same experimental environment. Taking SMD problems under a $(2 + 3)$ -dimensional setting as an example, Table 3 reports the average (*avg*) and standard deviation (*std*) values of the *UA* and *LA* obtained by HTTAO, HTTAO-KS, HTTAO-CF, HTTAO-EL, and CFHTTAO. Furthermore, the Friedman test is adopted to record the mean rank (*MR*) and rank (*R*) of the algorithms and further assess the relative importance of each improvement mechanism in Table 4, where UL and LL denote upper-level and lower-level optimization, respectively.

As shown in Table 3, HTTAO-KS, HTTAO-CF, and HTTAO-EL generally achieve smaller *UA* and *LA* values than HTTAO, indicating the positive contribution of the three improvement mechanisms. Specifically, the knowledge-sharing mechanism improves the utilization of superior information among triangular topological units, the coordinated feedback mechanism strengthens the interaction between upper-level and lower-level optimization, and the elite learning evolution mechanism enhances the guiding ability of elite solutions. Moreover, CFHTTAO obtains highly competitive results on most test problems, demonstrating that the integration of the three mechanisms can further improve the solution accuracy of HTTAO.

Table 3. Experimental results of CFHTTAO and its related variants for SMD(2 + 3) problems.

Problem		HTTAO		HTTAO-KS		HTTAO-CF		HTTAO-EL		CFHTTAO	
		UA	LA	UA	LA	UA	LA	UA	LA	UA	LA
SMD1	<i>avg</i>	5.98×10^{-3}	4.94×10^{-3}	2.42×10^{-6}	2.04×10^{-6}	2.57×10^{-7}	1.64×10^{-5}	5.51×10^{-8}	3.39×10^{-8}	7.79×10^{-22}	3.95×10^{-22}
	<i>std</i>	2.22×10^{-3}	4.04×10^{-3}	1.14×10^{-6}	1.13×10^{-6}	2.74×10^{-7}	1.65×10^{-5}	6.05×10^{-8}	3.73×10^{-8}	1.04×10^{-21}	4.81×10^{-22}
SMD2	<i>avg</i>	5.25×10^{-7}	4.46×10^{-5}	1.64×10^{-7}	1.63×10^{-8}	4.75×10^{-8}	1.30×10^{-6}	2.59×10^{-8}	3.46×10^{-5}	1.29×10^{-9}	1.64×10^{-7}
	<i>std</i>	5.74×10^{-7}	1.37×10^{-5}	2.04×10^{-7}	2.05×10^{-7}	6.45×10^{-8}	2.10×10^{-6}	1.92×10^{-8}	3.77×10^{-5}	1.89×10^{-9}	1.99×10^{-7}
SMD3	<i>avg</i>	2.46×10^{-6}	1.34×10^{-8}	1.74×10^{-7}	1.83×10^{-5}	1.52×10^{-12}	1.52×10^{-12}	1.38×10^{-14}	8.30×10^{-14}	4.44×10^{-29}	8.76×10^{-31}
	<i>std</i>	4.11×10^{-6}	6.43×10^{-9}	2.59×10^{-8}	2.74×10^{-5}	2.17×10^{-12}	2.17×10^{-12}	1.12×10^{-14}	8.87×10^{-14}	4.25×10^{-29}	1.79×10^{-30}
SMD4	<i>avg</i>	1.23×10^{-5}	1.23×10^{-4}	5.33×10^{-9}	8.95×10^{-6}	1.61×10^{-13}	7.35×10^{-9}	5.67×10^{-14}	1.05×10^{-14}	5.88×10^{-24}	2.01×10^{-20}
	<i>std</i>	1.60×10^{-5}	7.62×10^{-5}	1.20×10^{-9}	1.53×10^{-5}	4.92×10^{-14}	1.17×10^{-14}	2.69×10^{-13}	1.27×10^{-8}	2.28×10^{-23}	7.77×10^{-20}
SMD5	<i>avg</i>	6.94×10^{-5}	6.94×10^{-5}	3.99×10^{-7}	3.48×10^{-6}	2.15×10^{-7}	4.62×10^{-7}	1.16×10^{-6}	2.59×10^{-4}	4.02×10^{-11}	8.58×10^{-8}
	<i>std</i>	1.05×10^{-4}	1.05×10^{-4}	4.68×10^{-7}	5.12×10^{-6}	3.69×10^{-7}	4.03×10^{-7}	1.48×10^{-6}	4.09×10^{-4}	8.99×10^{-11}	1.92×10^{-7}
SMD6	<i>avg</i>	1.18×10^{-1}	1.43×10^{-2}	5.13×10^{-2}	5.33×10^{-2}	8.50×10^{-7}	3.65×10^{-3}	3.50×10^{-2}	3.61×10^{-2}	2.45×10^{-15}	1.06×10^{-15}
	<i>std</i>	7.69×10^{-2}	1.84×10^{-2}	3.14×10^{-2}	8.48×10^{-2}	1.42×10^{-6}	3.83×10^{-3}	3.54×10^{-2}	5.12×10^{-2}	7.46×10^{-15}	2.32×10^{-15}
SMD7	<i>avg</i>	6.63×10^{-4}	6.63×10^{-2}	3.53×10^{-7}	2.05×10^{-5}	1.92×10^{-6}	7.17×10^{-6}	1.94×10^{-6}	5.39×10^{-5}	1.70×10^{-8}	1.29×10^{-2}
	<i>std</i>	3.09×10^{-3}	3.09×10^{-2}	1.81×10^{-7}	1.60×10^{-5}	3.30×10^{-6}	9.78×10^{-6}	2.08×10^{-6}	1.97×10^{-5}	3.75×10^{-8}	2.73×10^{-2}
SMD8	<i>avg</i>	2.60×10^{-5}	2.60×10^{-5}	1.76×10^{-7}	6.71×10^{-5}	1.59×10^{-5}	2.21×10^{-5}	3.30×10^{-8}	2.54×10^{-5}	1.92×10^{-15}	2.23×10^{-11}
	<i>std</i>	3.63×10^{-5}	3.63×10^{-5}	2.06×10^{-7}	6.92×10^{-5}	3.14×10^{-5}	2.83×10^{-5}	3.09×10^{-8}	2.89×10^{-5}	3.09×10^{-15}	4.91×10^{-11}
SMD9	<i>avg</i>	8.89×10^{-3}	8.97×10^{-3}	1.98×10^{-7}	1.66×10^{-4}	5.61×10^{-8}	5.74×10^{-6}	2.94×10^{-11}	9.67×10^{-9}	1.10×10^{-19}	5.43×10^{-18}
	<i>std</i>	1.09×10^{-2}	1.08×10^{-2}	6.29×10^{-9}	2.76×10^{-4}	4.06×10^{-8}	3.04×10^{-6}	4.99×10^{-11}	1.51×10^{-8}	1.40×10^{-19}	4.63×10^{-18}
SMD10	<i>avg</i>	1.39×10^0	1.93×10^0	2.36×10^{-4}	3.62×10^{-3}	9.99×10^{-6}	6.65×10^{-4}	1.38×10^{-5}	3.42×10^{-4}	3.21×10^{-6}	8.12×10^{-5}
	<i>std</i>	9.24×10^{-1}	1.29×10^0	3.75×10^{-4}	4.14×10^{-3}	6.31×10^{-6}	5.46×10^{-4}	2.12×10^{-5}	2.06×10^{-4}	4.37×10^{-6}	5.21×10^{-5}
SMD11	<i>avg</i>	5.91×10^{-1}	7.54×10^{-1}	9.42×10^{-3}	1.26×10^{-1}	8.44×10^{-3}	3.70×10^{-1}	3.57×10^{-2}	5.13×10^{-1}	1.67×10^{-3}	7.77×10^{-4}
	<i>std</i>	7.31×10^{-2}	9.57×10^{-2}	1.11×10^{-2}	8.31×10^{-1}	9.40×10^{-3}	2.56×10^{-1}	1.76×10^{-2}	1.45×10^{-1}	6.02×10^{-3}	2.80×10^{-3}
SMD12	<i>avg</i>	1.95×10^0	3.16×10^0	2.25×10^{-5}	4.98×10^{-3}	3.13×10^{-5}	7.58×10^{-3}	8.49×10^{-7}	9.41×10^{-5}	7.09×10^{-7}	1.15×10^{-4}
	<i>std</i>	2.42×10^{-1}	7.38×10^{-1}	2.37×10^{-5}	6.30×10^{-3}	4.76×10^{-5}	1.23×10^{-2}	5.14×10^{-7}	3.86×10^{-5}	8.85×10^{-7}	9.08×10^{-5}

Table 4. Ranking results of CFHTTAO and its related variants for SMD(2+3)

Algorithm	UL		LL	
	MR	R	MR	R
HTTAO	5.00	5	4.58	5
HTTAO-KS	3.50	4	3.42	4
HTTAO-CF	2.83	3	2.75	2
HTTAO-EL	2.67	2	2.83	3
CFHTTAO	1.00	1	1.42	1

In Table 4, the Friedman ranking results are further reported. A smaller MR value indicates better overall performance. It can be observed that CFHTTAO achieves the best ranking in both upper-level and lower-level optimization, with MR values of 1.00 and 1.42, respectively. Among the single-mechanism variants, HTTAO-EL obtains the best upper-level ranking, suggesting that the elite learning evolution mechanism contributes more prominently to upper-level optimization. In addition, HTTAO-CF achieves the best lower-level ranking among the single-mechanism variants, indicating that the coordinated feedback mechanism plays a more important role in improving lower-level optimization. Overall, the Friedman test results confirm that the three mechanisms are effective, and their combined use in CFHTTAO leads to the best overall performance.

5.3. Comparative Analysis Under Different Dimensions

The combinations of upper-level and lower-level dimensions in SMD problems are $(2 + 3)$, $(5 + 5)$, and $(10 + 10)$. Here, $SMD(p + q)$ is defined as the SMD problem with p upper-level dimensions and q lower-level dimensions. Thus, $SMD(2 + 3)$, $SMD(5 + 5)$, and $SMD(10 + 10)$ correspond to upper-level and lower-level dimension settings of $(2, 3)$, $(5, 5)$, and $(10, 10)$, respectively. As the problem dimension increases, the difficulty of solving the problem also increases accordingly [57]. To comprehensively evaluate algorithmic performance under different dimensional settings and assess the effectiveness of the algorithms in solving BOPs with increasing dimensionality, Tables 5–7 present the experimental results of CFHTTAO and other algorithms for solving SMD problems with all dimensional configurations described above.

The experimental results report the average (*avg*) and standard deviation (*std*) of the *UA* and *LA* obtained by each algorithm in solving the corresponding problem, which reflect the solution accuracy and stability of the algorithm at both levels of the bilevel problem. *UA* and *LA* represent the absolute deviations between the function values obtained by the algorithms and actual optimal function values at the upper and lower levels, respectively. Smaller average values of *UA* and *LA* indicate better solution accuracy at both levels, while smaller standard deviations of *UA* and *LA* reflect stronger algorithmic stability.

5.3.1. Comparative Experiments for SMD(2 + 3) Problems

The SMD problems with the dimensional combination of $(2 + 3)$ are first considered, where the upper-level and lower-level dimensions are set to 2 and 3, respectively. The corresponding comparative results of CFHTTAO and the other algorithms are presented in Table 5, including the average and standard deviation values of *UA* and *LA*.

As shown in Table 5, CFHTTAO achieves highly competitive solution accuracy and stability on the SMD test problems with the dimensional combination of (2 + 3). Taking SMD1 as an example, the average UA value obtained by CFHTTAO is 7.79×10^{-22} , whereas the corresponding values obtained by HHPSODE, HMDE, BIFOA, SABILDE, and BiWOA are 1.52×10^{-1} , 3.37×10^{-3} , 8.65×10^{-1} , 5.70×10^{-7} , and 1.45×10^{-13} , respectively. Similarly, the standard deviation of UA obtained by CFHTTAO on SMD1 is 1.04×10^{-21} , which is also lower than those of the other algorithms. These results indicate that, for SMD problems with the dimensional combination of (2 + 3), CFHTTAO produces the smallest absolute deviations, while also exhibiting the strongest stability.

5.3.2. Comparative Experiments for SMD(5 + 5) Problems

Furthermore, to investigate the performance of the algorithms on more challenging problems, both the upper-level and lower-level dimensions of the SMD problems are increased to 5, and the corresponding results for the dimensional combination of (5 + 5) are presented in Table 6.

As shown in Table 6, when the dimensional combination increases to (5 + 5), CFHTTAO still maintains strong overall superiority on most SMD problems. For SMD1–SMD6 and SMD9, CFHTTAO still obtains high-precision solutions as the search space expands. Although some algorithms obtain smaller UA values, their LA values are often much larger. For example, on SMD5, CFHTTAO achieves average UA and LA values of 1.13×10^{-8} and 2.07×10^{-6} , respectively, whereas BIFOA obtains corresponding values of 1.90×10^{-7} and 3.31×10^1 , respectively. Overall, CFHTTAO shows better comprehensive performance in terms of both upper-level and lower-level accuracy.

5.3.3. Comparative Experiments for SMD(10 + 10) Problems

The upper-level and lower-level dimensions of SMD problems are both increased to 10, and the solution capability of the algorithms is examined under the highest-dimensional setting of the SMD problems. The corresponding experimental results under the dimensional combination of (10 + 10) are shown in Table 7.

As can be seen from Table 7, when the dimensional combination is further increased to (10 + 10), all the algorithms face a more challenging optimization task due to the enlarged search space and the stronger coupling between upper-level and lower-level variables. Under this setting, CFHTTAO still maintains the best or highly competitive performance on most SMD problems. For example, on SMD1, CFHTTAO obtains average UA and LA values of 8.46×10^{-19} and 6.53×10^{-19} , respectively, which are smaller than those of HHPSODE, HMDE, BIFOA, SABILDE, and BiWOA. In conclusion, Tables 5–7 present the experimental results of the algorithms on the SMD test problems with upper-level dimensions of 2, 5, and 10 and corresponding lower-level dimensions of 3, 5, and 10, respectively. The results demonstrate that CFHTTAO consistently exhibits superior performance in solving complex BOPs under different dimensional settings. Moreover, as the dimensionality of the problems increases and the optimization tasks become more challenging, CFHTTAO maintains competitive performance, which further confirms its stronger applicability and robustness in handling complex BOPs.

Table 5. Comparison results of CFHTTAO and other algorithms for SMD(2 + 3) problems.

Problem		CFHTTAO		HHPSODE		HMDE		BIFOA		SABILDE		BiWOA	
		UA	LA	UA	LA	UA	LA	UA	LA	UA	LA	UA	LA
SMD1	avg	7.79×10^{-22}	3.95×10^{-22}	1.52×10^{-1}	1.21×10^{-1}	3.37×10^{-3}	2.51×10^{-3}	8.65×10^{-1}	7.45×10^{-1}	5.70×10^{-7}	3.74×10^{-7}	1.45×10^{-13}	7.16×10^{-14}
	std	1.04×10^{-21}	4.81×10^{-22}	1.15×10^{-1}	7.49×10^{-2}	1.49×10^{-3}	1.28×10^{-3}	7.67×10^{-13}	1.86×10^{-7}	3.00×10^{-7}	1.91×10^{-7}	2.99×10^{-13}	1.44×10^{-13}
SMD2	avg	1.29×10^{-9}	1.64×10^{-7}	2.02×10^{-1}	5.04×10^{-1}	1.24×10^{-4}	1.73×10^{-1}	5.70×10^{-8}	1.30×10^1	2.31×10^{-7}	9.78×10^{-5}	5.14×10^{-4}	5.43×10^{-3}
	std	1.89×10^{-9}	1.99×10^{-7}	1.53×10^{-1}	2.73×10^{-1}	5.10×10^{-5}	1.39×10^{-1}	5.57×10^{-8}	9.28×10^0	2.27×10^{-7}	1.19×10^{-4}	3.94×10^{-4}	4.29×10^{-3}
SMD3	avg	4.44×10^{-29}	8.76×10^{-31}	1.35×10^{-1}	2.93×10^{-1}	4.55×10^{-2}	3.43×10^{-1}	1.07×10^0	2.86×10^0	2.10×10^{-6}	3.09×10^{-6}	3.14×10^{-15}	1.60×10^{-15}
	std	4.25×10^{-29}	1.79×10^{-30}	1.14×10^{-1}	7.70×10^{-2}	2.88×10^{-2}	3.10×10^{-1}	7.57×10^{-13}	8.10×10^{-7}	1.32×10^{-6}	2.35×10^{-6}	2.51×10^{-15}	1.28×10^{-15}
SMD4	avg	5.88×10^{-24}	2.01×10^{-20}	5.95×10^{-4}	6.94×10^{-1}	5.24×10^{-5}	2.30×10^{-1}	3.23×10^{-8}	3.88×10^0	6.22×10^{-9}	3.78×10^{-2}	1.49×10^{-15}	4.12×10^{-3}
	std	2.28×10^{-23}	7.77×10^{-20}	3.33×10^{-4}	2.87×10^{-1}	6.17×10^{-5}	3.38×10^{-1}	4.42×10^{-8}	1.47×10^0	8.56×10^{-9}	8.36×10^{-2}	3.12×10^{-15}	5.02×10^{-3}
SMD5	avg	4.02×10^{-11}	8.58×10^{-8}	4.96×10^{-3}	5.47×10^{-1}	5.72×10^{-2}	1.06×10^{-1}	1.22×10^{-7}	8.92×10^0	1.35×10^{-8}	6.85×10^{-6}	2.60×10^{-4}	9.54×10^{-3}
	std	8.99×10^{-11}	1.92×10^{-7}	2.16×10^{-3}	2.97×10^{-1}	1.40×10^{-1}	1.53×10^{-1}	7.00×10^{-8}	1.63×10^0	1.68×10^{-8}	5.03×10^{-6}	2.73×10^{-4}	1.01×10^{-2}
SMD6	avg	2.45×10^{-15}	1.06×10^{-15}	9.37×10^{-1}	4.30×10^{-1}	3.90×10^{-1}	1.37×10^{-1}	1.43×10^0	5.56×10^0	1.41×10^{-1}	3.38×10^{-2}	1.50×10^{-1}	7.77×10^{-2}
	std	7.46×10^{-15}	2.32×10^{-15}	5.35×10^{-1}	6.20×10^{-1}	4.42×10^{-1}	2.46×10^{-1}	4.92×10^{-5}	4.90×10^{-5}	1.91×10^{-1}	5.00×10^{-2}	1.84×10^{-1}	1.04×10^{-1}
SMD7	avg	1.70×10^{-8}	1.29×10^{-2}	1.36×10^{-2}	7.84×10^{-2}	4.09×10^{-3}	8.04×10^1	3.37×10^{-8}	2.75×10^1	5.21×10^{-4}	5.00×10^1	1.07×10^{-2}	5.03×10^1
	std	3.75×10^{-8}	2.73×10^{-2}	2.69×10^{-2}	5.31×10^{-2}	7.34×10^{-3}	1.11×10^2	1.48×10^{-8}	3.36×10^1	1.16×10^{-3}	1.12×10^2	2.26×10^{-2}	1.13×10^2
SMD8	avg	1.92×10^{-15}	2.23×10^{-11}	7.63×10^{-2}	4.21×10^{-1}	1.25×10^{-4}	4.35×10^{-1}	1.40×10^{-3}	9.89×10^0	1.89×10^{-7}	6.27×10^{-4}	5.17×10^{-5}	1.20×10^{-2}
	std	3.09×10^{-15}	4.91×10^{-11}	1.41×10^{-1}	3.12×10^{-1}	7.76×10^{-5}	3.04×10^{-1}	3.13×10^{-3}	5.07×10^0	3.79×10^{-7}	1.08×10^{-3}	4.29×10^{-5}	5.06×10^{-3}
SMD9	avg	1.10×10^{-19}	5.43×10^{-18}	2.46×10^{-3}	2.28×10^0	1.13×10^{-7}	7.72×10^{-5}	1.57×10^{-7}	2.16×10^1	4.15×10^{-8}	1.59×10^{-4}	3.22×10^{-4}	7.34×10^{-4}
	std	1.40×10^{-19}	4.63×10^{-18}	2.37×10^{-3}	2.44×10^0	8.62×10^{-8}	1.24×10^{-4}	2.35×10^{-7}	1.35×10^1	6.48×10^{-8}	1.56×10^{-4}	7.20×10^{-4}	1.60×10^{-3}
SMD10	avg	3.21×10^{-6}	8.12×10^{-5}	9.53×10^0	4.38×10^1	2.03×10^{-3}	2.19×10^0	1.20×10^1	3.05×10^1	1.17×10^1	1.55×10^1	1.19×10^{-2}	6.04×10^{-1}
	std	4.37×10^{-6}	5.21×10^{-5}	4.42×10^0	3.08×10^1	1.24×10^{-3}	1.07×10^0	9.77×10^0	1.47×10^0	8.77×10^0	2.07×10^1	1.06×10^{-2}	3.98×10^{-1}
SMD11	avg	1.67×10^{-3}	7.77×10^{-4}	4.42×10^{-3}	1.55×10^0	6.70×10^{-4}	1.22×10^0	2.03×10^0	5.24×10^1	1.00×10^0	1.25×10^0	8.68×10^{-2}	1.56×10^{-1}
	std	6.02×10^{-3}	2.80×10^{-3}	3.23×10^{-3}	1.13×10^0	1.28×10^{-3}	4.46×10^{-1}	0.00×10^0	5.48×10^{-5}	1.13×10^{-3}	3.85×10^{-1}	7.87×10^{-2}	7.77×10^{-2}
SMD12	avg	7.09×10^{-7}	1.15×10^{-4}	9.85×10^{-1}	5.79×10^0	3.24×10^{-3}	2.61×10^0	7.02×10^0	2.89×10^1	3.00×10^0	1.03×10^1	7.64×10^{-1}	3.42×10^0
	std	8.85×10^{-7}	9.08×10^{-5}	1.00×10^0	7.59×10^0	3.16×10^{-3}	1.05×10^0	7.66×10^{-1}	7.22×10^{-1}	1.77×10^{-3}	8.69×10^0	1.64×10^0	1.76×10^0

Table 6. Comparison results of CFHTTAO and other algorithms for SMD(5 + 5) problems.

Problem		CFHTTAO		HHPSODE		HMDE		BIFOA		SABILDE		BiWOA	
		UA	LA	UA	LA	UA	LA	UA	LA	UA	LA	UA	LA
SMD1	avg	2.93×10^{-20}	2.34×10^{-20}	4.80×10^0	3.34×10^0	5.66×10^{-1}	4.46×10^{-1}	3.23×10^0	3.22×10^0	3.00×10^{-5}	1.60×10^{-5}	5.55×10^{-6}	2.45×10^{-6}
	std	3.56×10^{-20}	3.49×10^{-20}	2.26×10^0	1.84×10^0	2.33×10^{-1}	1.69×10^{-1}	2.24×10^{-11}	7.22×10^{-8}	1.48×10^{-5}	1.28×10^{-5}	1.24×10^{-5}	5.47×10^{-6}
SMD2	avg	7.83×10^{-7}	1.29×10^{-5}	3.10×10^0	8.33×10^0	8.68×10^{-4}	2.42×10^0	9.45×10^{-8}	2.17×10^1	3.32×10^{-4}	3.08×10^{-1}	5.27×10^{-3}	5.74×10^{-1}
	std	1.28×10^{-6}	1.43×10^{-5}	3.34×10^0	7.98×10^0	6.53×10^{-4}	8.59×10^{-1}	8.84×10^{-8}	1.38×10^1	4.38×10^{-4}	2.40×10^{-1}	2.10×10^{-3}	4.41×10^{-1}
SMD3	avg	6.34×10^{-20}	1.03×10^{-22}	6.17×10^0	5.64×10^0	5.70×10^0	5.91×10^0	3.24×10^0	6.48×10^0	5.33×10^{-4}	9.17×10^{-4}	4.45×10^{-10}	3.42×10^{-10}
	std	8.23×10^{-19}	1.60×10^{-21}	4.93×10^0	4.04×10^0	2.39×10^0	9.87×10^{-1}	7.20×10^{-12}	7.99×10^{-12}	4.88×10^{-4}	9.54×10^{-4}	9.36×10^{-10}	7.17×10^{-10}
SMD4	avg	2.97×10^{-21}	1.93×10^{-21}	5.18×10^{-2}	4.90×10^0	7.78×10^{-4}	3.89×10^0	1.17×10^{-7}	9.72×10^0	4.97×10^{-4}	2.19×10^0	9.77×10^{-2}	2.07×10^0
	std	3.67×10^{-21}	1.78×10^{-21}	5.72×10^{-2}	1.30×10^0	1.08×10^{-3}	1.40×10^0	7.85×10^{-8}	9.81×10^{-1}	5.31×10^{-4}	1.04×10^0	8.31×10^{-2}	2.09×10^0
SMD5	avg	1.13×10^{-8}	2.07×10^{-6}	1.87×10^{-1}	1.36×10^1	4.99×10^{-3}	1.86×10^1	1.90×10^{-7}	3.31×10^1	3.53×10^{-3}	2.38×10^0	1.46×10^{-1}	4.79×10^0
	std	2.16×10^{-8}	3.41×10^{-6}	1.04×10^{-1}	4.10×10^0	4.13×10^{-3}	8.21×10^0	1.78×10^{-7}	1.20×10^1	3.78×10^{-3}	2.23×10^0	2.32×10^{-1}	3.21×10^0
SMD6	avg	2.77×10^{-15}	3.91×10^{-15}	3.66×10^0	4.77×10^0	6.04×10^{-1}	4.44×10^0	1.27×10^{-7}	5.03×10^1	3.93×10^{-2}	5.22×10^{-1}	4.76×10^{-1}	2.95×10^{-1}
	std	3.65×10^{-15}	5.22×10^{-15}	2.27×10^0	1.54×10^0	8.38×10^{-1}	2.56×10^0	9.56×10^{-8}	1.53×10^1	6.36×10^{-2}	4.29×10^{-1}	7.40×10^{-1}	3.88×10^{-1}
SMD7	avg	4.24×10^{-5}	3.33×10^{-1}	1.28×10^{-3}	1.14×10^2	3.40×10^{-3}	2.98×10^2	7.85×10^0	1.69×10^3	4.40×10^{-3}	3.73×10^2	5.07×10^{-3}	9.41×10^2
	std	4.61×10^{-5}	3.24×10^{-1}	1.29×10^{-3}	1.93×10^2	2.89×10^{-3}	1.82×10^2	3.60×10^0	7.61×10^2	4.25×10^{-3}	3.64×10^2	3.49×10^{-3}	8.85×10^2
SMD8	avg	1.31×10^{-6}	1.36×10^{-4}	5.76×10^{-3}	3.07×10^1	4.77×10^{-3}	4.30×10^1	7.94×10^{-7}	1.01×10^2	6.96×10^{-2}	3.27×10^1	1.60×10^{-1}	4.50×10^0
	std	2.34×10^{-6}	2.11×10^{-4}	1.78×10^{-3}	1.37×10^1	4.19×10^{-3}	9.85×10^0	5.22×10^{-7}	3.25×10^1	9.28×10^{-2}	1.56×10^1	1.99×10^{-1}	4.64×10^0
SMD9	avg	8.87×10^{-18}	8.82×10^{-17}	3.86×10^{-2}	2.28×10^1	2.97×10^{-3}	4.68×10^0	2.60×10^{-7}	3.22×10^1	6.25×10^{-2}	2.16×10^0	1.49×10^{-1}	2.05×10^{-1}
	std	1.08×10^{-17}	8.41×10^{-17}	2.42×10^{-2}	1.33×10^1	1.87×10^{-3}	2.40×10^0	1.69×10^{-7}	1.74×10^1	1.40×10^{-1}	4.09×10^0	3.34×10^{-1}	4.57×10^{-1}
SMD10	avg	2.85×10^1	2.93×10^1	3.96×10^1	7.82×10^1	3.31×10^1	5.75×10^1	2.03×10^1	1.40×10^2	2.21×10^1	3.71×10^1	1.40×10^1	1.20×10^1
	std	1.88×10^0	1.67×10^0	9.31×10^0	2.07×10^1	9.43×10^0	1.02×10^1	2.18×10^0	2.73×10^1	1.00×10^1	3.52×10^1	5.70×10^{-1}	4.95×10^{-1}
SMD11	avg	7.77×10^{-3}	3.10×10^{-1}	1.93×10^{-1}	1.46×10^1	6.48×10^0	1.16×10^1	1.67×10^0	2.10×10^1	4.99×10^0	9.07×10^0	5.25×10^0	5.31×10^0
	std	1.67×10^{-2}	2.93×10^{-1}	1.55×10^{-1}	2.12×10^1	8.29×10^0	8.59×10^0	8.38×10^{-1}	2.16×10^0	3.69×10^0	4.02×10^0	2.12×10^0	2.19×10^0
SMD12	avg	3.66×10^1	2.97×10^1	5.35×10^1	4.37×10^1	4.62×10^1	3.59×10^1	4.97×10^1	3.95×10^1	2.80×10^1	1.78×10^1	4.28×10^1	3.06×10^1
	std	1.21×10^0	1.16×10^0	6.40×10^0	9.42×10^0	3.00×10^0	8.08×10^0	6.17×10^{-1}	8.38×10^{-1}	7.42×10^{-1}	7.55×10^0	4.17×10^0	3.67×10^0

Table 7. Comparison results of CFHTTAO and other algorithms for SMD(10 + 10) problems.

Problem		CFHTTAO		HHPSODE		HMDE		BIFOA		SABILDE		BiWOA	
		UA	LA	UA	LA	UA	LA	UA	LA	UA	LA	UA	LA
SMD1	avg	8.46×10^{-19}	6.53×10^{-19}	4.46×10^1	3.45×10^1	5.79×10^1	4.54×10^1	1.80×10^1	1.73×10^1	4.23×10^{-1}	2.68×10^{-1}	2.60×10^{-8}	1.51×10^{-8}
	std	1.24×10^{-18}	1.16×10^{-18}	8.07×10^0	7.30×10^0	1.47×10^1	1.75×10^1	0.00×10^0	0.00×10^0	2.57×10^{-1}	2.27×10^{-1}	3.61×10^{-8}	2.64×10^{-8}
SMD2	avg	3.05×10^{-3}	3.63×10^{-2}	5.06×10^{-1}	5.49×10^1	9.84×10^{-3}	4.55×10^1	4.67×10^{-7}	7.73×10^1	1.69×10^{-2}	9.02×10^0	1.10×10^{-1}	1.45×10^0
	std	1.23×10^{-3}	2.42×10^{-2}	4.77×10^{-1}	3.01×10^1	9.78×10^{-3}	2.27×10^1	3.19×10^{-7}	2.57×10^1	2.00×10^{-2}	7.64×10^0	2.02×10^{-1}	7.39×10^{-1}
SMD3	avg	2.15×10^{-17}	3.19×10^{-17}	4.01×10^1	3.88×10^1	5.02×10^1	4.70×10^1	1.93×10^1	2.58×10^1	1.92×10^0	1.39×10^0	2.59×10^{-9}	1.29×10^{-9}
	std	2.66×10^{-16}	6.45×10^{-16}	9.10×10^0	5.94×10^0	3.57×10^0	1.44×10^0	0.00×10^0	0.00×10^0	1.94×10^0	1.85×10^0	3.38×10^{-9}	2.08×10^{-9}
SMD4	avg	4.33×10^{-18}	2.67×10^{-15}	8.77×10^{-2}	1.79×10^1	6.19×10^{-3}	1.49×10^1	2.41×10^{-7}	3.74×10^1	1.26×10^{-3}	4.14×10^0	1.78×10^{-2}	3.72×10^0
	std	8.43×10^{-18}	4.53×10^{-15}	7.24×10^{-2}	4.68×10^0	7.05×10^{-3}	4.52×10^0	6.10×10^{-8}	1.75×10^0	1.13×10^{-3}	8.52×10^{-1}	2.69×10^{-2}	1.01×10^0
SMD5	avg	7.47×10^{-4}	3.29×10^{-1}	4.32×10^{-2}	2.32×10^2	1.31×10^{-1}	3.02×10^2	1.73×10^{-6}	7.44×10^2	7.99×10^{-2}	6.56×10^1	3.09×10^{-1}	4.29×10^0
	std	4.56×10^{-4}	3.29×10^{-1}	5.77×10^{-2}	5.75×10^1	1.53×10^{-1}	1.87×10^2	2.02×10^{-6}	5.09×10^1	1.16×10^{-1}	4.83×10^1	3.64×10^{-1}	1.79×10^0
SMD6	avg	1.30×10^{-15}	2.13×10^{-14}	1.05×10^1	4.37×10^1	1.44×10^1	5.07×10^1	3.37×10^{-7}	1.15×10^2	3.73×10^0	3.03×10^1	3.11×10^{-2}	1.08×10^{-1}
	std	1.98×10^{-15}	2.27×10^{-14}	5.43×10^0	1.54×10^1	1.23×10^1	1.58×10^1	1.13×10^{-7}	1.28×10^1	5.07×10^0	2.39×10^1	5.77×10^{-2}	2.02×10^{-1}
SMD7	avg	3.10×10^{-3}	9.03×10^{-1}	4.52×10^{-3}	9.17×10^2	6.96×10^{-3}	8.79×10^2	9.84×10^0	1.55×10^3	4.84×10^{-3}	6.18×10^2	7.64×10^{-2}	1.01×10^3
	std	5.08×10^{-3}	1.49×10^0	3.64×10^{-3}	7.25×10^2	6.44×10^{-3}	9.61×10^2	6.78×10^0	1.21×10^3	3.14×10^{-3}	9.78×10^2	1.06×10^{-1}	9.84×10^2
SMD8	avg	1.67×10^{-2}	7.99×10^{-1}	8.76×10^{-1}	1.62×10^2	2.41×10^0	2.85×10^2	5.53×10^{-1}	4.37×10^1	1.51×10^{-2}	8.50×10^1	6.35×10^{-1}	2.93×10^1
	std	1.53×10^{-2}	7.13×10^{-1}	6.26×10^{-1}	8.81×10^1	2.94×10^0	2.37×10^1	4.95×10^{-1}	6.92×10^0	1.50×10^{-2}	7.03×10^1	1.15×10^0	2.37×10^1
SMD9	avg	2.32×10^{-13}	8.60×10^{-12}	3.58×10^{-2}	6.26×10^1	7.04×10^{-3}	7.97×10^1	4.85×10^{-7}	9.84×10^1	1.25×10^{-1}	9.78×10^0	4.32×10^{-1}	2.38×10^0
	std	2.96×10^{-13}	1.50×10^{-11}	3.41×10^{-2}	2.82×10^1	9.02×10^{-3}	2.15×10^1	2.28×10^{-7}	1.12×10^1	1.70×10^{-1}	1.23×10^1	9.62×10^{-1}	5.31×10^0
SMD10	avg	3.93×10^1	3.62×10^1	9.00×10^0	4.15×10^2	6.62×10^1	1.61×10^2	8.41×10^1	9.28×10^1	4.87×10^1	4.90×10^1	3.05×10^1	1.82×10^1
	std	2.13×10^0	3.54×10^0	7.92×10^0	9.87×10^1	2.25×10^1	1.42×10^1	2.18×10^1	2.06×10^1	5.28×10^0	8.31×10^0	2.29×10^{-1}	8.06×10^{-1}
SMD11	avg	1.18×10^0	1.08×10^0	2.18×10^0	4.49×10^1	7.88×10^1	9.92×10^1	7.29×10^1	7.02×10^1	5.11×10^1	6.29×10^1	1.13×10^1	1.11×10^1
	std	1.06×10^0	4.27×10^{-1}	1.57×10^0	6.69×10^0	3.29×10^1	3.43×10^1	4.18×10^1	4.12×10^1	2.11×10^1	2.22×10^1	1.98×10^0	2.19×10^0
SMD12	avg	7.35×10^1	5.50×10^1	1.54×10^2	1.17×10^2	1.08×10^2	9.25×10^1	9.02×10^1	6.54×10^1	5.85×10^1	5.00×10^1	9.21×10^1	4.40×10^1
	std	3.10×10^0	7.65×10^0	3.69×10^1	2.44×10^1	1.95×10^1	1.69×10^1	2.39×10^0	1.44×10^0	1.28×10^1	1.12×10^1	7.72×10^0	5.10×10^0

5.4. Analysis of Computational Results

Specific computational results are essential for validating the effectiveness of the algorithm, because they provide an intuitive demonstration of the obtained solutions and computational performance under different dimensional settings and test problems. The experimental results in the preceding subsection demonstrate that CFHTTAO exhibits significant advantages in solving BOPs across varying dimensions. To further assess the effectiveness of the proposed algorithm in addressing BOPs, Tables 8–10 report the specific computational results obtained by CFHTTAO for the SMD problems with different dimensional combinations. In Tables 8–10, x_u^* and x_l^* denote the upper-level and lower-level optimal solutions, respectively. UA^* and LA^* , respectively, represent the upper-level and lower-level absolute deviations between the function values of the obtained optimal solutions and the actual optimal function values, which reflect the upper-level and lower-level accuracy of the obtained optimal solutions.

Table 8. The optimal computation results of SMD(2 + 3) problems obtained by CFHTTAO.

Problem	x_u^*	x_l^*	UA^*	LA^*
SMD1	$(-3.01 \times 10^{-14}, 3.05 \times 10^{-14})$	$(-1.91 \times 10^{-14}, -2.16 \times 10^{-14}, 1.92 \times 10^{-13})$	2.86×10^{-26}	2.77×10^{-26}
SMD2	$(4.88 \times 10^{-11}, -2.90 \times 10^{-5})$	$(5.96 \times 10^{-8}, 6.16 \times 10^{-8}, 1.00 \times 10^0)$	2.06×10^{-13}	8.39×10^{-10}
SMD3	$(7.27 \times 10^{-17}, 7.49 \times 10^{-17})$	$(5.24 \times 10^{-17}, 3.61 \times 10^{-17}, 2.50 \times 10^{-17})$	1.56×10^{-32}	6.23×10^{-34}
SMD4	$(-3.02 \times 10^{-15}, -2.59 \times 10^{-16})$	$(-2.08 \times 10^{-15}, -2.17 \times 10^{-15}, 0.00 \times 10^0)$	6.13×10^{-32}	6.72×10^{-32}
SMD5	$(1.39 \times 10^{-9}, 4.90 \times 10^{-8})$	$(1.00 \times 10^0, 1.00 \times 10^0, -2.44 \times 10^{-4})$	8.96×10^{-19}	2.41×10^{-15}
SMD6	$(1.45 \times 10^{-8}, 1.09 \times 10^{-8})$	$(-1.46 \times 10^{-8}, -1.46 \times 10^{-8}, -1.66 \times 10^{-8})$	8.25×10^{-20}	9.67×10^{-16}
SMD7	$(-1.30 \times 10^{-4}, -1.33 \times 10^{-4})$	$(1.58 \times 10^{-6}, 1.61 \times 10^{-6}, 1.00 \times 10^0)$	1.35×10^{-12}	2.62×10^{-8}
SMD8	$(7.45 \times 10^{-14}, 3.18 \times 10^{-15})$	$(1.00 \times 10^0, 1.00 \times 10^0, -2.20 \times 10^{-3})$	2.91×10^{-17}	3.72×10^{-13}
SMD9	$(1.95 \times 10^{-10}, 5.51 \times 10^{-11})$	$(7.99 \times 10^{-11}, 5.41 \times 10^{-11}, -1.24 \times 10^{-10})$	2.29×10^{-22}	7.92×10^{-20}
SMD10	$(1.00 \times 10^0, 1.00 \times 10^0)$	$(1.00 \times 10^0, 1.00 \times 10^0, 7.85 \times 10^{-1})$	5.28×10^{-8}	9.17×10^{-5}
SMD11	$(5.49 \times 10^{-7}, 1.53 \times 10^{-13})$	$(-3.64 \times 10^{-9}, -3.58 \times 10^{-9}, 3.68 \times 10^{-1})$	3.77×10^{-15}	6.07×10^{-13}
SMD12	$(1.00 \times 10^0, 1.00 \times 10^0)$	$(1.00 \times 10^0, 1.00 \times 10^0, -5.76 \times 10^{-5})$	9.38×10^{-8}	1.02×10^{-4}

Table 9. The optimal computation results of SMD(5 + 5) problems obtained by CFHTTAO.

Problem	x_u^*	x_l^*	UA^*	LA^*
SMD1	$(2.86 \times 10^{-13}, 2.86 \times 10^{-13}, 2.86 \times 10^{-13}, -1.82 \times 10^{-13}, -1.82 \times 10^{-13})$	$(-1.93 \times 10^{-13}, -7.28 \times 10^{-14}, 6.25 \times 10^{-15}, -1.40 \times 10^{-13}, 7.04 \times 10^{-14})$	4.20×10^{-25}	3.54×10^{-25}
SMD2	$(-6.10 \times 10^{-7}, -6.10 \times 10^{-7}, -6.10 \times 10^{-7}, 1.90 \times 10^{-5}, 1.90 \times 10^{-5})$	$(-2.17 \times 10^{-10}, -2.17 \times 10^{-10}, -2.17 \times 10^{-10}, 1.00 \times 10^0, 1.00 \times 10^0)$	4.06×10^{-10}	1.13×10^{-9}
SMD3	$(1.81 \times 10^{-17}, 1.81 \times 10^{-17}, 1.81 \times 10^{-17}, -1.24 \times 10^{-17}, -1.24 \times 10^{-17})$	$(1.60 \times 10^{-15}, 1.61 \times 10^{-15}, 1.57 \times 10^{-15}, 8.89 \times 10^{-16}, 5.40 \times 10^{-16})$	8.68×10^{-30}	1.08×10^{-30}
SMD4	$(-1.03 \times 10^{-11}, -4.95 \times 10^{-12}, -7.42 \times 10^{-12}, -2.82 \times 10^{-12}, -3.72 \times 10^{-12})$	$(-7.80 \times 10^{-12}, -7.87 \times 10^{-12}, -7.88 \times 10^{-12}, 0.00 \times 10^0, 0.00 \times 10^0)$	9.56×10^{-25}	2.18×10^{-23}
SMD5	$(2.57 \times 10^{-5}, 3.54 \times 10^{-5}, 1.49 \times 10^{-5}, 3.47 \times 10^{-5}, 3.42 \times 10^{-5})$	$(1.00 \times 10^0, 1.00 \times 10^0, 1.00 \times 10^0, -5.74 \times 10^{-3}, -5.73 \times 10^{-3})$	2.81×10^{-11}	6.67×10^{-9}
SMD6	$(5.98 \times 10^{-11}, -8.55 \times 10^{-11}, 1.27 \times 10^{-10}, 7.02 \times 10^{-11}, -1.76 \times 10^{-10})$	$(4.77 \times 10^{-11}, -1.18 \times 10^{-10}, -1.18 \times 10^{-10}, -2.02 \times 10^{-10}, -7.04 \times 10^{-11})$	3.55×10^{-21}	1.15×10^{-19}
SMD7	$(-3.43 \times 10^{-1}, -3.08 \times 10^{-2}, -1.45 \times 10^{-1}, -1.49 \times 10^{-1}, -1.49 \times 10^{-1})$	$(4.98 \times 10^{-4}, 5.18 \times 10^{-4}, 3.71 \times 10^{-4}, 6.48 \times 10^{-1}, 1.01 \times 10^0)$	2.11×10^{-8}	6.31×10^{-2}
SMD8	$(2.19 \times 10^{-10}, 2.19 \times 10^{-10}, 2.19 \times 10^{-10}, -2.73 \times 10^{-7}, -2.73 \times 10^{-7})$	$(1.00 \times 10^0, 1.00 \times 10^0, 1.00 \times 10^0, -6.43 \times 10^{-3}, -6.43 \times 10^{-3})$	1.94×10^{-11}	1.55×10^{-9}
SMD9	$(-1.86 \times 10^{-10}, -3.66 \times 10^{-10}, -1.07 \times 10^{-10}, -2.66 \times 10^{-10}, -1.94 \times 10^{-10})$	$(-3.89 \times 10^{-11}, 6.90 \times 10^{-11}, 9.60 \times 10^{-11}, 2.36 \times 10^{-10}, -1.56 \times 10^{-10})$	1.92×10^{-20}	4.49×10^{-19}
SMD10	$(6.68 \times 10^{-1}, 5.92 \times 10^{-1}, -1.25 \times 10^0, 5.20 \times 10^{-1}, -1.25 \times 10^0)$	$(4.53 \times 10^{-1}, 1.71 \times 10^{-1}, 1.89 \times 10^{-1}, 1.30 \times 10^0, 2.74 \times 10^{-1})$	1.12×10^1	2.05×10^1
SMD11	$(-1.02 \times 10^{-1}, -1.10 \times 10^{-1}, -1.09 \times 10^{-1}, -2.42 \times 10^{-1}, -2.40 \times 10^{-1})$	$(6.48 \times 10^{-4}, 4.36 \times 10^{-5}, 1.81 \times 10^{-4}, 3.68 \times 10^{-1}, 3.68 \times 10^{-1})$	1.13×10^{-5}	1.86×10^{-1}
SMD12	$(-1.58 \times 10^0, -1.58 \times 10^0, 7.07 \times 10^{-1}, 9.99 \times 10^{-1}, 9.98 \times 10^{-1})$	$(2.57 \times 10^{-1}, 3.41 \times 10^{-2}, 2.06 \times 10^{-1}, -7.85 \times 10^{-1}, -7.85 \times 10^{-1})$	3.56×10^1	3.04×10^1

Table 10. The optimal computation results of SMD(10 + 10) problems obtained by CFHTTAO.

Problem	x_u^*	x_l^*	UA^*	LA^*
SMD1	$(-6.16 \times 10^{-14}, 2.00 \times 10^{-13}, 1.70 \times 10^{-14},$ $1.69 \times 10^{-14}, -5.95 \times 10^{-15}, 1.40 \times 10^{-11},$ $3.74 \times 10^{-11}, 1.52 \times 10^{-11}, 1.87 \times 10^{-11},$ $1.84 \times 10^{-11})$	$(3.05 \times 10^{-11}, 3.09 \times 10^{-11}, 3.08 \times 10^{-11},$ $3.16 \times 10^{-11}, 3.09 \times 10^{-11}, 3.21 \times 10^{-11},$ $3.25 \times 10^{-11}, 3.22 \times 10^{-11}, 3.22 \times 10^{-11},$ $3.23 \times 10^{-11})$	8.32×10^{-21}	5.81×10^{-21}
SMD2	$(-4.16 \times 10^{-3}, -3.25 \times 10^{-3}, -4.04 \times 10^{-4},$ $-1.51 \times 10^{-2}, 1.06 \times 10^{-2}, -3.44 \times 10^{-2},$ $-3.71 \times 10^{-2}, -4.08 \times 10^{-2}, 4.32 \times 10^{-2},$ $-2.31 \times 10^{-2})$	$(8.60 \times 10^{-4}, -2.72 \times 10^{-2}, 4.04 \times 10^{-4},$ $-3.20 \times 10^{-3}, -1.99 \times 10^{-3}, 9.33 \times 10^{-1},$ $9.29 \times 10^{-1}, 9.43 \times 10^{-1}, 1.10 \times 10^0,$ $1.00 \times 10^0)$	6.77×10^{-5}	7.43×10^{-3}
SMD3	$(-1.30 \times 10^{-14}, -1.30 \times 10^{-14}, -1.30 \times 10^{-14},$ $-1.30 \times 10^{-14}, -1.30 \times 10^{-14}, -1.01 \times 10^{-14},$ $-1.01 \times 10^{-14}, -1.01 \times 10^{-14}, -1.01 \times 10^{-14},$ $-1.01 \times 10^{-14})$	$(-9.02 \times 10^{-16}, -2.27 \times 10^{-16}, -4.87 \times 10^{-16},$ $-3.04 \times 10^{-15}, -5.44 \times 10^{-16}, -4.22 \times 10^{-15},$ $-5.53 \times 10^{-15}, -1.90 \times 10^{-15}, -8.17 \times 10^{-15},$ $-1.33 \times 10^{-14})$	1.66×10^{-27}	2.95×10^{-28}
SMD4	$(2.44 \times 10^{-9}, 3.35 \times 10^{-9}, 2.11 \times 10^{-9},$ $1.93 \times 10^{-9}, 2.23 \times 10^{-9}, 2.51 \times 10^{-9},$ $4.71 \times 10^{-10}, 3.14 \times 10^{-9}, 6.66 \times 10^{-10},$ $2.53 \times 10^{-9})$	$(-1.72 \times 10^{-9}, -1.99 \times 10^{-9}, -2.49 \times 10^{-9},$ $-2.30 \times 10^{-9}, -3.45 \times 10^{-9}, 0.00 \times 10^0,$ $0.00 \times 10^0, 0.00 \times 10^0, 0.00 \times 10^0,$ $0.00 \times 10^0)$	2.37×10^{-20}	9.11×10^{-16}
SMD5	$(1.70 \times 10^{-2}, 1.19 \times 10^{-3}, 1.65 \times 10^{-2},$ $3.06 \times 10^{-3}, -6.68 \times 10^{-2}, -2.98 \times 10^{-4},$ $1.75 \times 10^{-2}, 8.29 \times 10^{-2}, -4.89 \times 10^{-3},$ $-2.48 \times 10^{-2})$	$(9.89 \times 10^{-1}, 9.97 \times 10^{-1}, 1.02 \times 10^0,$ $1.08 \times 10^0, 1.20 \times 10^0, -1.69 \times 10^{-1},$ $-8.50 \times 10^{-2}, 2.79 \times 10^{-1}, -1.97 \times 10^{-1},$ $-2.07 \times 10^{-1})$	6.57×10^{-5}	1.79×10^{-2}
SMD6	$(-6.03 \times 10^{-8}, 7.32 \times 10^{-8}, -3.37 \times 10^{-8},$ $1.32 \times 10^{-7}, 6.80 \times 10^{-8}, 5.18 \times 10^{-8},$ $4.89 \times 10^{-8}, 5.44 \times 10^{-8}, -1.37 \times 10^{-8},$ $1.04 \times 10^{-8})$	$(1.59 \times 10^{-7}, 1.77 \times 10^{-7}, -9.95 \times 10^{-8},$ $-1.01 \times 10^{-7}, 1.09 \times 10^{-7}, 4.15 \times 10^{-8},$ $1.97 \times 10^{-8}, 1.45 \times 10^{-7}, -8.16 \times 10^{-8},$ $-3.81 \times 10^{-8})$	1.73×10^{-17}	1.05×10^{-13}
SMD7	$(-3.66 \times 10^{-2}, -3.66 \times 10^{-2}, -3.65 \times 10^{-2},$ $-3.65 \times 10^{-2}, -3.66 \times 10^{-2}, -3.71 \times 10^{-1},$ $-3.98 \times 10^{-1}, -2.93 \times 10^{-1}, -3.92 \times 10^{-1},$ $-4.69 \times 10^{-1})$	$(-7.27 \times 10^{-5}, -4.61 \times 10^{-5}, 2.71 \times 10^{-5},$ $4.33 \times 10^{-5}, -3.15 \times 10^{-5}, 4.72 \times 10^{-1},$ $4.14 \times 10^{-1}, 6.95 \times 10^{-1}, 3.70 \times 10^{-1},$ $5.61 \times 10^{-1})$	4.66×10^{-8}	7.58×10^{-1}
SMD8	$(-3.80 \times 10^{-2}, -3.40 \times 10^{-2}, -3.80 \times 10^{-2},$ $-3.80 \times 10^{-2}, -3.80 \times 10^{-2}, -9.74 \times 10^{-5},$ $-9.74 \times 10^{-5}, -9.72 \times 10^{-5}, -9.74 \times 10^{-5},$ $-9.73 \times 10^{-5})$	$(9.91 \times 10^{-1}, 9.21 \times 10^{-1}, 8.17 \times 10^{-1},$ $6.25 \times 10^{-1}, 2.05 \times 10^{-1}, -3.74 \times 10^{-2},$ $-3.74 \times 10^{-2}, -3.74 \times 10^{-2}, -3.74 \times 10^{-2},$ $-3.74 \times 10^{-2})$	3.74×10^{-5}	4.07×10^{-1}
SMD9	$(-5.11 \times 10^{-8}, -5.68 \times 10^{-8}, -5.31 \times 10^{-8},$ $-5.85 \times 10^{-8}, -5.58 \times 10^{-8}, 5.40 \times 10^{-8},$ $5.33 \times 10^{-8}, 5.34 \times 10^{-8}, 5.34 \times 10^{-8},$ $5.34 \times 10^{-8})$	$(-3.96 \times 10^{-10}, -6.93 \times 10^{-9}, -2.16 \times 10^{-9},$ $-8.97 \times 10^{-9}, -1.14 \times 10^{-8}, -2.11 \times 10^{-8},$ $-2.19 \times 10^{-8}, -2.19 \times 10^{-8}, -2.19 \times 10^{-8},$ $-2.18 \times 10^{-8})$	9.32×10^{-16}	4.38×10^{-14}
SMD10	$(4.00 \times 10^{-1}, 9.98 \times 10^{-1}, 6.13 \times 10^{-1},$ $7.56 \times 10^{-1}, 7.34 \times 10^{-1}, 5.26 \times 10^{-1},$ $-1.75 \times 10^0, 9.53 \times 10^{-1}, 7.39 \times 10^{-1},$ $-1.74 \times 10^0)$	$(3.32 \times 10^{-1}, -1.19 \times 10^0, 5.38 \times 10^{-1},$ $3.70 \times 10^{-1}, -1.17 \times 10^0, -1.92 \times 10^{-1},$ $-8.07 \times 10^{-1}, -1.58 \times 10^{-1}, -5.78 \times 10^{-1},$ $-1.71 \times 10^{-1})$	3.42×10^1	3.41×10^1
SMD11	$(-5.22 \times 10^{-9}, 3.44 \times 10^{-9}, 3.44 \times 10^{-9},$ $3.44 \times 10^{-9}, 3.44 \times 10^{-9}, -3.98 \times 10^{-1},$ $-3.98 \times 10^{-1}, -4.07 \times 10^{-1}, -3.98 \times 10^{-1},$ $-3.98 \times 10^{-1})$	$(-7.42 \times 10^{-4}, -7.21 \times 10^{-4}, -7.15 \times 10^{-4},$ $-7.05 \times 10^{-4}, -7.32 \times 10^{-4}, 3.68 \times 10^{-1},$ $3.68 \times 10^{-1}, 3.68 \times 10^{-1}, 3.68 \times 10^{-1},$ $3.68 \times 10^{-1})$	4.37×10^{-5}	8.00×10^{-1}
SMD12	$(8.44 \times 10^{-1}, -2.18 \times 10^0, 9.75 \times 10^{-1},$ $1.16 \times 10^0, -2.19 \times 10^0, 9.97 \times 10^{-1},$ $9.95 \times 10^{-1}, 9.81 \times 10^{-1}, 9.96 \times 10^{-1},$ $9.88 \times 10^{-1})$	$(-1.03 \times 10^0, 1.85 \times 10^{-1}, -1.03 \times 10^0,$ $2.29 \times 10^{-1}, 3.46 \times 10^{-1}, -8.39 \times 10^{-2},$ $1.35 \times 10^{-2}, -5.59 \times 10^{-1}, -1.87 \times 10^{-1},$ $2.50 \times 10^{-2})$	6.80×10^1	5.99×10^1

5.4.1. Computation Results of SMD(2 + 3) Problems

The optimal solutions obtained by CFHTTAO and their corresponding performance indicators are presented to give a more direct assessment of its effectiveness on BOPs. Table 8 reports the best computational results obtained by CFHTTAO on each SMD test problem with an upper-level dimension of 2 and a lower-level dimension of 3, allowing a more direct evaluation of its actual performance.

It can be seen from Table 8 that CFHTTAO achieves high solution accuracy on the $(2 + 3)$ -dimensional SMD problems. For example, for SMD3, the upper-level optimal solution obtained by CFHTTAO is $x_u^* = (7.27 \times 10^{-17}, 7.49 \times 10^{-17})$, and the lower-level optimal solution is $x_l^* = (5.24 \times 10^{-17}, 3.61 \times 10^{-17}, 2.50 \times 10^{-17})$. The corresponding UA^* and LA^* of the optimal solution (x_u^*, x_l^*) are 1.56×10^{-32} and 6.23×10^{-34} , respectively, indicating that the deviation from the actual optimum is almost negligible. From the perspectives of both UA^* and LA^* , CFHTTAO exhibits outstanding performance on all SMD test problems under the $(2 + 3)$ -dimensional setting. In addition, for approximately 83.33% of the test problems, the UA^* values are lower than 10^{-10} , further demonstrating the strong optimization capability of CFHTTAO on the $(2 + 3)$ -dimensional SMD problems.

5.4.2. Computation Results of SMD(5 + 5) Problems

As the problem dimension increases, the search space expands exponentially, placing higher demands on the algorithm's global exploration and local exploitation capabilities. Table 9 presents the optimal solutions and performance indicators obtained by CFHTTAO in solving the $(5 + 5)$ -dimensional SMD test problems, further evaluating the performance of algorithm in handling BOPs with increased dimensionality.

The results in Table 9 indicate that, in the $(5 + 5)$ dimensional setting, CFHTTAO still maintains high solution accuracy on most test problems, except for SMD10 and SMD12. Taking SMD1 as an example, the UA^* and LA^* values obtained by CFHTTAO are 4.20×10^{-25} and 3.54×10^{-25} , respectively. For SMD3, the UA^* and LA^* values achieved by CFHTTAO are 8.68×10^{-30} and 1.08×10^{-30} , respectively. As the problem size increases from $(2 + 3)$ dimensions to $(5 + 5)$ dimensions, the total dimensionality doubles, which significantly increases the difficulty of the optimization task. Nevertheless, for most SMD test problems with $(5 + 5)$ dimensions, CFHTTAO still obtains solutions close to the true optimum.

5.4.3. Computation Results of SMD(10 + 10) Problems

BOPs under high-dimensional settings pose severe challenges to optimization algorithms due to the rapid expansion of the search space and the increasingly complex relationships between upper-level and lower-level decision variables. To illustrate the capability of CFHTTAO for solving more complex BOPs, Table 10 shows the optimal solutions and corresponding performance indicators obtained by CFHTTAO for the SMD test problems under the highest-dimensional settings.

It can be observed from Table 10 that increasing the problem dimension to $(10 + 10)$ significantly intensifies the difficulty of BOPs, owing to the enlarged search space and the more complex relationships between upper-level and lower-level variables. Despite these challenges, CFHTTAO still performs well. In terms of UA^* , except for SMD10 and SMD12, CFHTTAO achieves relatively high solution accuracy on the remaining SMD problems. In particular, it maintains a high level of optimization performance on SMD1, SMD3, SMD4, SMD6, and SMD9. In terms of LA^* , CFHTTAO also exhibits good performance on most SMD problems. Although high-dimensional BOPs are inherently difficult to solve, the results show that CFHTTAO has strong scalability for solving more complex high-dimensional BOPs.

5.5. Analysis of Convergence Curves

To visually illustrate the solution accuracy and convergence speed of the algorithms during the iteration process, the convergence curves of the algorithms are presented in this section. Taking all SMD problems with $(2 + 3)$ dimensions as examples, Figure 5 shows the convergence curves of CFHTTAO, HHPSODE, HMDE, BIFOA, SABILDE and BiWOA, in which the upper-level accuracy metric UA is used as the representative metric [6] to illustrate the convergence of algorithms during the iterative process.

As shown in Figure 5, the convergence curves further demonstrate the convergence speed of CFHTTAAO on the SMD(2 + 3) problems. For most test problems, CFHTTAAO exhibits a faster decrease during the early iterations. The convergence curves of CFHTTAAO are generally located below those of the other algorithms and decrease more rapidly, indicating better solution accuracy and faster convergence. Therefore, the convergence curves further confirm the effectiveness of the proposed algorithm.

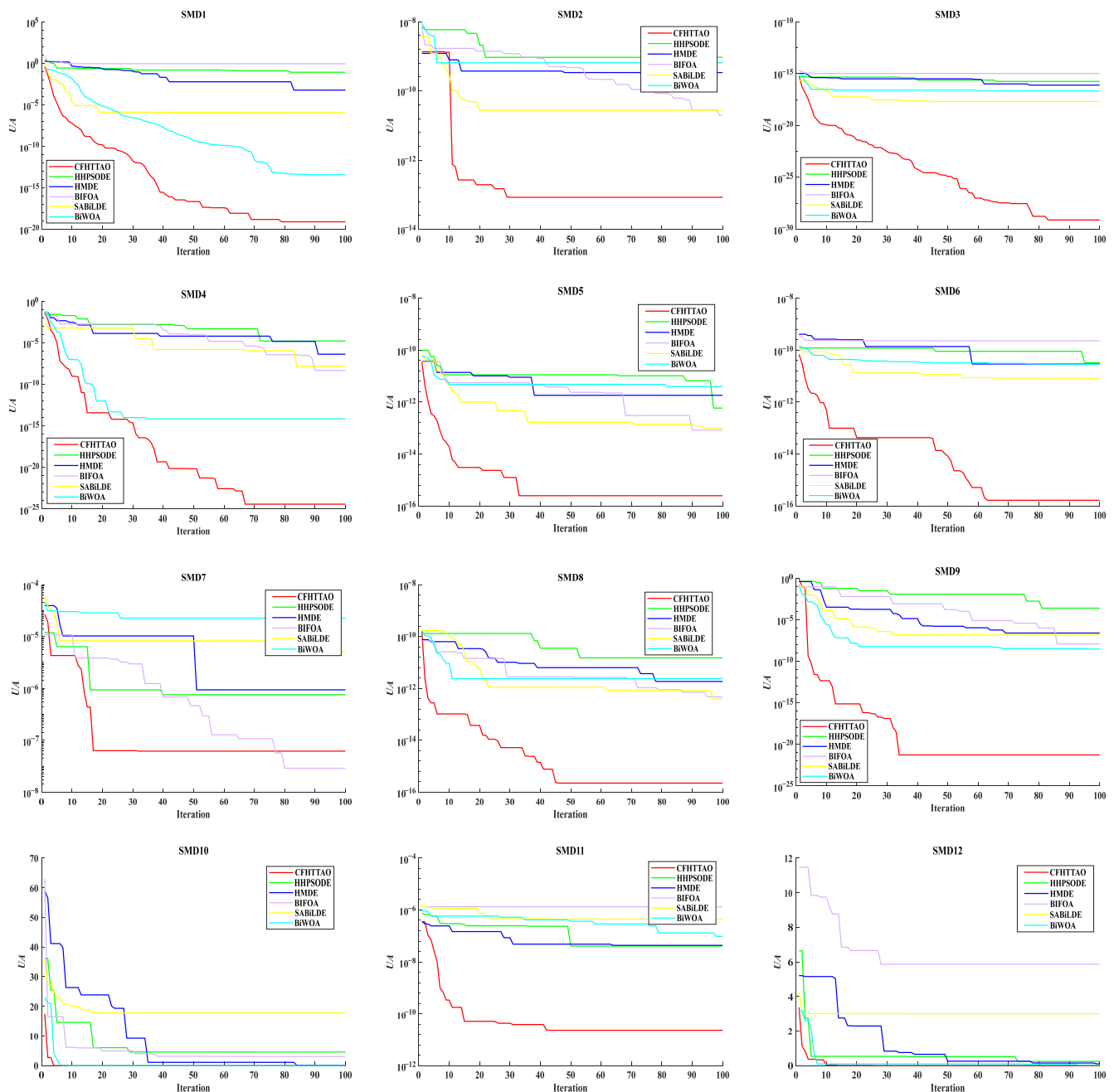


Figure 5. Convergence curves of algorithms on all SMD problems with (2 + 3) dimensions.

5.6. Statistical Analysis

To provide a more comprehensive evaluation of algorithms, it is essential to employ statistical analysis to further examine performance differences among algorithms. The Friedman test is a classical statistical method [58]. In this paper, the Friedman test is employed to statistically analyze the performance of the proposed CFHTTAO and other algorithms on the SMD problems under the dimensional settings of $(2 + 3)$, $(5 + 5)$, and $(10 + 10)$.

In the Friedman test, a smaller mean ranking value indicates better overall optimization performance and superior algorithmic effectiveness. In addition, the significance level is set to 0.05 [58]. A p -value below 0.05 indicates a statistically significant difference. Otherwise, the difference is considered statistically insignificant. Tables 11 and 12 report the ranking results and significance levels, respectively, and MR and R stand for mean rank and rank, respectively. Figure 6 intuitively illustrates the mean rank obtained by each algorithm through bar charts.

Table 11. Ranking results of algorithms by Friedman test.

Problem	Algorithm	UL		LL	
		MR	R	MR	R
SMD(2 + 3)	CFHTTAO	1.08	1	1.00	1
	HHPSODE	5.08	6	4.58	5
	HMDE	3.58	4	3.83	4
	BIFOA	4.67	5	5.67	6
	SABiLDE	3.17	2	3.08	3
	BiWOA	3.42	3	2.83	2
SMD(5 + 5)	CFHTTAO	1.50	1	1.17	1
	HHPSODE	4.92	6	4.58	5
	HMDE	4.17	4	4.17	4
	BIFOA	2.92	2	5.83	6
	SABiLDE	3.33	3	2.83	3
	BiWOA	4.17	4	2.42	2
SMD(10 + 10)	CFHTTAO	1.50	1	1.25	1
	HHPSODE	4.17	5	4.67	4
	HMDE	4.92	6	5.00	5
	BIFOA	3.25	2	5.00	5
	SABiLDE	3.25	2	3.00	3
	BiWOA	3.92	4	2.08	2

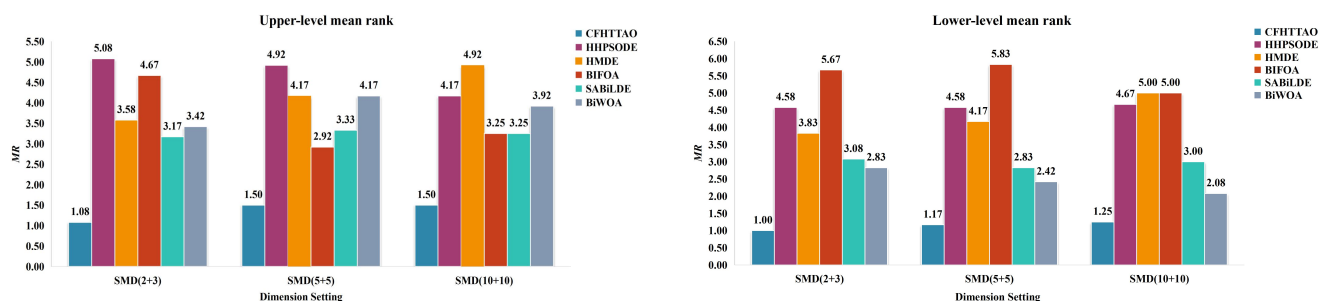


Figure 6. Bar charts of the mean rank.

As shown in Table 11 and Figure 6, CFHTTAO achieves the smallest mean rank under all dimensional settings of SMD(2 + 3), SMD(5 + 5), and SMD(10 + 10), indicating that it consistently delivers the best overall performance across different dimensions. Moreover, CFHTTAO attains the best mean rank in both upper-level and lower-level optimization compared with other hierarchical algorithms. For instance, on SMD(2 + 3), CFHTTAO achieves *MR* values of 1.08 and 1.00 in UL and LL, respectively, ranking first at both levels and outperforming HHPSODE, HMDE, BIFOA, SABILDE, and BiWOA. These results demonstrate the competitive optimization capability of CFHTTAO in solving BOPs.

Table 12. *p*-values obtained from the Friedman test.

Problem	UL	LL
SMD(2 + 3)	3.00×10^{-6}	2.27×10^{-8}
SMD(5 + 5)	1.45×10^{-4}	2.90×10^{-9}
SMD(10 + 10)	3.17×10^{-4}	1.33×10^{-8}

As can be clearly observed from Table 12, all *p*-values obtained for SMD(2 + 3), SMD(5 + 5), and SMD(10 + 10) are smaller than 0.05, suggesting that statistically significant differences exist among the algorithms across the SMD problems under different dimensional settings. Specifically, the largest *p*-value is 3.17×10^{-4} for the upper-level optimization of SMD(10 + 10), while the smallest *p*-value is 2.90×10^{-9} for the lower-level optimization of SMD(5 + 5). Together with the Friedman ranking results, these statistical results further confirm that CFHTTAO achieves performance advantages.

5.7. Analysis of Computational Cost

In this section, the computational cost of algorithms is evaluated by recording the number of function evaluations (FEs) and CPU time. To reflect the ability of the algorithm to reach the acceptable accuracy within a limited computational budget, the number of function evaluations and CPU time are recorded until either the accuracy of 10^{-2} is reached or the maximum number of iterations is exhausted [59,60]. Taking SMD(2 + 3) as an example, Table 13 reports the average and standard deviation of the total number of function evaluations and CPU time obtained by each algorithm.

As shown in Table 13, the FEs of all algorithms are relatively large because of the nested structure of BOPs. In BOPs, the lower-level optimization is embedded in the upper-level optimization, and each upper-level candidate solution usually requires a corresponding lower-level search. The total FEs are accumulated through upper-level and lower-level evaluations. Despite the computational burden caused by the nested structure of BOPs, CFHTTAO still demonstrates competitive computational efficiency. For example, on SMD1, CFHTTAO requires 1.85×10^5 FEs and 4.36 s, which are lower than those of other algorithms. CFHTTAO also demonstrates competitive performance on SMD3, SMD6, SMD8, SMD9, and SMD11. Although CFHTTAO does not always achieve the lowest FEs and CPU time on all problems, it still remains computationally efficient on most test problems.

Table 13. Computational cost results of algorithms for SMD(2 + 3) problems.

Problem		CFHTTAO		HHPSODE		HMDE		BIFOA		SABILDE		BiWOA	
		FEs	CPU	FEs	CPU	FEs	CPU	FEs	CPU	FEs	CPU	FEs	CPU
SMD1	<i>avg</i>	1.85×10^5	4.36×10^0	9.70×10^6	1.72×10^2	6.32×10^6	1.04×10^2	2.94×10^5	4.05×10^1	3.29×10^5	5.23×10^0	1.93×10^5	7.57×10^0
	<i>std</i>	1.19×10^5	2.87×10^0	1.11×10^3	9.50×10^{-1}	2.29×10^6	3.82×10^1	2.35×10^4	2.80×10^0	2.28×10^4	7.70×10^{-1}	4.31×10^4	2.99×10^0
SMD2	<i>avg</i>	3.10×10^5	1.03×10^1	8.06×10^6	1.73×10^2	1.44×10^6	2.45×10^1	1.55×10^5	1.95×10^1	3.17×10^5	4.92×10^0	5.29×10^5	2.84×10^1
	<i>std</i>	5.00×10^4	5.92×10^0	1.82×10^6	7.11×10^1	6.03×10^5	1.11×10^1	5.06×10^4	6.81×10^0	5.55×10^4	7.20×10^{-1}	2.05×10^4	4.19×10^0
SMD3	<i>avg</i>	2.55×10^5	5.65×10^0	8.23×10^6	1.52×10^2	8.76×10^6	1.61×10^2	3.05×10^5	3.59×10^1	4.02×10^5	6.16×10^0	1.66×10^5	8.93×10^0
	<i>std</i>	4.66×10^4	6.80×10^{-1}	2.54×10^6	4.04×10^1	1.19×10^6	3.14×10^1	5.95×10^3	8.94×10^0	8.14×10^4	1.33×10^0	6.92×10^4	3.63×10^0
SMD4	<i>avg</i>	1.48×10^5	3.50×10^0	1.70×10^6	3.02×10^1	9.93×10^5	1.75×10^1	2.40×10^5	3.30×10^1	2.37×10^5	2.86×10^0	2.28×10^5	9.54×10^0
	<i>std</i>	3.38×10^3	7.00×10^{-2}	9.62×10^5	1.66×10^1	1.47×10^5	1.55×10^0	6.95×10^3	9.73×10^0	4.06×10^4	1.28×10^0	1.22×10^5	4.30×10^0
SMD5	<i>avg</i>	2.18×10^5	4.90×10^0	3.20×10^6	5.50×10^1	1.37×10^6	2.17×10^1	2.41×10^5	3.44×10^1	5.45×10^5	8.08×10^0	4.95×10^5	2.49×10^1
	<i>std</i>	4.19×10^4	1.22×10^0	1.97×10^6	3.32×10^1	4.75×10^5	7.34×10^0	1.72×10^4	6.90×10^{-1}	1.96×10^5	2.70×10^0	9.58×10^4	2.83×10^0
SMD6	<i>avg</i>	2.81×10^5	6.04×10^0	6.93×10^6	1.28×10^2	6.75×10^6	1.20×10^2	2.92×10^5	4.44×10^1	1.72×10^6	2.64×10^1	5.45×10^5	3.47×10^1
	<i>std</i>	6.54×10^4	1.00×10^0	2.70×10^6	5.09×10^1	2.51×10^6	4.55×10^1	1.74×10^4	4.50×10^0	3.86×10^5	5.94×10^0	3.50×10^4	1.37×10^1
SMD7	<i>avg</i>	2.06×10^5	4.62×10^0	7.35×10^6	1.37×10^2	5.04×10^5	8.58×10^0	2.24×10^5	2.55×10^1	2.16×10^5	3.35×10^0	5.06×10^5	2.34×10^1
	<i>std</i>	5.32×10^4	1.06×10^0	1.59×10^6	3.13×10^1	5.14×10^4	7.60×10^{-1}	4.12×10^4	2.71×10^0	7.85×10^4	1.13×10^0	1.66×10^5	1.40×10^1
SMD8	<i>avg</i>	2.45×10^5	5.31×10^0	6.52×10^6	1.22×10^2	2.08×10^6	3.36×10^1	2.53×10^5	3.61×10^1	5.74×10^5	8.57×10^0	4.58×10^5	1.96×10^1
	<i>std</i>	6.45×10^4	1.44×10^0	3.34×10^6	4.45×10^1	1.48×10^6	2.33×10^1	1.47×10^4	2.69×10^0	1.43×10^5	2.21×10^0	5.46×10^4	2.90×10^0
SMD9	<i>avg</i>	1.61×10^5	5.50×10^0	7.23×10^6	1.22×10^2	3.19×10^6	5.45×10^1	2.61×10^5	4.01×10^1	2.18×10^6	3.95×10^1	1.95×10^5	1.37×10^1
	<i>std</i>	3.32×10^4	3.21×10^0	8.23×10^6	1.39×10^2	1.08×10^6	1.87×10^1	1.68×10^3	4.87×10^0	3.20×10^6	5.83×10^1	4.84×10^4	3.44×10^0
SMD10	<i>avg</i>	8.30×10^5	1.69×10^1	1.78×10^7	3.49×10^2	4.07×10^6	1.01×10^2	3.04×10^5	4.67×10^1	2.08×10^6	3.83×10^1	4.98×10^5	6.33×10^1
	<i>std</i>	9.87×10^4	2.64×10^0	6.97×10^5	1.20×10^1	2.30×10^6	5.70×10^1	8.44×10^3	4.51×10^0	3.94×10^5	6.72×10^0	1.51×10^5	8.01×10^0
SMD11	<i>avg</i>	1.79×10^5	4.36×10^0	1.70×10^7	3.30×10^2	1.22×10^6	2.64×10^1	3.05×10^5	3.86×10^1	2.13×10^6	3.60×10^1	5.68×10^5	4.03×10^1
	<i>std</i>	2.81×10^3	4.70×10^{-1}	6.74×10^5	2.34×10^1	9.67×10^5	1.98×10^1	5.95×10^3	1.20×10^1	1.01×10^6	1.47×10^1	2.10×10^4	4.07×10^0
SMD12	<i>avg</i>	1.74×10^6	4.56×10^1	1.74×10^7	3.60×10^2	3.91×10^6	1.05×10^2	3.16×10^5	4.54×10^1	2.10×10^6	4.12×10^1	6.12×10^5	8.06×10^1
	<i>std</i>	6.02×10^5	1.64×10^1	1.19×10^6	1.54×10^1	5.02×10^6	1.34×10^2	1.17×10^4	3.84×10^0	5.99×10^5	1.05×10^1	2.05×10^4	1.39×10^1

6. Practical Application

To further verify the effectiveness of the proposed CFHTTAO in practical applications, it is applied to solve the inverse optimal control problem. The inverse optimal control problem exhibits a nested decision-making structure, and the solution process involves two interrelated optimization problems at the upper and lower levels. As described below, the inverse optimal control problem is regarded as a bilevel optimization problem consisting of upper-level and lower-level problems [6].

In the upper level of the inverse optimal control problem, the objective is to minimize the deviation (E) between the computed optimal control and the obtained dataset, which is formulated as follows.

$$\min_{(\alpha_1, \alpha_2, \alpha_3)} E = \int_0^1 [(z_1(m) - z_1^*(m))^2 + (z_2(m) - z_2^*(m))^2 + (u(m) - u^*(m))^2] dm \quad (15)$$

where α_1 , α_2 , and α_3 denote the upper-level decision variables constrained by the equality constraint $\alpha_1 + \alpha_2 + \alpha_3 = 25.1$. $z_1(m)$, $z_2(m)$, and $u(m)$ are influenced by the lower-level decision variables β_1 , β_2 , and β_3 , as expressed below.

$$\begin{aligned} z_1(m) &= m^5 + \beta_3 m^4 + \beta_2 m^3 + \beta_1 m^2 - m, \\ z_2(m) &= 5m^4 + 4\beta_3 m^3 + 3\beta_2 m^2 + 2\beta_1 m - 1, \\ u(m) &= 5m^4 + (4\beta_3 + 20)m^3 + (3\beta_2 + 12\beta_3)m^2 + (2\beta_1 + 6\beta_2)m - 1 + 2\beta_1. \end{aligned} \quad (16)$$

$z_1^*(m)$, $z_2^*(m)$, and $u^*(m)$ are given by [61]

$$\begin{aligned} z_1^*(m) &= m^5 - 12.1543m^4 + 19.2045m^3 - 7.3436m^2 - m, \\ z_2^*(m) &= 5m^4 - 48.6172m^3 + 57.6135m^2 - 14.687m - 1, \\ u^*(m) &= 5m^4 - 28.62m^3 - 88.238m^2 + 100.54m - 15.69. \end{aligned} \quad (17)$$

In the lower level of the inverse optimal control problem, the objective is to minimize the continuous time cost function (J), which is expressed as follows.

$$\min_{(\beta_1, \beta_2, \beta_3)} J = \mathbf{C} \mathbf{A}^T \mathbf{B} \quad (18)$$

where β_1 , β_2 , and β_3 denote lower-level decision variables, $\mathbf{C}$ denotes the row vector composed of the upper-level decision variables, $\mathbf{A}$ is a matrix, and $\mathbf{B}$ is constructed from the lower-level decision variables [6]. They are defined as

$$\mathbf{C} = [\alpha_1 \quad \alpha_2 \quad \alpha_3], \quad \mathbf{A} = \begin{bmatrix} 0.138 & 1.778 & 73.92 \\ -0.250 & 1.333 & 37.33 \\ -0.177 & 2.285 & 54.28 \\ -0.130 & 3.000 & 115.0 \\ 0.333 & 3.000 & 27.00 \\ 0.250 & 4.000 & 64.00 \\ 0.285 & 3.200 & 35.20 \\ 0.200 & 1.333 & 9.330 \\ 0.142 & 1.800 & 22.80 \\ 0.111 & 2.285 & 47.08 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_1 \beta_2 \\ \beta_2 \beta_3 \\ \beta_1 \beta_3 \\ \beta_1^2 \\ \beta_2^2 \\ \beta_3^2 \end{bmatrix} \quad (19)$$

The above inverse optimal control problem is solved using CFHTTAO, HHPSODE, HMDE, BIFOA, SABILDE, and BiWOA. For a fair comparison, all algorithms are tested

under the same experimental conditions, and the corresponding results are reported in Table 14. The experimental results are rounded to two decimal places.

Table 14. Comparison results of algorithms for the inverse optimal control problem.

Algorithm	E	J	α_1	α_2	α_3	β_1	β_2	β_3
CFHTTAO	0.04	−16.11	10.97	14.00	0.12	−7.28	18.92	−11.99
HHPSODE	5.28	−16.06	12.22	12.72	0.16	−5.65	18.40	−12.27
HMDE	2.74	−16.07	15.90	9.09	0.11	−8.89	20.77	−12.75
BIFOA	3.40	−14.36	18.31	6.71	0.08	−8.65	21.77	−13.61
SABiLDE	0.60	−15.32	13.26	11.73	0.11	−6.69	17.73	−11.35
BiWOA	0.14	−10.42	16.30	8.73	0.07	−7.00	18.79	−12.04

According to the experimental results in Table 14, CFHTTAO shows a clear advantage in solving the inverse optimal control problem. CFHTTAO obtains an upper-level objective value E of 0.04 and a lower-level objective value J of −16.11, both of which are smaller than those obtained by other algorithms. Since both the upper-level and lower-level problems of this inverse optimal control problem are minimization problems, the smaller values of E and J indicate that CFHTTAO provides superior performance. Therefore, the results demonstrate the effectiveness and competitiveness of CFHTTAO in solving the inverse optimal control problem and further support its applicability to practical BOPs.

7. Conclusions

In this paper, CFHTTAO is proposed for solving complex BOPs. Considering the nested structure of BOPs, the upper-level framework of CFHTTAO serves as the main program and the lower-level framework of CFHTTAO is implemented as a nested subprogram. On this basis, improvement mechanisms are designed to enhance the solution capability of the algorithm. Specifically, the knowledge-sharing mechanism enables superior information to be shared among different units, the KNN-based approximation technique is used to assist lower-level optimization, the coordinated feedback mechanism adaptively selects a suitable lower-level search method, and the elite learning evolution mechanism helps reduce the risk of being trapped in local optima. To evaluate the performance of CFHTTAO, experiments are conducted on SMD problems with gradually increasing dimensional settings, namely SMD(2 + 3), SMD(5 + 5), and SMD(10 + 10). Experimental results show that CFHTTAO outperforms other algorithms developed for solving BOPs. Moreover, as the dimensionality of the SMD problems increases, CFHTTAO still maintains competitive performance. Finally, CFHTTAO is applied to the inverse optimal control problem, demonstrating its practical applicability to bilevel optimization. Future research could be extended to solve multi-objective BOPs and multi-follower BOPs.

Author Contributions: Conceptualization, X.Y. and Y.L.; methodology, X.Y. and Y.L.; software, X.Y. and J.L.; validation, Y.L., J.L., and H.Z.; formal analysis, X.Y., Y.L. and J.L.; investigation, X.Y. and Y.L.; resources, H.Z. and Q.G.; data curation, X.Y. and J.L.; writing—original draft preparation, X.Y. and Y.L.; writing—review and editing, X.Y. and J.L.; visualization, H.Z. and Q.G.; supervision, H.Z. and Q.G.; project administration, Y.L.; funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 72104069), Henan Science and Technology Department (Nos. 182102310886 and 162102110109), and Henan University Research and Practice Project on Undergraduate Teaching Reform (No. HDXJJG2026-108).

Data Availability Statement: The original contributions presented in this study are included in the article. Future inquiries can be directed to the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

BOPs	Bilevel optimization problems
TTAO	Triangulation topology aggregation optimizer
CFHTTAO	Coordinated feedback-based hierarchical triangulation topology aggregation optimizer
PSO	Particle swarm optimization
DE	Differential evolution
HHPSODE	Hierarchical hybrid particle swarm optimization and differential evolution based algorithm
HMDE	Hierarchical modified differential evolution algorithm
BIFOA	Bilevel improved fruit fly optimization algorithm
KNN	k-nearest neighbors

References

- Escudero, L.F.; Monge, J.F.; Rodríguez-Chía, A.M. On pricing-based equilibrium for network expansion planning. A multi-period bilevel approach under uncertainty. *Eur. J. Oper. Res.* **2020**, *287*, 262–279. <https://doi.org/10.1016/j.ejor.2020.03.048>.
- Calvete, H.I.; Galé, C.; Iranzo, J.A.; Camacho-Vallejo, J.-F.; Casas-Ramírez, M.-S. A matheuristic for solving the bilevel approach of the facility location problem with cardinality constraints and preferences. *Comput. Oper. Res.* **2020**, *124*, 105066. <https://doi.org/10.1016/j.cor.2020.105066>.
- Janngam, K.; Suantai, S.; Cho, Y.J.; Kaewkhao, A.; Wattanataweekul, R. A novel inertial viscosity algorithm for bilevel optimization problems applied to classification problems. *Mathematics* **2023**, *11*, 3241. <https://doi.org/10.3390/math11143241>.
- Camacho-Vallejo, J.-F.; Dávila, D.; Nucamendi-Guillén, S. A hierarchized green supply chain with customer selection, routing, and nearshoring. *Comput. Ind. Eng.* **2023**, *178*, 109151. <https://doi.org/10.1016/j.cie.2023.109151>.
- Casella, V.; Ferro, G.; Parodi, L.; Robba, M. Maximizing shared benefits in renewable energy communities: A Bilevel optimization model. *Appl. Energy* **2025**, *386*, 125562. <https://doi.org/10.1016/j.apenergy.2025.125562>.
- Lin, L.; Liu, T.; Zhang, H.; Leng, J.; Wei, L.; Liu, Q. Combined knowledge transfer and adaptive coordinate systems approach for evolutionary bilevel optimization. *Expert Syst. Appl.* **2023**, *228*, 120309. <https://doi.org/10.1016/j.eswa.2023.120309>.
- Camacho-Vallejo, J.-F.; Corpus, C.; Villegas, J.G. Metaheuristics for bilevel optimization: A comprehensive review. *Comput. Oper. Res.* **2024**, *161*, 106410. <https://doi.org/10.1016/j.cor.2023.106410>.
- Sinha, A.; Soun, T.; Deb, K. Using Karush-Kuhn-Tucker proximity measure for solving bilevel optimization problems. *Swarm Evol. Comput.* **2019**, *44*, 496–510. <https://doi.org/10.1016/j.swevo.2018.06.004>.
- Lai, Y.; Liu, H.-L.; Xu, Y.; Chen, L. Elite knowledge transfer within lower-level searches for bilevel optimization. *Swarm Evol. Comput.* **2025**, *98*, 102097. <https://doi.org/10.1016/j.swevo.2025.102097>.
- Zhou, X.; Tian, J.; Wang, Z.; Yang, C.; Huang, T.; Xu, X. Nonlinear bilevel programming approach for decentralized supply chain using a hybrid state transition algorithm. *Knowl.-Based Syst.* **2022**, *240*, 108119. <https://doi.org/10.1016/j.knosys.2022.108119>.
- Yang, D.; Jiao, J.R.; Ji, Y.; Du, G.; Helo, P.; Valente, A. Joint optimization for coordinated configuration of product families and supply chains by a leader-follower Stackelberg game. *Eur. J. Oper. Res.* **2015**, *246*, 263–280. <https://doi.org/10.1016/j.ejor.2015.04.022>.
- Lu, F.; Yan, T.; Bi, H.; Feng, M.; Wang, S.; Huang, M. A bilevel whale optimization algorithm for risk management scheduling of information technology projects considering outsourcing. *Knowl.-Based Syst.* **2022**, *235*, 107600. <https://doi.org/10.1016/j.knosys.2021.107600>.
- Jia, Y.-H.; Mei, Y.; Zhang, M. A bilevel ant colony optimization algorithm for capacitated electric vehicle routing problem. *IEEE Trans. Cybern.* **2021**, *52*, 10855–10868. <https://doi.org/10.1109/TCYB.2021.3069942>.
- Zhao, S.; Zhang, T.; Cai, L.; Yang, R. Triangulation topology aggregation optimizer: A novel mathematics-based meta-heuristic algorithm for continuous optimization and engineering applications. *Expert Syst. Appl.* **2024**, *238*, 121744. <https://doi.org/10.1016/j.eswa.2023.121744>.

15. Siva Subramanian, R.; Nithya, T.; Sudha, K.; Dinesh, M.G. Enhanced network security through an intelligent deep learning-based intrusion detection system with optimized performance. *Knowl.-Based Syst.* **2025**, 114322. <https://doi.org/10.1016/j.knosys.2025.114322>.
16. Dahou, A.; Abd Elaziz, M.; Mohamed, H.; Dahou, A.H.; Al-qaness, M.A.A.; Ghetas, M.; Ewess, A.; Zheng, Z. Linguistic feature fusion for Arabic fake news detection and named entity recognition using reinforcement learning and swarm optimization. *Neurocomputing* **2024**, 598, 128078. <https://doi.org/10.1016/j.neucom.2024.128078>.
17. Jeroslow, R.G. The polynomial hierarchy and a simple model for competitive analysis. *Math. Program.* **1985**, 32, 146–164. <https://doi.org/10.1007/BF01586088>.
18. Prokopyev, O.A.; Ralphs, T.K. On the complexity of finding locally optimal solutions in bilevel linear optimization. *Oper. Res.* **2026**. <https://doi.org/10.1287/opre.2024.1411>.
19. Ma, W.; Wang, M.; Zhu, X. Hybrid particle swarm optimization and differential evolution algorithm for bi-level programming problem and its application to pricing and lot-sizing decisions. *J. Intell. Manuf.* **2015**, 26, 471–483. <https://doi.org/10.1007/s10845-013-0803-5>.
20. Wang, M.; Zhang, R.; Zhu, X. A bi-level programming approach to the decision problems in a vendor-buyer eco-friendly supply chain. *Comput. Ind. Eng.* **2017**, 105, 299–312. <https://doi.org/10.1016/j.cie.2017.01.008>.
21. Wang, G.; Ma, L.; Chen, J. A bilevel improved fruit fly optimization algorithm for the nonlinear bilevel programming problem. *Knowl.-Based Syst.* **2017**, 138, 113–123. <https://doi.org/10.1016/j.knosys.2017.09.038>.
22. Wu, L.; Liu, Z.; Wei, H.-L.; Wang, R. An efficient bilevel differential evolution algorithm with adaptation of lower level population size and search radius. *Memetic Comput.* **2021**, 13, 227–247. <https://doi.org/10.1007/s12293-021-00335-8>.
23. Camacho-Vallejo, J.-F.; Dávila, D. Nested evolutionary algorithms for solving a bi-level warehouse location problem that considers inventory decisions. *Ann. Oper. Res.* **2026**, 358, 895–928. <https://doi.org/10.1007/s10479-024-06182-3>.
24. Wang, X.; Niu, L.; Li, Z.; Tang, P.; Liu, X.; Mou, P. Bilevel interactive optimization for low-carbon product family design and multi-factory collaborative production configuration. *Adv. Eng. Inform.* **2026**, 69, 104052. <https://doi.org/10.1016/j.aei.2025.104052>.
25. Tang, X.; Jia, C.; Qu, P. EAPO: A Multi-Strategy-Enhanced Artificial Protozoa Optimizer and Its Application to 3D UAV Path Planning. *Mathematics* **2025**, 14, 153. <https://doi.org/10.3390/math14010153>.
26. Yu, X.; Fang, Y. A Multi-Strategy Improved Catch Fish Optimization Algorithm for Microgrid Scheduling Optimization and Real-World Engineering Applications. *Mathematics* **2026**, 14, 1342. <https://doi.org/10.3390/math14081342>.
27. Xiao, Y.; Cui, H.; Khurma, R.A.; Castillo, P.A. Artificial lemming algorithm: A novel bionic meta-heuristic technique for solving real-world engineering optimization problems. *Artif. Intell. Rev.* **2025**, 58, 84. <https://doi.org/10.1007/s10462-024-11023-7>.
28. Guo, Q.; Yang, X.; Li, K.; Li, D. Parameters identification of magnetorheological damper based on particle swarm optimization algorithm. *Eng. Appl. Artif. Intell.* **2025**, 143, 110016. <https://doi.org/10.1016/j.engappai.2025.110016>.
29. Dang, Q.; Zhang, G.; Wang, L.; Yang, S.; Zhan, T. A generative adversarial networks model based evolutionary algorithm for multimodal multi-objective optimization. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**. <https://doi.org/10.1109/TETCI.2024.3397996>.
30. Abd Elaziz, M.; Essa, F.A.; Khalil, H.A.; El-Sebaey, M.S.; Khedr, M.; Elsheikh, A. Productivity prediction of a spherical distiller using a machine learning model and triangulation topology aggregation optimizer. *Desalination* **2024**, 585, 117744. <https://doi.org/10.1016/j.desal.2024.117744>.
31. Khamies, M.; Hashem, M.; Arya, Y.; Mahmoud, A.A. Triangulation topology aggregation optimizer-based ADRC-PR strategy for frequency stability improvement of hybrid-renewable power grid. *Chaos Solitons Fractals* **2026**, 208, 118201. <https://doi.org/10.1016/j.chaos.2026.118201>.
32. Chen, Y.; Chi, H.; Ji, Z.; Li, G.; Gou, L. Optimal control and operation strategy for an integrated photovoltaic/thermal-pressure retarded osmosis-solid oxide fuel cell system. *Renew. Energy* **2025**, 254, 123749. <https://doi.org/10.1016/j.renene.2025.123749>.
33. Wang, Z.; Zhang, Y.; Hu, J.; Zhang, Y.; Li, L.; Li, W. A hybrid ensemble-optimized bidirectional gated recurrent unit method for short-term load forecasting of electric vehicle charging stations. *J. Energy Storage* **2026**, 152, 120768. <https://doi.org/10.1016/j.est.2026.120768>.
34. Xu, Z.; Li, S.; Guo, Y.; Qiao, R. STL-Twinsformer-TTAO: A short-term photovoltaic power forecasting model based on seasonal-trend interactive learning. *Renew. Energy* **2026**, 263, 125462. <https://doi.org/10.1016/j.renene.2026.125462>.
35. Qu, Z.; Meng, Y.; Hou, X.; Chi, R.; Ai, Y.; Wu, Z. Integrated energy short-term multivariate load forecasting based on PatchTST secondary decoupling reconstruction for progressive layered extraction multi-task learning network. *Expert Syst. Appl.* **2025**, 269, 126446. <https://doi.org/10.1016/j.eswa.2025.126446>.
36. Zhang, L.; Niu, B.; Huang, H.; Zhao, Y.; Liu, J. AITTAO: An improved triangular topology aggregation optimizer for parameter optimization and real-world problem processing. *Clust. Comput.* **2025**, 28, 823. <https://doi.org/10.1007/s10586-025-05306-z>.
37. Vidhyalakshmi, M.K.; Unhelkar, B.; Kshirsagar, P.R.; Thiagarajan, R. Enhanced inter-camera person re-identification leveraging mixed-order relation-aware recurrent neural network. *Neurocomputing* **2025**, 638, 130123. <https://doi.org/10.1016/j.neucom.2025.130123>.

38. Olan, F.; Arakpogun, E.O.; Suklan, J.; Nakpodia, F.; Damij, N.; Jayawickrama, U. Artificial intelligence and knowledge sharing: Contributing factors to organizational performance. *J. Bus. Res.* **2022**, *145*, 605–615. <https://doi.org/10.1016/j.jbusres.2022.03.008>.
39. Ertek, G.; Tokdemir, G.; Sevinç, M.; Tunç, M.M. New knowledge in strategic management through visually mining semantic networks. *Inf. Syst. Front.* **2017**, *19*, 165–185. <https://doi.org/10.1007/s10796-015-9591-0>.
40. Xu, F.; Li, H.; Pun, C.-M.; Hu, H.; Li, Y.; Song, Y.; Gao, H. A new global best guided artificial bee colony algorithm with application in robot path planning. *Appl. Soft Comput.* **2020**, *88*, 106037. <https://doi.org/10.1016/j.asoc.2019.106037>.
41. Deng, L.; Liu, S. A sine cosine algorithm guided by elite pool strategy for global optimization. *Appl. Soft Comput.* **2024**, *164*, 111946. <https://doi.org/10.1016/j.asoc.2024.111946>.
42. Angelo, J.S.; Krempser, E.; Barbosa, H.J.C. Differential evolution assisted by a surrogate model for bilevel programming problems. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 1784–1791. <https://doi.org/10.1109/CEC.2014.6900529>.
43. Shepard, D. A two-dimensional interpolation function for irregularly-spaced data. In Proceedings of the 1968 23rd ACM National Conference, New York, NY, USA, 27–29 August 1968; pp. 517–524. <https://doi.org/10.1145/800186.810616>.
44. Priya, G.V.; Ganguly, S. Multi-swarm surrogate model assisted PSO algorithm to minimize distribution network energy losses. *Appl. Soft Comput.* **2024**, *159*, 111616. <https://doi.org/10.1016/j.asoc.2024.111616>.
45. Wang, J.; Tang, H.; Lei, D. A feedback-based artificial bee colony algorithm for energy-efficient flexible flow shop scheduling problem with batch processing machines. *Appl. Soft Comput.* **2024**, *153*, 111254. <https://doi.org/10.1016/j.asoc.2024.111254>.
46. Lan, K.-T.; Lan, C.-H. Notes on the distinction of Gaussian and Cauchy mutations. In Proceedings of the 2008 Eighth International Conference on Intelligent Systems Design and Applications, Kaohsiung, Taiwan, 26–28 November 2008; Volume 1, pp. 272–277. <https://doi.org/10.1109/ISDA.2008.237>.
47. Dong, Y.; Li, M.; Zhou, M. Multi-threshold image segmentation based on the improved Dragonfly algorithm. *Mathematics* **2024**, *12*, 854. <https://doi.org/10.3390/math12060854>.
48. Guo, W.; Zhang, M.; Dai, F.; Xiao, X.; Zhang, L. A Dual-Mode Elite-Guided Multi-Objective Marine Predator Algorithm Enhanced by GNG Network and Directional Sampling. *Neurocomputing* **2026**, *685*, 133590. <https://doi.org/10.1016/j.neucom.2026.133590>.
49. Zhou, X.; Lu, J.; Huang, J.; Zhong, M.; Wang, M. Enhancing artificial bee colony algorithm with multi-elite guidance. *Inf. Sci.* **2021**, *543*, 242–258. <https://doi.org/10.1016/j.ins.2020.07.037>.
50. Long, W.; Jiao, J.; Liang, X.; Wu, T.; Xu, M.; Cai, S. Pinhole-imaging-based learning butterfly optimization algorithm for global optimization and feature selection. *Appl. Soft Comput.* **2021**, *103*, 107146. <https://doi.org/10.1016/j.asoc.2021.107146>.
51. Liu, J.; Mei, Y.; Li, X. An analysis of the inertia weight parameter for binary particle swarm optimization. *IEEE Trans. Evol. Comput.* **2015**, *20*, 666–681. <https://doi.org/10.1109/TEVC.2015.2503422>.
52. Liu, R.; Gao, J.; Zhang, J.; Meng, D.; Lin, Z. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 10045–10067. <https://doi.org/10.1109/TPAMI.2021.3132674>.
53. Sinha, A.; Malo, P.; Deb, K. Test problem construction for single-objective bilevel optimization. *Evol. Comput.* **2014**, *22*, 439–477. https://doi.org/10.1162/EVCO_a_00116.
54. Chen, L.; Liu, H.-L.; Tan, K.C.; Li, K. Transfer learning-based parallel evolutionary algorithm framework for bilevel optimization. *IEEE Trans. Evol. Comput.* **2021**, *26*, 115–129. <https://doi.org/10.1109/TEVC.2021.3095313>.
55. Sinha, A.; Malo, P.; Deb, K. Evolutionary algorithm for bilevel optimization using approximations of the lower level optimal solution mapping. *Eur. J. Oper. Res.* **2017**, *257*, 395–411. <https://doi.org/10.1016/j.ejor.2016.08.027>.
56. He, X.; Zhou, Y.; Chen, Z. Evolutionary bilevel optimization based on covariance matrix adaptation. *IEEE Trans. Evol. Comput.* **2018**, *23*, 258–272. <https://doi.org/10.1109/TEVC.2018.2849000>.
57. Xu, J.; Nelson, B.L.; Hong, L.J. An adaptive hyperbox algorithm for high-dimensional discrete optimization via simulation problems. *INFORMS J. Comput.* **2013**, *25*, 133–146. <https://doi.org/10.1287/ijoc.1110.0481>.
58. Houssein, E.H.; Abdalkarim, N.; Samee, N.A.; Alabdulhafith, M.; Mohamed, E. Improved Kepler Optimization Algorithm for enhanced feature selection in liver disease classification. *Knowl.-Based Syst.* **2024**, *297*, 111960. <https://doi.org/10.1016/j.knosys.2024.111960>.
59. Dong, Y.; Hansen, P.C.; Hochstenbach, M.E.; Brogaard Riis, N.A. Fixing nonconvergence of algebraic iterative reconstruction with an unmatched backprojector. *SIAM J. Sci. Comput.* **2019**, *41*, A1822–A1839. <https://doi.org/10.1137/18M1206448>.

60. Bock, A.A.; Andersen, M.S. A new matrix truncation method for improving approximate factorization preconditioners. *Numer. Linear Algebra Appl.* **2025**, *32*, e70012. <https://doi.org/10.1002/nla.70012>.
61. Suryan, V.; Sinha, A.; Malo, P.; Deb, K. Handling inverse optimal control problems using evolutionary bilevel optimization. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 1893–1900. <https://doi.org/10.1109/CEC.2016.7744019>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.