Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 21, No. 1, 2026

Tuckman Optimization Algorithm: A novel metaheuristic inspired by Tuckman's Stages of Group Development for solving benchmark and engineering problems

Meifeng Shi, Faxiang Wang *

College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, 400054, China

Abstract: This paper proposes a novel Tuckman Optimization Algorithm (TOA) based on Tuckman's Stages of Group Development for solving optimization and engineering design problems. TOA is the first to embed this theory into an intelligent optimization framework, simulating team dynamics to guide population search behavior. Its core innovations include a team control parameter and a team structure reconfiguration mechanism. The control parameter dynamically adjusts stage weights and evolutionary strategies based on iteration progress, enhancing adaptability and convergence stability. The team structure reconfiguration mechanism, based on a logistic growth model, replaces inferior individuals to maintain population vitality and avoid premature convergence. Experimental results on the 10-dimensional CEC2017, CEC2020, and CEC2022 benchmark suites show that TOA outperforms three baseline algorithms, ten advanced metaheuristics, and the CEC2017 winner LSHADE SPACMA in both accuracy and robustness. Additional tests on engineering design problems further validate TOA's effectiveness and broad applicability in real-world optimization scenarios.

Keywords: Tuckman's Stages of Group Development; Meta-heuristic; Optimization Problems; Engineering Applications.

1. Introduction

With the increasing scarcity of global resources and the growing complexity of demand, optimization problems have become significantly more intricate across various application domains [1]. This is particularly evident in critical tasks such as enhancing production efficiency, optimizing resource scheduling, and managing supply chains, which increasingly involve large-scale and highly uncertain optimization challenges. Traditional mathematical optimization methods—such as the conjugate gradient method and quasi-Newton methods—while effective for analytically tractable problems, often struggle with high-dimensional nonlinear problems or complex constrained optimization tasks [2]. These methods rely heavily on precise gradient information and well-defined mathematical models, making it difficult to obtain robust global optima in such scenarios.

In this context, metaheuristic algorithms have emerged as powerful tools for tackling complex optimization problems owing to their low dependency on problem structure and gradient information. These algorithms typically emulate evolutionary or socio-inspired mechanisms and exhibit strong global search capabilities, making them particularly effective for solving non-convex, discontinuous, or multimodal problems. However, existing metaheuristic algorithms still face significant challenges, including premature convergence, loss of population diversity, and an imbalance between exploration and exploitation. These issues become increasingly pronounced when dealing with large-scale optimization tasks of higher complexity, ultimately constraining solution quality and algorithmic stability.

Tuckman's Stages of Group Development is an influential theory in social psychology that divides team evolution into five stages: Forming, Storming, Norming, Performing, Adjourning. The first three stages, in particular, reveal key dynamics in the gradual establishment of effective team collaboration mechanisms [3]. This evolutionary process closely mirrors the transition from initial search to convergence in optimization algorithms, providing a solid theoretical foundation for developing novel search strategies [4].

Inspired by Tuckman's Stages of Group Development, a novel Tuckman Team Maturity Development Algorithm named TOA is proposed in this paper. TOA integrates behavioral mechanisms corresponding to each developmental stage as follows to enhance optimization performance.

- Forming Stage: Enhances population diversity and global exploration by mimicking the initial process of mutual acquaintance and adaptation among team members.
- Storming Stage: Facilitates escape from the local optimum by emulating dynamic interactions such as internal conflicts, opinion clashes, and role identification.
- Norming Stage: Improves convergence accuracy and stability via simulating efficient collaborative behaviors that emerge after consensus is reached and goal alignment is strengthened.

Additionally, TOA introduces a team control parameter and a team structure reconfiguration mechanism. The team control parameter is used to dynamically adjust the weight distribution and evolutionary strategies of each stage based on the current iteration progress, further enhancing the adaptability and convergence stability. The team structure reconfiguration mechanism based on a logistic growth model is designed to replace inferior individuals, ensuring population vitality and preventing premature convergence to local Optimization.

To rigorously evaluate the performance of TOA, comprehensive experiments are conducted using benchmark benchmark suite CEC2017, CEC2020, and CEC2022 against

leading algorithms including:

- Three foundational metaheuristics: Particle Swarm Optimization (PSO)[5], Whale Optimization Algorithm (WOA)[6], and Harris Hawks Optimization (HHO)[7].
- Ten state-of-the-art algorithms: Dung Beetle Optimizer (DBO)[8], Great Wall Construction Algorithm (GWCA)[9], Artemisinin Optimization (AO)[10], Triangulation Topology Aggregation Optimizer (TTAO)[11], Black-winged Kite Algorithm (BKA)[12], Blood-sucking leech Optimizer (BSLO)[13], Electric Eel Foraging Optimization (EEFO)[14], Fata Morgana Algorithm(FATA)[15], Coral Fish Optimization Algorithm (CFOA)[16], and Walrus Optimizer (WO)[17].
- The CEC 2017 winner: LSHADE-SPACMA[18].

Additionally, all competing algorithms are applied to five real-world engineering problems, such as pressure vessel design and photovoltaic module parameter estimation. Experimental results confirm TOA's statistically superior solution accuracy, convergence stability, and robustness across benchmark suite, demonstrating strong applicability in complex engineering domains. The experimental results demonstrate that TOA achieves superior solution accuracy, average performance, and robustness, indicating its strong applicability in complex practical engineering domains.

The rest of the paper is organized as follows. Section 2 gives the related work. The motivation of this paper is illustrated in Section 3. Section 4 presents the details of the proposed TOA. In Section 5, we analyze the computational complexity. Section 6 discusses the experimental results of TOA in comparison with the competing algorithms. Section 7 comprehensively tests the stability, robustness, and convergence curves of TOA and competitors on constrained engineering problems. Section 8 concludes the paper.

2. Related Work

In recent years, numerous metaheuristic algorithms have been proposed by researchers in the field. Based on the nature of their underlying inspiration, these algorithms can generally be categorized into four types: bio-inspired, human or socio-inspired, evolution-based, and physics-based metaheuristics.

2.1. Bio-inspired Metaheuristics

Bio-inspired metaheuristics emulate collective intelligence observed in animal swarms, insect colonies, avian flocks, and microbial systems. Foundational work includes Particle Swarm Optimization (PSO)[5], which formalized avian foraging dynamics into a mathematical model of collaborative search through position updating mechanisms. Subsequently, Yang et al. proposed several biologically inspired algorithms, such as Cuckoo Search (CS)[19], which mimics brood parasitism and L´evy flight behavior of cuckoos; Firefly Algorithm (FA)[20], based on bioluminescent attraction; Bat Algorithm (BA)[21], which simulates bat echolocation and hunting strategies; and Flower Pollination Algorithm (FPA)[22], modeled on the process of flower pollination. These works greatly enriched the ecosystem of nature-inspired optimization models.

Mirjalili's research team significantly advanced bio-inspired metaheuristics by exploring predator-prey dynamics and social behaviors, introducing the Grey Wolf Optimizer (GWO)[23], Ant Lion Optimizer (ALO)[24], Mothflame Optimization (MFO)[25], Dragonfly Algorithm (DA)[26], and Whale Optimization Algorithm (WOA)[6], each capturing aspects of collective hunting, prey pursuit, and population cooperation. Additional contributions include Marine Predators Algorithm (MPA)[27], simulating ecological niche transitions in marine environments; African Vultures Optimization Algorithm (AVOA)[28], inspired by vulture foraging behaviors; Sparrow Search Algorithm (SSA)[29], modeled after the early-warning mechanisms of sparrow flocks; Harris Hawk Optimization (HHO)[7], based on Harris hawks' cooperative hunting strategies; Dung Beetle Optimizer (DBO)[8], imitating dung beetles' rolling and burying behaviors; and more recently, Snake Optimizer (SO)[30] and Crested Porcupine Optimizer (CPO)[31], drawing from reptilian locomotion and porcupine defensive behavior. Other notable developments include the Black-winged Kite Algorithm (BKA)[12], proposed in recent years.

2.2. Human or Socio-inspired Metaheuristics

These algorithms derive inspiration from human individual and group behaviors in learning, decision-making, cooperation, social interaction, and cognition. They abstract corresponding optimization strategies and information update mechanisms to form highly adaptive search models. Some algorithms are directly inspired by educational contexts, such as Teaching–Learning-Based Optimization (TLBO)[32], Student Psychology Based Optimization (SPBO)[33], and Preschool Education Optimization Algorithm (PEOA)[34], which model population evolution based on the learning process of students under various teaching influences. Others are based on everyday human activities, including Chef-Based Optimization Algorithm (CBOA) [35], Football Team Training Algorithm (FTTA) [36], and Archer Algorithm (AA) [37].

Socioeconomic and political behaviors also serve as a rich source of inspiration. Examples include Election-Based Optimization Algorithm (EBOA)[38], which models collective decision-making through voting mechanisms; Enterprise Development Algorithm (ED)[1], inspired by key elements of business growth; City Council Evolution (CCE)[39], Political Optimizer (PO)[40], and Information Acquisition Optimizer (IAO)[41], all of which model organizational decision-making and social dynamics.

2.3. Evolution-based Metaheuristics

Grounded in the principles of natural evolution, these algorithms simulate biological processes such as mutation, selection, and recombination to solve complex optimization problems. The core concept draws from Darwin's theory of natural selection, mapping the principle of "survival of the fittest" onto iterative search in solution space. A seminal example is the Genetic Algorithm (GA)[42], proposed by John H. Holland, which systematically introduced concepts such as chromosomes, genes, crossover, and mutation into computational models. Subsequent variants and extensions, such as Differential Evolution (DE)[43], Evolution Strategy (ES)[44], Evolutionary Programming (EP)[45], and Memetic Algorithm (MA)[46], have further enriched the evolutionary computation paradigm by enhancing parameter control, local search capabilities, and adaptive mutation.

Building on this foundation, newer algorithms integrate evolutionary concepts with ecological dynamics to create more expressive optimization frameworks. Examples include Biogeography-based Optimization (BBO)[47], modeled on species migration and geographic isolation; Forest

Optimization Algorithm (FOA)[48], simulating interspecies competition in forest ecosystems; and Tree-Seed Algorithm (TSA)[49], inspired by seed dispersion strategies. Other notable developments include Monkey King Evolution (MKE)[50], Evolutionary Mating Algorithm (EMA)[51], Stochastic Fractal Search (SFS)[52], and GTBO[53], which mimics beetles' mating and defensive behaviors. These algorithms, with evolution at their core, continuously enhance adaptability and efficiency in high-dimensional, multimodal, and dynamic optimization scenarios by integrating animal behaviors and biological features.

2.4. Physics-based Metaheuristics

Inspired by physical phenomena and fundamental laws in nature, this class of algorithms models processes such as energy diffusion, gravitational attraction, heat conduction, and particle motion, abstracting them into search behaviors within the solution space. Among the early examples, Simulated Annealing (SA)[54] remains one of the most influential physicalinspired algorithms. It borrows from the annealing process of solid materials, where temperature control and energy minimization prevent premature convergence to local optima.

Later developments extended the application of physical principles into optimization, resulting in methods such as Gravitational Search Algorithm (GSA)[55], Multiverse Optimizer (MVO)[56], and Electromagnetism-like Algorithm (EM), which simulate gravitational interactions between celestial bodies, cosmic expansion and contraction, and interactions among charged particles to enhance global search capabilities. Additional algorithms further incorporate concepts from optics, electromagnetism, and thermodynamics, for example, Young's Double-slit Experiment Optimizer (YDSE)[57], which employs wave interference patterns to guide multi-path exploration; Lightning Search Algorithm (LSA)[58], modeling the branching and energy discharge of lightning; and Water Cycle Algorithm (WCA)[59], simulating the dynamic hydrological cycle among rivers, rainfall, and oceans for solution updates.

More recently, numerous optimization models have emerged based on gas diffusion, thermal exchange, and electric field induction, including Equilibrium Optimizer (EO)[60], Thermal Exchange Optimization (TEO)[61], Henry Gas Solubility Optimization (HGSO)[62], Mine Blast Algorithm (MBA)[63], and mass-balance models based on conservation principles within control volumes.

3. Motivation

Tuckman's Stages of Group Development [64], proposed by American psychologist Bruce Tuckman in 1965, is a widely recognized framework for describing the developmental stages of a team as it evolves from formation to dissolution. Initially composed of four stages—Forming, Storming, Norming, and Performing—the model was later extended in 1977, in collaboration with Jensen, to include a fifth stage: Adjourning. This addition aimed to more comprehensively reflect the disbanding process of teams after completing a task or project.

Emerging during the mid-20th century—a period marked by growing academic interest in organizational behavior and management psychology—the model is grounded in empirical observation and case analysis of team dynamics. It posits that effective team performance is not instantaneous but rather the outcome of a progressive and often nonlinear transition through multiple psychological and operational stages.

The Tuckman model has found widespread application across various domains, including business management, project coordination, education, collaborative research, and sports organizations. By correctly identifying the stage at which a team currently resides, leaders and managers can deploy targeted communication, coordination, and motivation strategies to mitigate internal conflicts and enhance collaborative efficiency.

It is important to note that while the stages are theoretically sequential, real-world teams may experience regressions, stagnation, or even skip certain stages depending on the members' prior working relationships. Thus, the model should be applied with flexibility and contextual awareness, particularly in complex, adaptive systems such as algorithmic frameworks that mimic collaborative behaviors. The five stages of the Tuckman model are briefly summarized as follows:

- **Forming**: Team members are newly assembled and generally unfamiliar with each other. This stage is characterized by exploratory behavior, dependency on initial guidance, and unclear role distribution. The focus lies in establishing structure and initial interpersonal connections.
- **Storming**: As individual personalities emerge, conflicts may arise related to role allocation, task delegation, and decision-making authority. This stage often challenges the team's internal cohesion and requires strong conflict resolution and communication strategies.
- **Norming**: The team gradually develops shared norms, mutual trust, and a sense of collective identity. Responsibilities become clearer, and coordination improves as team members align around common objectives and operating principles.
- **Performing**: With roles and structures firmly in place, the team achieves high efficiency and self-regulation. Emphasis shifts toward task execution and problem-solving, with team members demonstrating strong collaboration and productivity.
- **Adjourning**: Upon task completion or project closure, the team moves into the dissolution stage. This stage typically involves outcome evaluation, reflection on group performance, and emotional detachment among members.

In summary, the Tuckman model offers a systematic lens for analyzing team behavior and coordination patterns. When incorporated into algorithmic design—particularly in swarm intelligence or metaheuristic frameworks—the model can provide theoretical grounding for the design of dynamic, cooperative search mechanisms that simulate human team evolution and enhance both global exploration and local exploitation capabilities.

4. Tuckman Optimization Algorithm

In this Section, we gives the details of the proposed Tuckman Optimization Algorithm (TOA). The pseudocode and flowchart of TOA are presented in Algorithm 1 and Fig.5, respectively.

4.1. Initialization process

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,d} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & X_{n,d} \end{bmatrix} \quad (1)$$

$$X_{i,j} = LB_j + \text{rand}[0,1] \times \left(UB_j - LB_j\right), \quad \begin{cases} i = 1,2,\ldots,n \\ j = 1,2,\ldots,d \end{cases} \quad (2)$$

Where, n denotes the number of potential solutions, d represents the dimensionality of the given problem, Xi,j indicates the value of the ith individual in the jth dimension, and rand[0,1] is a uniformly distributed random number in the range [0,1].

4.2. Forming stage

During the Forming stage of Tuckman's Group Development Model, members come together for the first time and are generally unfamiliar with one another. The primary objective at this stage is for individuals to gradually understand task goals, clarify role allocations, and establish their functional positioning within the team. According to the Jigsaw Method proposed by Aronson[65], teams consisting of 3 to 5 members are more likely to achieve knowledge complementarity and effective collaboration. A team that is too small may lack sufficient knowledge diversity, while a team that is too large may encounter information redundancy and communication barriers. Therefore, in the proposed model, individuals reference the Jigsaw mechanism and interact with 3 to 5 randomly selected peers to learn capabilities aligned with team objectives, thereby improving individual adaptability within the group.

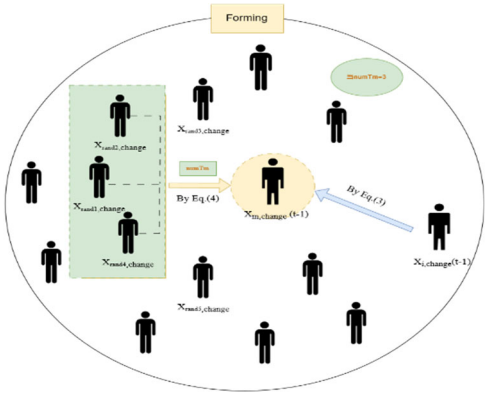


Fig. 1 Details of the forming stage

To model the cognitive evolution of individual capabilities during this stage, we introduce the following learning formulation that simulates information flow and capability alignment among team members:

$$X_i^{change}(t) = X_i^{change}(t-1) + KL \cdot \left(X_i^{change}(t-1) - X_m^{change}(t-1)\right) \quad (3)$$

where Xichange(t) denotes the ability value of individual i at iteration t in the selected dimension change; the term $\left(X_i\right.$ change$(t-1) - $ Xmchange$\left.^{(t-1)}\right)$ reflects the deviation from the peer group's average capability, thus representing the individual's adjustment during the learning process. Xmchange(t−1) refers to the average capability of a set of randomly selected team members in the previous iteration, and is calculated by:

$$X_m^{\text{change}}(t-1) = \frac{1}{\text{numTm}} \sum_{k=1}^{\text{numTm}} X_{\text{rand}_k}^{\text{change}}(t-1) \quad (4)$$

In the above, Xrandchangek denotes the ability value of the kth randomly selected member in the chosen dimension change during the previous iteration. The number of sampled members, numTm, is randomly chosen from the set {3,4,5} with equal probability, as defined below:

$$\text{numTm} = \begin{cases} 3, & \text{with probability } \frac{1}{3}, \\ 4, & \text{with probability } \frac{1}{3}, \\ 5, & \text{with probability } \frac{1}{3} \end{cases} \quad (5)$$

To reflect the adaptability and variability in individual learning efficiency, a dynamic adjustment factor KL is introduced and defined as follows:

$$KL = sign(r_1) \cdot sin\left(\frac{\pi}{2} \cdot \sqrt{r_2 \cdot TP}\right) \quad (6)$$

where r1 is a standard normally distributed random variable ~ N(0,1), whose sign determines the direction of learning—either converging toward or diverging from the peer average. r2 is a uniformly distributed random variable over the interval (0,1), introducing stochastic perturbation. TP denotes the team preparation progress of the individual, a parameter that quantifies the extent of readiness during the formation stage (explicitly defined in Equation.18).

Moreover, the capability dimension change $\in$ {1,2,...,d} is selected uniformly at random in each iteration to ensure a balanced and exploratory adaptation across all skill dimensions in the early stages of team development.

4.3. Storming stage

During the storming stage of Tuckman's Group Development Model, individual members begin to exhibit distinct personalities and independent thinking. This often leads to friction and conflicts, particularly regarding role assignments and task distributions. This stage is a critical transition period as the team evolves from initial formation to effective collaboration, placing significant demands on conflict resolution and communication skills.

To model the dynamic evolution of individual competencies and interpersonal conflicts during this stage, the following mathematical formulation is proposed:

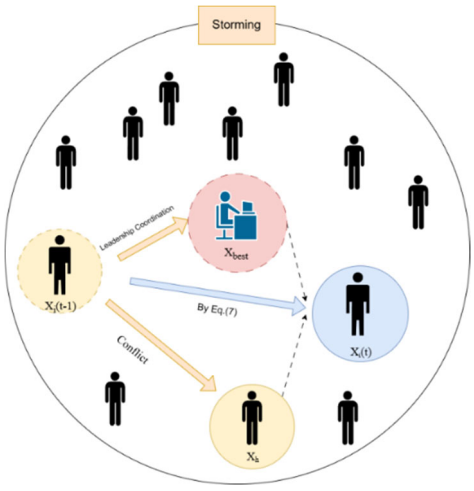


Fig. 2 Details of the storming stage

$$X_i(t) = X_i(t-1) + r_3 \cdot (X_{\text{best}} - X_i(t-1)) + r_4 \cdot (X_i(t-1) - X_h) \cdot \left(1 - \frac{g}{\text{MAXF}}\right) \quad (7)$$

where r3,r4 $\in$ R1×D are random vectors of dimension D, with each element uniformly distributed over the interval (0,1); Xbest denotes the bestperforming individual in the

current iteration, representing the role of a supervisor or guiding leader; Xh is a randomly selected peer individual from the team; D refers to the dimensionality of the problem, that is the number of capabilities required by the team. Variables g and MAXF represent the current iteration number and the predefined maximum number of iterations, respectively.

This model consists of two principal components. The first term, (Xbest − Xi), captures the process in which an individual learns from and aligns with the supervisor, reflecting ability enhancement under leadership and structured communication. The second term, (Xi −Xh), simulates the emergence of conflicts or tensions due to discrepancies in cognition or coordination among members. However, such conflicts can also stimulate learning and insight. The modulation factor $1-\frac{g}{\mathrm{MAXF}}$ gradually attenuates the influence of interpersonal conflict over time, representing the natural reduction of friction as team familiarity and cohesion increase. This design ensures a dynamic balance between exploration (divergence through interaction) and exploitation (convergence via leadership) throughout the team development process.

4.4. Norming stage

As the Tuckman's Group Development Model enters the norming stage, internal cooperation mechanisms begin to solidify. Team members exhibit enhanced awareness of collaboration, and mutual trust becomes more established. During this stage, the team tends to converge on shared goals and progressively align their behaviors, resulting in more efficient coordination and integration of resources. To model the process through which team members gradually

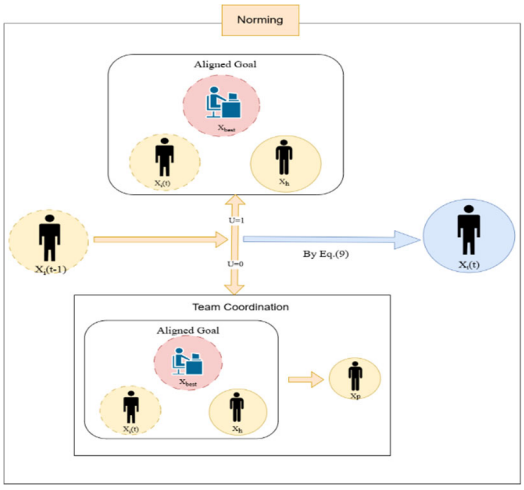


Fig. 3 Details of the norming stage

Confirm common objectives and enhance collective capabilities, the following mathematical formulation is introduced:

$$X_i(t) = \frac{X_h + X_i(t-1) + X_{\mathrm{best}}}{3} \tag{8}$$

This equation represents the updated state of team member Xi(t) at iteration t, computed as the average of their own previous state Xi(t − 1), the current team leader (i.e., the historical best solution) Xbest, and another randomly selected member Xh. This averaging process simulates the increasing consensus among team members during the norming stage.

To further capture individual adjustments during this convergence process, the following update mechanism is introduced:

$$X_i(t) = X_i(t) \cdot U_1 + \Big(X_i(t) + CI \cdot \big(X_i(t) - X_p\big)\Big) \cdot (1 - U_1) \tag{9}$$

where, the adaptive factor CI is used to describe the member's ability to reconcile personal differences and align with the group, defined as follows:

$$CI = \frac{1}{2} \cdot sign(r_5) \cdot TP \cdot sin\left(\frac{\pi}{2} \cdot \sqrt{\frac{g}{\mathrm{MAXF}}}\right) \tag{10}$$

In this expression, r5 is a standard normally distributed random variable, where its sign determines the direction of adjustment (i.e., alignment or divergence) during the coordination process. TP represents the current team preparation progress, while g and MAXF denote the current and maximum number of iterations, respectively.

Furthermore, a control factor U1 determines whether the member's behavior follows the current team consensus or adapts to alternative views. It is defined as:

$$U_1 = \begin{cases} 1 & \text{if } r_6 \cdot \frac{g}{\mathrm{MAXF}} > r_7 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Where r6 and r7 are uniformly distributed random variables in the interval (0,1).

When U1 = 1, it implies that the team has reached a consensus on a specific issue, and the member fully adopts the collective behavior. Conversely, when U1 = 0, it reflects the presence of divergent opinions within the team. In this case, the individual references another randomly selected member Xp and adjusts their behavior by applying the adaptive factor CI to bridge the difference, modeling the team dynamics sequence: disagreement-coordination convergence.

As the iteration count g increases, the probability of U1 being equal to 1 also increases, indicating that the likelihood of consensus rises over time. This mirrors real-world team development dynamics, where divergence decreases as collaboration deepens. From an algorithmic perspective, this mechanism introduces a probabilistic choice between direct inheritance and corrective adjustment, effectively balancing global exploration and local exploitation capabilities.

4.5. Team control

Based on the first three stages of the Tuckman model, Forming, Storming, and Norming, this study proposes a stage-regulated optimization algorithm referred to as the Tuckman Algorithm. In the initial forming stage, team members gradually engage in mutual understanding and knowledge acquisition through interaction. However, not all individuals may meet the comprehensive capability requirements of the team. To address this, the algorithm ranks members based on their performance (i.e., fitness) and replaces the lower-performing individuals with an equal number of new candidates.

Throughout the continuous selection and replacement process, the team composition gradually stabilizes as the optimization progresses. Once member replacement ceases, the algorithm skips the forming stage, which emphasizes mutual learning and familiarity. According to the Tuckman model, if team members have previously collaborated, the team-building process may bypass certain stages. Therefore, in the absence of further personnel changes and with sufficient mutual understanding, the algorithm focuses on iterative refinement within the storming and norming stages. As preparation time increases, the team evolves toward a stable and well-coordinated norming stage.

To simulate this staged evolution of team development, the following mathematical model is constructed to control the

progression of TOA:

$$\text{std} = \begin{cases} 3 \cdot U_2 + 2 \cdot (1 - U_2) & \text{if } LNP = 0 \\ 3 \cdot ct & \text{otherwise} \end{cases} \quad (12)$$

Where LNP denotes whether any new members have been introduced in the current iteration, with its value determined by Equation (to be defined).

When member replacement is still occurring, the current team stage is governed by an adaptive factor ct, defined as follows:

$$ct = \frac{r_8 \cdot g}{\text{MAXF}} \quad (13)$$

In this expression, r8 is a uniformly distributed random variable within the interval (0,1), g represents the current iteration number, and MAXF is the maximum number of iterations. This formulation models the natural transition of the team from the forming stage toward storming and eventually norming, as preparation progresses.

To further enhance the flexibility of stage switching, a control factor U2 is introduced, defined as:

$$U_2 = \begin{cases} 1 & \text{if } \frac{g}{\text{MAXF}} > r_9 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

Where r9 is another uniformly distributed random variable within the interval (0,1). The value of U2 is determined by comparing the current iteration progress $\frac{g}{\text{MAXF}}$ with r9, which governs the switching of algorithmic stages. As the iteration count increases, the probability of entering the norming stage also increases, driving the convergence of the algorithm. This mechanism reflects the real-world scenario in which a team gradually stabilizes and matures over time.

4.6. Team structure reconfiguration mechanism

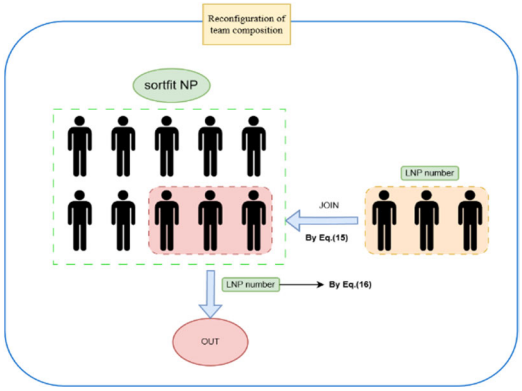


Fig. 4 Details of the team structure reconfiguration mechanism

As described earlier, during the initial stage of team formation, the team composition remains unstable. To identify members (i.e., candidate solutions) that better meet the team's requirements, an individual replacement mechanism is introduced at the end of each team stage. When the condition r10 > r11 is satisfied (where r10,r11 ~ U(0,1)), a certain number of inferior members, determined by the value of LNP, are replaced with the same number of newly generated candidates. The generation of these new individuals is governed by the following model:

$$X_{\text{sort},j} = LB_j + \text{rand}_{[0,1]} \times (UB_j - LB_j), \quad \begin{cases} \text{sort} = 1,2,\dots,LNP \\ j = 1,2,\dots,d \end{cases} \quad (15)$$

where sort represents the index of the individuals to be replaced, LNP is the number of individuals selected for replacement, d denotes the dimensionality of the problem, and LBj and UBj are the lower and upper bounds for the jth dimension, respectively.

Throughout the team preparation process, inferior individuals identified by the LNP parameter are continuously replaced until LNP = 0, indicating that all current members meet the team's capability requirements and the team has reached a stable configuration.

The number of members to be replaced, LNP, is determined by:

$$LNP = NP - PIM \quad (16)$$

where NP is the total number of team members and PIM is the number of individuals currently meeting the required capability threshold.

To simulate the gradual increase in suitable candidates under constrained resources and time, a logistic growth model [66] is employed to model the progression of PIM. The update rule is defined as:

$$PIM(t+1) = \left\lceil k \cdot \frac{NP}{1 + \frac{NP - PIM(t)}{PIM(t)} \cdot exp(-TP \cdot g)} \right\rceil \quad (17)$$

Algorithm 1 Tuckman Optimization Algorithm (TOA)

Input: *NP*, *MAXF*, *LB*, *UB*, *D*, objective function *f*(*x*)
Output: Best solution X_{best}, best fitness value *Best score*
Initialize:
Initialize population x_i, i = 1,...,*NP* using Eq. 2 within [*LB*,*UB*]
Evaluate fitness $Obj_i = f(x_i)$
Identify global best X_{best} and *GlobalMin*
Set iteration counter g = 1
while g < *MAXF* **do**
Calculate stage factor k using Eq. 20
Calculate *PIM* using Eq. 17
Calculate *TP* using Eq. 18
Determine the stage index *std* using Eq. 12
for i = 1 to *NP* **do**
  if *std* = 1 **then** ▷ Forming
    Update x_i using Eq. 3
  else if *std* = 2 **then** ▷ Storming
    Update x_i using Eq. 7
  else if *std* = 3 **then** ▷ Norming
    Update x_i using Eq. 9
  end if
  Update X_{best} and *GlobalMin* if necessary
end for
if rand > rand **then**
  Re-initialize the worst *LNP* individuals using Eq. 16
  end if
Record *Convergence curve*(g)
$g \leftarrow g + 1$
end while
return X_{best}, *Best score*

In this expression, PIM(t) denotes the number of qualified members at iteration t, g is the current iteration count, and TP represents the team's preparation progress, calculated as:

$$TP = \frac{PIM}{NP} \quad (18)$$

Drawing inspiration from the Pareto principle [67] (the 80/20 rule), which suggests that approximately 80% of outcomes stem from 20% of contributors, we assume that 20% of the initial members are elite individuals. This initial configuration is defined as:

$$PIM = 0.2 \cdot NP \quad (19)$$

To control the growth rate in the logistic model, a dynamic adjustment factor k is introduced to reflect the increasing suitability of members over time:

$$k = \sqrt{1 - \left(\frac{MAXF - g}{MAXF}\right)^2} \tag{20}$$

By integrating the above member filtering and stage control mechanism, the team evolves from an unstable exploratory state to a more stable and capable structure. As LNP decreases and PIM increases, the algorithm transitions from global exploration to local exploitation, achieving a dynamic balance between exploration and exploitation in the team formation process.

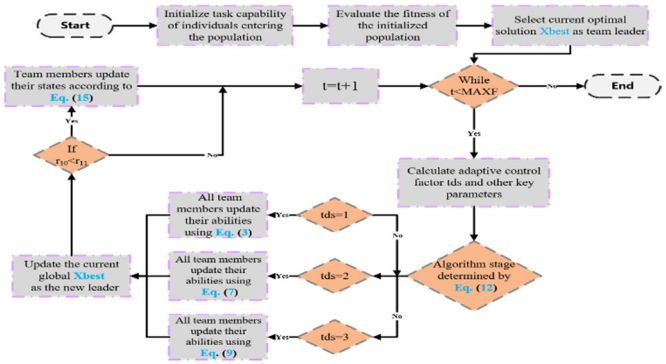


Fig. 5 flowchart of TOA

5. Computational Complexity Analysis

5.1. Time Complexity

The time complexity of TOA is mainly affected by the population size NP, problem dimension D, and maximum iteration number T. The computational complexity of the initialization phase is O(NP ×D); during each iteration, the individual update, boundary handling, and fitness evaluation processes in the Forming, Storming, and Norming phases collectively have a complexity on the order of O(NP×D). Since the entire iterative process repeats T times, the total runtime complexity of the algorithm can be expressed as: O(NP×(3T×D+1)).

5.2. Space Complexity

During the entire optimization process, the primary space overhead of the TOA comes from storing the population and the global best individual, which requires O(NP ×D) memory. Other variables, such as intermediate candidate solutions, are temporary and exist only during the update procedures, contributing only a constant-level overhead that does not affect the overall space scale and can thus be neglected. In summary, the space complexity of TOA is O(NP × D).

6. Experimental Results and Analysis

In this section, the optimization characteristics of the proposed TOA are first investigated through qualitative analysis on the IEEE CEC2017 benchmark suite. The analysis utilizes search space surface plots of representative benchmark functions, two-dimensional search trajectory plots, population mean fitness curves, single-dimensional perturbation trajectories, and overall convergence curves to provide an in-depth understanding of the algorithm's search behavior and evolutionary dynamics. Subsequently, quantitative performance evaluation is conducted on the IEEE CEC2017, CEC2020, and CEC2022 benchmark suites. The optimization performance of TOA is comprehensively assessed through comparative experiments against fourteen state-of-the-art metaheuristic algorithms. To ensure fairness, the key parameters of all comparative algorithms are configured according to their respective best-practice settings, as shown in Table 1.The evaluation metrics include mean fitness value, standard deviation (variance), Friedman ranking test, Wilcoxon signedrank tests at a significance level of 0.05, boxplot analysis, and convergence curve comparison, providing a thorough assessment of the algorithm's solution accuracy, stability, and convergence characteristics across a variety of complex optimization problems.

All experiments are conducted under a consistent computational environment to ensure fairness and reproducibility. The experimental platform consisted of MATLAB R2021a running on a Windows 10 (64-bit) operating system, equipped with an Intel(R) Core(TM) i7-9700 CPU at 3 GHz and 64 GB of RAM.

Tab. 1 Comparison of algorithmic parameters

Algorith	Parameters	Value	Year
PSO	w,c1andc2	0.8,0.5	1995
WOA	a	[2 0]	2016
HHO	Parameterless		2019
DBO	kandλ,b,s	0.1,0.3,0.5	2022
GWCA	P,Q,T,M,Cmax,Cmin	9,6,8.3,3,exp(3),exp(2)	2023
WO	P,beta	0.4,1.5	2024
AO	Parameterless		2024
TTAO	Parameterless		2024
BKA	p	0.9	2024
BSLO	m,a,b,t1,t2	0.8,0.97,0.001,20,20	2024
EEFO	b	1.5	2024
FATA	arf	0.2	2024
CFOA	Parameterless		2024
LSHADE SPACMA	Memorysize	5	2017
	Lrate,Pbestrate,Arcrate	0.8,0.11,1.4	
	Firstclasspercentage	0.5	

6.1. Performance analysis on IEEE CEC benchmark suite

As illustrated in Fig. 6, the visualizations depict the performance of the TOA on the CEC2017 benchmark suite under a configuration of 1000 iterations, a population size of 30, a problem dimension of 10, and 30 independent experimental runs. The visual outputs include the benchmark function surface plots, search trajectory plots in the first two dimensions, the population's average fitness curve, single-dimension trajectory analysis, and the overall convergence curve.

The function surface plots offer an intuitive understanding of the search landscapes associated with each benchmark function, effectively highlighting the differences among unimodal, multimodal, hybrid, and composition functions. This provides a clear visual reference for assessing the complexity and optimization difficulty of each category.

In the historical trajectory plots, red points indicate the best individual found by TOA at each iteration, while black points represent other candidate solutions. During the early

iterations, the search trajectories are widely distributed across the solution space, reflecting the strong global exploration capability of TOA. As the iterations progress, the distribution of solutions becomes increasingly focused around high-quality regions, demonstrating the algorithm's robust convergence behavior. Overall, TOA exhibits an effective transition from global exploration to local exploitation, combining wide search coverage with refined convergence.

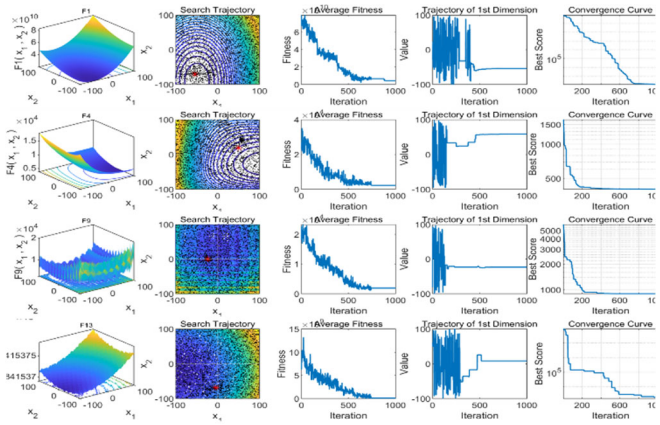


Fig. 6 Search space, search history, average fitness, Trajectory of 1st Dimension and convergence curve of TOA

The average fitness curve shows a downward trend in population fitness values as the number of iterations increases. Large fluctuations during the early exploration stage reflect high strategy-induced diversity, while the steady descent in later stages indicates the algorithm's strong convergence stability and local refinement ability.

The single-dimension trajectory plot further delineates the transition between exploration and exploitation stages. In the early stages, large-scale perturbations in variable values allow for broad exploration of the search space. As the algorithm enters the exploitation stage, these perturbations decrease in magnitude and become more localized, indicating focused local development. This demonstrates the adaptability of TOA in adjusting its search behavior across different evolutionary stages.

Finally, the convergence curve reveals that TOA rapidly escapes from local optima in the early iterations and achieves a smooth and stable convergence path. The absence of oscillations or stagnation at coarse approximations underscores the algorithm's ability to maintain solution quality and perform fine-grained local search during the later stages.

6.2. Performance evaluation on the CEC2017 benchmark suite

The CEC2017 benchmark suite is adopted to assess the robustness and convergence performance of TOA. To ensure the testbed's validity, the uncertain F2 function is excluded following official guidelines. The benchmark comprises four categories: unimodal functions (F1, F3), multimodal functions (F4–F10), hybrid functions (F11–F20), and composition functions (F21–F30), with all functions defined within the domain. All algorithms are tested under consistent settings: 10-dimensional problems, population size of 30, a maximum of 10,000 iterations, and 30 independent runs per function.

As illustrated in Fig.10a, the average rank curve shows that TOA achieves outstanding performance across most functions. Table 2andTable 3 reports

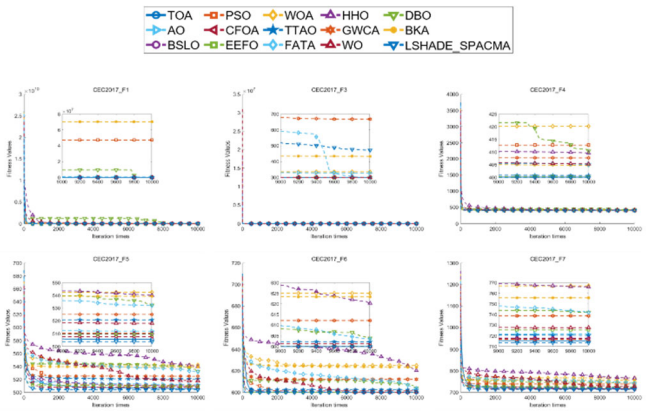


Fig. 7 Convergence performance of all the algorithms on representative CEC2017 benchmark suite

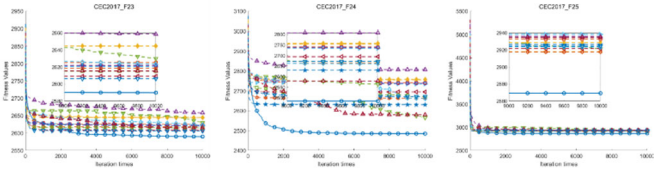


Fig. 8 Convergence performance of all the algorithms on representative CEC2017 benchmark suite

the mean, standard deviation, and average rank for each algorithm on all 29 functions (excluding F2). The best-performing results are highlighted in bold. TOA ranked within the top three on 26 out of 29 functions, with 18 functions achieving first place. Its overall average rank was 1.72, outperforming all peer algorithms, including the strongest baseline, LSHADE SPACMA, the official winner of CEC2017.

While TOA showed slightly lower performance on F12, F13, and F18—ranking fifth, fourth, and fourth respectively—it dominated all other hybrid functions, underscoring its capacity to handle nontrivial multi-source landscapes. These slight deficiencies are attributed to the particularly deceptive or irregular characteristics of a few hybrid functions

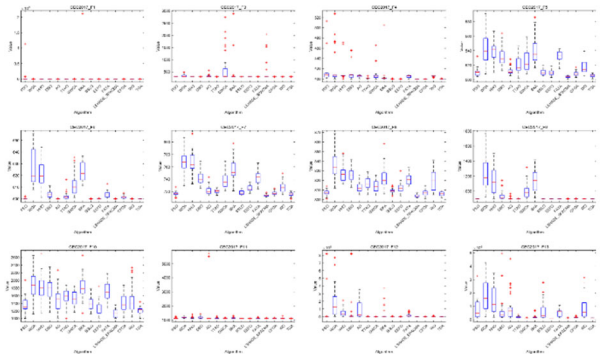
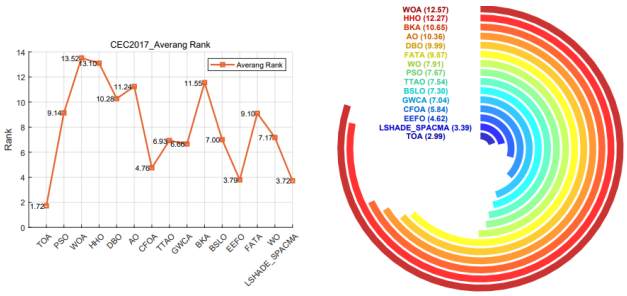


Fig. 9 CEC2017 Boxplot of TOA

In the convergence curves shown in Fig.7and Fig.8, it can be clearly observed that TOA achieves a leading position in terms of final convergence performance on the majority of CEC2017 benchmark suite, while also demonstrating superior convergence speed.

The boxplots in Fig.9 further illustrate the algorithm's stability. The compact or absent boxes in most functions, with few outliers, indicate a high degree of robustness and convergence reliability. Notably, in F23, F24, F25, and



(a) Mean Rank Line Plot (b) Friedman Rank Circular Plot

Fig. 10 CEC2017 Ranking Visualization of TOA

F28, although a few outliers exist, they correspond to high-quality solutions, reflecting TOA's strong exploration capability near global optima.

To more intuitively illustrate the superior overall optimization performance of TOA, Fig. 10

The Wilcoxon significance test results indicate that TOA achieves statistically significant performance improvements over most of the compared algorithms. Compared to the widely used CEC2017 champion algorithm LSHADE SPACMA, TOA demonstrates superior stability and exploration capability on hybrid and composition functions.

6.3. Performance evaluation on the CEC2020 benchmark suite

The CEC2020 benchmark suite comprises ten functions—unimodal (F1), multimodal (F2–F4), hybrid (F5–F7), and composition (F8–F10)—designed to evaluate algorithmic convergence and stability on small- to medium-scale problems. All experimental settings were kept consistent with those used for CEC2017.

Compared to its performance on CEC2017, TOA exhibits even greater balance on the CEC2020 benchmark suite, achieving top-three rankings on all ten functions, including six first-place finishes and an overall average rank of 1.6, thereby outpacing every other competing method. These results demonstrate TOA's strong adaptability and robustness across a wide variety of function landscapes.

As shown in Fig. 12, TOA delivers exceptional stability on functions F1–F8: the boxplots are highly concentrated with minimal variance and virtually no outliers. Although more outliers appear on F9 and F10, these correspond to higher-quality solutions, underscoring the algorithm's powerful exploratory capabilities.

Table 7 summarizes the Friedman ranking results for all algorithms; Fig.13a displays the average ranking convergence curves; and Fig.13b presents the Friedman race plots. Together, these visualizations further confirm TOA's clear leadership in overall optimization performance.

In the Wilcoxon significance tests reported in Table 8, TOA shows statistically significant advantages on the majority of functions. Even when benchmarked against the strong CEC2017 champion LSHADE SPACMA, TOA matches or exceeds its performance on 80% of the CEC2020 functions, particularly excelling in the hybrid and composition categories, highlighting its effective generalization to diverse function structures.

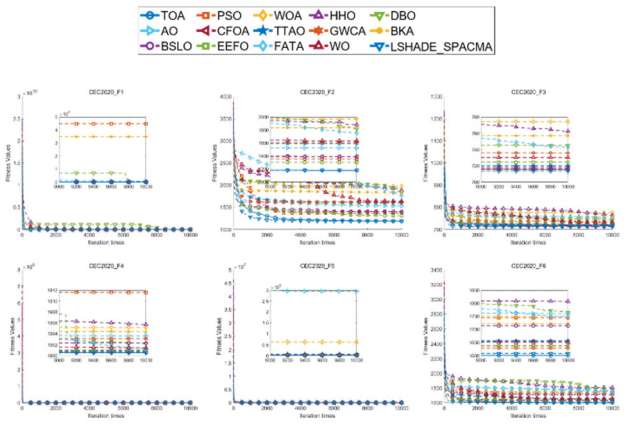


Fig. 11 Convergence curves of all the algorithms on the CEC2020 benchmark suite

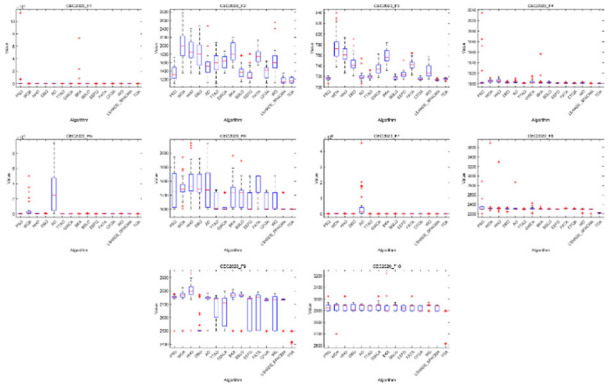
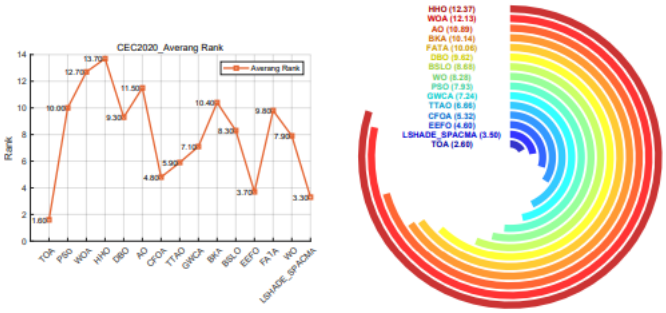


Fig. 12 CEC2020 Boxplot of TOA



(a) Mean Rank Line Plot (b) Friedman Rank Circular Plot

Fig. 13 CEC2020 Ranking Visualization of TOA

6.4. Performance evaluation on the CEC2022 benchmark suite

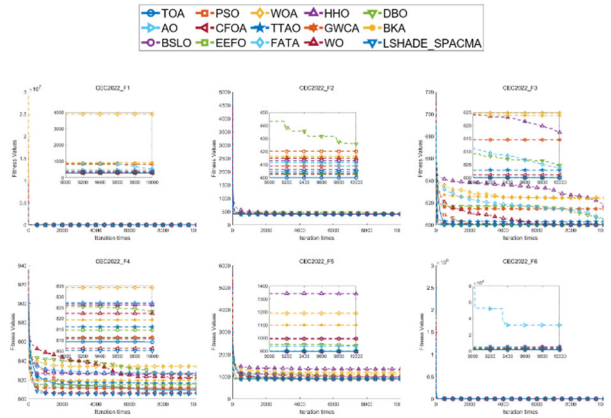


Fig. 14 Convergence Curves of all the algorithms on the CEC2022 benchmark suite.

The CEC2022 benchmark suite provided a more modern and challenging testbed, consisting of unimodal (F1), multimodal (F2–F5), hybrid (F6–F8), and composition (F9–F12) functions, all defined within. Consistent experimental settings were maintained.

all algorithms, including TOA, struggled to escape local optima due to the deceptive nature and convergence traps inherent to the function.

TOA displayed strong global search ability and adaptability in complex hybrid and composition functions. Nonetheless, it performed slightly less effectively on F6 and F11. These functions pose substantial structural complexity:

F6 integrates several rotated and shifted sub-functions, creating a highly nonlinear and interference-prone landscape that can mislead the optimizer. F11, as a composite function, features steep gradients and many local optima with strong attraction basins, which challenge the robustness and escape capability of any algorithm. The suboptimal performance of TOA on these functions may stem from limitations in its local search mechanism and sensitivity to weighted composite perturbations in high-dimensional spaces.

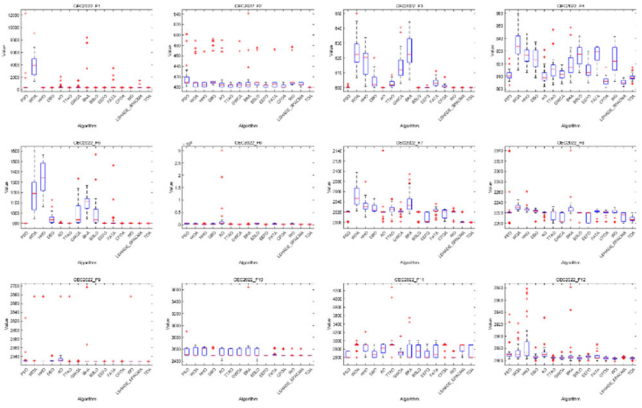
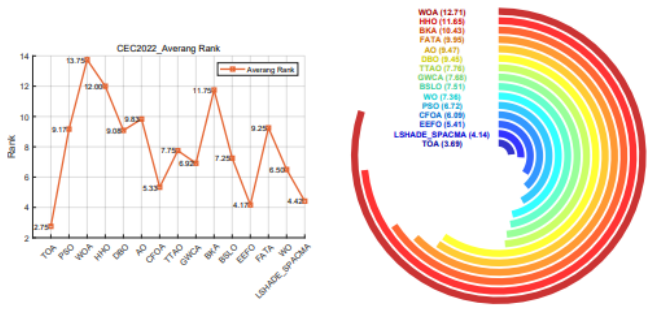


Fig. 15 CEC2022 Boxplot of TOA

Fig.15 illustrates that TOA produced short box lengths with few outliers for most functions in CEC2022, reflecting high stability. Even in more complex scenarios, the solution quality of TOA remained competitive.

The average ranking curve of the algorithms is illustrated in Fig.16a, while the Friedman Rank Circular Plot in Fig.16b offers an intuitive visualization of TOA's overall superior position. Furthermore, the Wilcoxon significance test results in Table 11 confirm that TOA demonstrates statistically significant advantages over the other compared algorithms. Even when competing against the strongest peer, EEFO, TOA still achieved better or comparable results on 66% of the benchmark functions.



(a) Average Rank Line Plot (b) Friedman Rank Circular Plot

Fig. 16 CEC2022 Ranking Visualization of TOA

In conclusion, TOA demonstrated consistent superiority across all three CEC benchmark suite. Its convergence precision, search stability, and adaptability to complex problem landscapes affirm its strength as a highperformance optimization algorithm.

7. Constrained engineering problems

To further evaluate the adaptability and effectiveness of the proposed TOA in practical engineering optimization scenarios, five representative and wellstudied engineering design problems are selected as benchmark cases. These include: Cantilever beam Design, Tubular column design.. These problems are widely encountered in structural engineering, mechanical design, and industrial systems, and are characterized by nonlinear objective functions, complex constraints, and strong inter-variable coupling—making them suitable for comprehensively testing the stability, robustness, and convergence behavior of metaheuristic algorithms.

In this experiment, TOA is compared against eight state-of-the-art metaheuristic algorithms from recent literature: AO, BKA, BSLO, DBO, FATA,

HHO, WO, and WOA. To ensure a fair and consistent comparison, all algorithms are configured with identical parameter settings, as detailed in Table 1. Each algorithm is executed independently 30 times on each benchmark problem under the same experimental conditions: a population size of 30 and a maximum of 1000 iterations.

For performance evaluation, three statistical indicators are reported: the best obtained solution, the mean objective value, and the standard deviation across 30 independent runs. The best-performing results for each problem are highlighted in bold. Additionally, to further analyze algorithmic performance dynamics, convergence curves are plotted for representative problems to visualize convergence rate, search behavior, and optimization stability throughout the iterations.

7.1. These experiments provide a comprehensive assessment of the global search capability and solution quality of the TOA across diverse and realistic engineering design landscapes.

7.2. Cantilever beam

The Cantilever beam structure consists of five hollow square cross-sectional elements. The objective of the optimization process is to minimize the overall weight of the beam while satisfying a vertical displacement constraint. The problem involves five decision variables, each corresponding to the side length of a structural element, and a nonlinear constraint related to the deflection of the beam.The structural diagram is illustrated in Fig. 17a. The mathematical formulas are as follows:

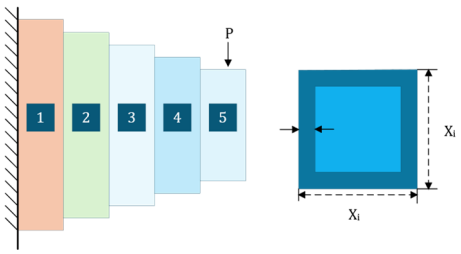


Fig.17 Schematic diagram of the cantilever beam

• **Decision Variables:**

$X = [x_1, x_2, x_3, x_4, x_5]$

• **Objective Function:**

$$f(X) = 0.0624 \times (x_1 + x_2 + x_3 + x_4 + x_5) \times L$$

• **Constraint:**

$$g_1(X) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leqslant 0$$

• **Bounds:**

$$0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100$$

As shown in Table 2. The performance comparison between TOA and eight widely recognized baseline algorithms (AO, BKA, BSLO, DBO, FATA, HHO, WO, and WOA) demonstrates the superiority of TOA across all key metrics. Specifically, TOA achieved the best results in terms of both the minimum and average objective values, while also obtaining the smallest standard deviation (i.e., 4.35151 × 10−6), indicating excellent consistency and robustness in its search behavior.

Based on these findings, it can be concluded that TOA is not only highly effective in solving the Cantilever beam optimization problem, but it also demonstrates clear advantages over alternative approaches in terms of convergence accuracy, stability, and global optimization capability.

Tab.2 Optimization results comparison of the algorithms on the cantilever beam design problem

Algorithm	x_1	x_2	x_3	x_4	x_5	Best Score	Mean Score	Std Dev	Rank
TOA	6.0156	5.3115	4.4922	3.4994	2.1549	1.33995737	1.33996219	4.35151E-06	1
WOA	6.0337	4.6923	4.7177	3.6841	2.8493	1.37137407	1.47446040	9.16548E-02	9
HHO	6.0982	5.2083	4.4588	3.5770	2.1413	1.34057938	1.34295900	1.85910E-03	6
DBO	6.0133	5.3143	4.4876	3.5045	2.1540	1.33995900	1.33999612	3.97672E-05	2
AO	6.0177	5.3247	4.4935	3.4788	2.1594	1.33998347	1.39295628	1.00719E-01	8
BKA	6.0191	5.3117	4.4958	3.5017	2.1455	1.33996157	1.35608837	5.68894E-02	7
BSLO	6.0166	5.3176	4.4826	3.4968	2.1602	1.33996592	1.34011884	2.06025E-04	4
FATA	6.0205	5.3111	4.4821	3.5033	2.1569	1.33997030	1.34008324	6.39609E-05	3
WO	6.0612	5.3062	4.4499	3.5048	2.1530	1.34005444	1.34061697	4.49225E-04	5

7.3. Tubular column

The Tubular column design problem is a classical real-world engineering optimization task. The primary objective is to minimize the overall design cost of the column subject to six engineering constraints. The problem involves two decision variables: d and T, representing the column diameter and wall thickness, respectively. The structural diagram is illustrated in Fig. 18.

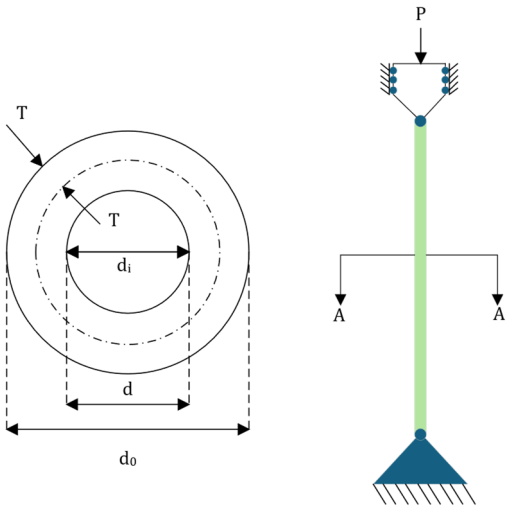


Fig.18 Schematic diagram of the tubular column

The mathematical formulas is as follows: **Decision Variables:**

$X = [x_1, x_2] = [d, T]$

Objective Function:

$$f(X) = 9.8x_1x_2 + 2x_1$$

Constraints:

$$\begin{cases} g_1(X) = \dfrac{P}{\pi x_1 x_2 \sigma_y} - 1 \leq 0 \\ g_2(X) = \dfrac{8PL^2}{\pi^3 E x_1 x_2 (x_1^2 + x_2^2)} - 1 \leq 0 \\ g_3(X) = \dfrac{2}{x_1} - 1 \leq 0 \\ g_4(X) = \dfrac{x_1}{14} - 1 \leq 0 \\ g_5(X) = \dfrac{0.2}{x_2} - 1 \leq 0 \\ g_6(X) = \dfrac{x_2}{0.8} - 1 \leq 0 \end{cases}$$

Search Space:

$2 \leq x_1 \leq 14, \quad 0.2 \leq x_2 < 0.8$

As shown in Table 3, TOA achieves the same best historical value as algorithms such as DBO, BKA, and BSLO, all reaching the global optimum. In terms of average performance, TOA performs comparably to DBO and BKA, while slightly lagging behind DBO in terms of standard deviation, which reaches 1.29560 × 10−12. Nevertheless, TOA still outperforms all other competing algorithms in this regard.

Taken together, these results validate the effectiveness and reliability of TOA in solving the Tubular column design problem, showing clear advantages over most other leading algorithms. Monstrating comprehensive superiority in both convergence reliability and optimization performance.

Tab.3. Optimization results comparison of the algorithms on the tubular column design problem

Algorithm	x_1	x_2	Best Score	Mean Score	Std Dev	Rank
TOA	5.452180737	0.291626429	26.48636147	26.48636147	1.29560E-12	2
WOA	5.452647921	0.291601443	26.48729585	26.69451244	1.67239E-01	8
HHO	5.452571098	0.291605551	26.48714219	26.53100054	4.36294E-02	7
DBO	5.452180737	0.291626429	26.48636147	26.48636147	6.80000E-15	1
AO	5.456118691	0.292040798	26.52764811	28.45904051	2.19216E+00	9
BKA	5.452180737	0.291626429	26.48636147	26.48636431	1.50019E-05	4
BSLO	5.452180737	0.291626429	26.48636147	26.48636147	2.08503E-09	3
FATA	5.452059913	0.291649632	26.48701425	26.48938794	1.70717E-03	6
WO	5.452180711	0.291626433	26.48636156	26.48642589	1.87509E-04	5

8. Conclusion and future work

This study introduces a novel metaheuristic algorithm named the Tuckman Optimization Algorithm (TOA), which innovatively integrates principles from group psychology into optimization design. TOA implements a stage-driven evolutionary mechanism modeled on Tuckman's stages, Forming, Storming, and Norming, to dynamically govern population behavior. Core parameters such as Personnel Improvement Metric, Team Preparation, and stage index are adjusted during optimization to effectively balance the global exploration and local exploitation.

To further enhance population diversity and adaptability, a reconfiguration mechanism based on the logistic growth model is embedded, enabling the team composition to be updated in a self-organizing manner throughout the search process. This mechanism improves the algorithm's robustness and its ability to avoid premature convergence.

Extensive benchmark experiments across CEC2017, CEC2020, and CEC2022 benchmark suites, as well as the results on constrained engineering design problems, demonstrate that TOA consistently outperforms all the competing algorithms in convergence accuracy, stability, and scalability.

It is worth noting that TOA converges more slowly initially due to its emphasis on broad exploration of the solution space in the early stages. However, this exploration enables the algorithm to effectively escape local optima and achieve superior global search performance in later iterations. Overall, TOA achieves a well-coordinated balance between exploration and exploitation, offering a promising and versatile tool for solving complex real-world optimization problems.

Currently, TOA primarily draws on the "Forming," "Storming," and "Norming" stages from the Tuckman model to simulate the evolutionary process of group behavior. Future research could further incorporate the "Performing" and "Adjourning" stages to enhance the algorithm's local refinement capability and convergence efficiency during later optimization phases. Moreover, introducing the "Adjourning" stage may facilitate the development of adaptive termination criteria, thereby improving the flexibility and stability of the algorithm. Additionally, integrating multi-objective optimization concepts could expand TOA's applicability and performance in handling multi-objective problems, laying a foundation for its use in complex real-world decision-making scenarios.

Acknowledgments

This work is supported by the Youth Project of Science and Technology Research Program of Chongqing Education Commission of China (NO. KJQN202401101), and the Postgraduate Innovation Project of Chongqing University of Technology (NO. gzlcx20243149).

References

[1] Truong, D.N., Chou, J.S.: Metaheuristic algorithm inspired by enterprise development for global optimization and structural engineering problems with frequency constraints. Engineering Structures 318(000), 25 (2024)

[2] Sowmya, R., Premkumar, M., Jangir, P.: Newton-raphson-based optimizer: A new population-based metaheuristic algorithm for continuous optimization problems. Engineering Applications of Artificial Intelligence 128(000), 44 (2024)

[3] Tuckman, B.W., Jensen, M.A.C.: Stages of small-group development revisited. Group & organization studies 2(4), 419–427 (1977)

[4] Zirar, A., Muhammad, N., Upadhyay, A., Kumar, A., Garza-Reyes, J.A.: Exploring lean team development from the tuckman's model perspective. Production Planning & Control 36(4), 442–463 (2025)

[5] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE, Perth, Australia (1995)

[6] Mirjalili, S., Lewis, A.: The whale optimization algorithm. Advances in Engineering Software 95, 51–67 (2016)

[7] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H.: Harris hawks optimization: Algorithm and applications. Future Generation Computer Systems 97, 849–872 (2019)

[8] Xue, J., Shen, B.: Dung beetle optimizer: a new meta-heuristic algorithm for global optimization. The Journal of Supercomputing (2022). https: //doi.org/10.1007/s11227-022-04959-6

[9] Guan, Z., Ren, C., Niu, J., Wang, P., Shang, Y.: Great wall construction algorithm: A novel meta-heuristic algorithm for engineer problems. Expert Systems with Applications 233(000), 30 (2023)

[10] Yuan, C., Zhao, D., Heidari, A.A., Liu, L., Chen, Y., Wu, Z., Chen, H.: Artemisinin optimization based on malaria therapy:

Algorithm and applications to medical image segmentation. Displays 84 (2024)

[11] 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 Systems with Applications 238, 121744 (2024). https://doi.org/10.1016/j.eswa. 2023.121744

[12] Wang, J., Wang, W.-c., Hu, X.-x., Qiu, L., Zang, H.-f.: Black-winged kite algorithm: a nature-inspired meta-heuristic for solving benchmark functions and engineering problems. Artificial Intelligence Review 57(4), 98 (2024)

[13] Bai, J., Nguyen-Xuan, H., Atroshchenko, E., Kosec, G., Wang, L., Abdel Wahab, M.: Blood-sucking leech optimizer. Advances in Engineering Software 195, 103696 (2024). https://doi.org/10.1016/j.advengsoft. 2024.103696

[14] Zhao, W., Wang, L., Zhang, Z., Fan, H., Zhang, J., Mirjalili, S., Khodadadi, N., Cao, Q.: Electric eel foraging optimization: A new bio-inspired optimizer for engineering applications. Expert Systems with Applications 238, 122200 (2024). https://doi.org/10.1016/j.eswa.2023.122200

[15] Qi, A., Zhao, D., Heidari, A.A., Liu, L., Chen, Y., Chen, H.: Fata: An efficient optimization method based on geophysics. Neurocomputing 607, 36 (2024).

https://doi.org/10.1016/j.neucom.2024.128289

[16] Jia, H., Wen, Q., Wang, Y., Mirjalili, S.: Catch fish optimization algorithm: a new human behavior algorithm for solving clustering problems. Cluster computing (9), 27 (2024)

[17] Han, M., Du, Z., Yuen, K., Zhu, H., Li, Y., Yuan, Q.: Walrus optimizer: A novel nature-inspired metaheuristic algorithm. Expert Systems with Applications, 122413 (2023).

https://doi.org/10.1016/j.eswa.2023.122413

[18] Mohamed, A.W., et al.: Lshade with semi-parameter adaptation hybrid with cma-es for solving cec 2017 benchmark problems. In: 2017 IEEE Congress on Evolutionary Computation (CEC) (2017)

[19] Yang, X.-S., Deb, S.: Cuckoo search via l´evy flights. In: 2009 World Congress on Nature & Biologically Inspired Computing, Coimbatore, India, pp. 210–214 (2009)

[20] Yang, X.-S.: Firefly algorithms for multimodal optimization. In: Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications, Hokkaido University, Sapporo, Japan (2009)

[21] Yang, X.-S.: A new metaheuristic bat-inspired algorithm. In: International Workshop on Nature Inspired Cooperative Strategies for Optimization, Tenerife, Spain (2008)

[22] Yang, X.-S., Karamanoglu, M., He, X.: Flower pollination algorithm: A novel approach for multiobjective optimization. Engineering Optimization 46(9), 1222–1237 (2014)

[23] Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Advances in Engineering Software 69, 46–61 (2014)

[24] Mirjalili, S.: The ant lion optimizer. Advances in Engineering Software 83, 80–98 (2015)

[25] Mirjalili, S.: Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-Based Systems 89, 228–249 (2015)

[26] Mirjalili, S.: Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Computing and Applications 27, 1053–1073 (2016)

[27] Faramarzi, A., Heidarinejad, M., Mirjalili, S., Gandomi, A.H.: Marine predators algorithm: A nature-inspired metaheuristic. Expert Systems with Applications 152, 113377 (2020)

[28] Abdollahzadeh, B., Gharehchopogh, F.S., Mirjalili, S.: African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. Computers & Industrial Engineering 158, 107408 (2021)

[29] Xue, J., Shen, B., Zhang, C.: Sparrow search algorithm: Theory and applications. Neural Computing and Applications 32(10), 7641–7664 (2020)

[30] Hashim, F.A., Hussien, A.G.: Snake optimizer: A novel meta-heuristic optimization algorithm. Knowledge-Based Systems 242, 108320 (2022).

https://doi.org/10.1016/j.knosys.2022.108320

[31] Abdel-Basset, M., Mohamed, R., Abouhawwash, M.: Crested porcupine optimizer: A new nature-inspired metaheuristic. Knowledge-Based Systems 284, 111257 (2024)

[32] Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching–learning-based optimization: An optimization method for continuous non-linear large-scale problems. Information Sciences 183(1), 1–15 (2012)

[33] Das, B., Mukherjee, V., Das, D.: Student psychology based optimization algorithm: A new population based optimization algorithm for solving optimization problems. Advances in Engineering software 146, 102804 (2020)

[34] Trojovsky`, P.: A new human-based metaheuristic algorithm for solving optimization problems based on preschool education. Scientific Reports 13(1), 21472 (2023)

[35] Onay, F.K.: A novel improved chef-based optimization algorithm with gaussian random walk-based diffusion process for global optimization and engineering problems. Mathematics and Computers in Simulation 212, 195–223 (2023)

[36] Tian, Z., Gai, M.: Football team training algorithm: A novel sport-inspired meta-heuristic optimization algorithm for global optimization. Expert Systems with Applications 245, 123088 (2024)

[37] Leiva, V., Dhiman, G.: Archery algorithm: A novel stochastic optimization algorithm for solving optimization problems. energy 19, 22 (2022)

[38] Trojovsky`, P., Dehghani, M.: A new optimization algorithm based on mimicking the voting process for leader selection. PeerJ Computer Science 8, 976 (2022)

[39] Pira, E.: City councils evolution: a socio-inspired metaheuristic optimization algorithm. Journal of Ambient Intelligence and Humanized Computing 14(9), 12207–12256 (2023)

[40] Askari, Q., Younas, I., Saeed, M.: Political optimizer: A novel socioinspired meta-heuristic for global optimization. Knowledge-based systems 195, 105709 (2020)

[41] Wu, X., Li, S., Jiang, X., Zhou, Y.: Information acquisition optimizer: a new efficient algorithm for solving numerical and constrained engineering optimization problems. The Journal of Supercomputing 80(18), 25736– 25791 (2024)

[42] Holland, J.H.: Genetic algorithms. Scientific American 267(1), 66–73 (1992)

[43] Storn, R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)

[44] Rechenberg, I.: Evolutionstrategie: Optimierung Technischer Systeme Nach Prinzipien des Biologischen Evolution. Fromman-Hozlboog Verlag, Stuttgart (1973)

[45] Fogel, L.J., Owens, A.J., Walsh, M.J.: Artificial Intelligence Through Simulated Evolution. Wiley-IEEE Press, ??? (1966)

[46] Moscato, P., Cotta, C., Mendes, A., et al.: Memetic algorithms. New optimization techniques in engineering 141, 53–85 (2004)

[47] Simon, D.: Biogeography-based optimization. IEEE transactions on evolutionary computation 12(6), 702–713 (2008)

[48] Ghaemi, M., Feizi-Derakhshi, M.-R.: Forest optimization algorithm. Expert systems with applications 41(15), 6676–6687 (2014)

[49] Kiran, M.S.: Tsa: Tree-seed algorithm for continuous optimization. Expert Systems with Applications 42(19), 6686–6698 (2015)

[50] Meng, Z., Pan, J.-S.: Monkey king evolution: a new memetic evolutionary algorithm and its application in vehicle fuel consumption optimization. Knowledge-Based Systems 97, 144–157 (2016)

[51] Sulaiman, M.H., Mustaffa, Z., Saari, M.M., Daniyal, H., Mirjalili, S.: Evolutionary mating algorithm. Neural Computing and Applications 35(1), 487–516 (2023)

[52] Salimi, H.: Stochastic fractal search: a powerful metaheuristic algorithm. Knowledge-based systems 75, 1–18 (2015). https://doi.org/10.1016/j. knosys.2014.11.035

[53] Tarkhaneh, O., Alipour, N., Chapnevis, A., Shen, H.: Golden tortoise beetle optimizer: a novel nature-inspired meta-heuristic algorithm for engineering problems. arXiv preprint arXiv:2104.01521 (2021)

[54] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220(4598), 671–680 (1983)

[55] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: Gsa: A gravitational search algorithm. Information Sciences 179(13), 2232–2248 (2009). https: //doi.org/10.1016/j.ins.2009.03.004

[56] Sayed, G.I., Darwish, A., Hassanien, A.E.: Quantum multiverse optimization algorithm for optimization problems. Neural Computing and Applications 31, 2763–2780 (2019)

[57] Hu, G., Guo, Y., Zhong, J., Wei, G.: Iydse: Ameliorated young's double-slit experiment optimizer for applied mechanics and engineering. Computer Methods in Applied Mechanics and Engineering 412, 116062 (2023)

[58] Shareef, H., Ibrahim, A.A., Mutlag, A.H.: Lightning search algorithm. Applied Soft Computing 36, 315–333 (2015)

[59] Eskandar, H., Sadollah, A., Bahreininejad, A., Hamdi, M.: Water cycle algorithm - a novel metaheuristic optimization method for solving constrained engineering optimization problems. Computers & Structures 110–111, 151–161 (2012)

[60] Faramarzi, A., Heidarinejad, M., Stephens, B., Mirjalili, S.: Equilibrium optimizer: A novel optimization algorithm. Knowledge-Based Systems 191, 105190 (2020)

[61] Kaveh, A., Dadras, A.: A novel meta-heuristic optimization algorithm: Thermal exchange optimization. Advances in Engineering Software 110, 69–84 (2017)

[62] Hashim, F., Houssein, E.H., Mabrouk, M.S., et al.: Henry gas solubility optimization: A novel physics-based algorithm. Future Generation Computer Systems 101, 646–667 (2019)

[63] Salimi, M.: Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Applied Soft Computing 48, 533–551 (2016). https://doi.org/10.1016/j.asoc.2016.07. 028

[64] Bonebright, D.A.: 40 years of storming: a historical review of tuckman's model of small group development. Human Resource Development International 13(1), 111–120 (2010)

[65] Kordaki, M., Siempos, H.: The jigsaw collaborative method within the online computer science classroom. In: International Conference on Computer Supported Education, vol. 2, pp. 65–72 (2010). SCITEPRESS

[66] Verhulst, P.-F.: Notice sur la loi que la population poursuit dans son accroissement. Correspondance Math´ematique et Physique 10, 113–121 (1838)

[67] Yamashita, K., McIntosh, S., Kamei, Y., Hassan, A.E., Ubayashi, N.: Revisiting the applicability of the pareto principle to core development teams in open source software projects. In: Proceedings of the 14th International Workshop on Principles of Software Evolution, pp. 46–55 (2015)