



Triangulation topology aggregation optimizer: A novel mathematics-based meta-heuristic algorithm for continuous optimization and engineering applications

Shijie Zhao^{a,b,c,*}, Tianran Zhang^a, Liang Cai^a, Ronghua Yang^a

^a Institute of Intelligence Science and Optimization, Liaoning Technical University, Fuxin 123000, China

^b The School of Geomatics, Liaoning Technical University, Fuxin 123000, China

^c Institute for Optimization and Decision Analytics, Liaoning Technical University, Fuxin 123000, China

ARTICLE INFO

Keywords:

Meta-heuristic
Triangulation topology aggregation optimizer
Triangular similarity
Optimization
Engineering Applications

ABSTRACT

In recent years, numerous *meta-heuristic* algorithms based on swarm intelligence have been proposed and widely popularized. Although algorithms are designed by some specific behaviors of organisms, their heuristic paradigms and constructed modules are similar, resulting in algorithms still suffer from imbalance between exploration and exploitation for complex optimization problems. Combining mathematical properties with a stochastic search process in *meta-heuristic* algorithms contributes to breaking the traditional single/dual population-based evolutionary paradigm and facilitating individuals forward optimization. Taking this as motivation, this article proposes a novel mathematics-based *meta-heuristic* algorithm, named Triangulation Topology Aggregation Optimizer (TTAO), for solving continuous optimization and engineering applications. The core of the proposed algorithm is based on similar triangular topology in mathematics. The TTAO algorithm contains two strategies, i.e., the generic aggregation and the local aggregation, which contributes to iteratively constructing multiple similar triangular topological units to balance the exploration and exploitation. The former generates new vertexes through positive information exchange between different triangular topological units. And the latter constructs new units at promising locations according to the local optimum vertex of each unit. The performance of the TTAO algorithm is evaluated in comparison with some competitive algorithms on CEC2017 functions for different dimensions and 8 real-world engineering problems. Moreover, the Wilcoxon rank sum test is used to verify the effectiveness of the proposed algorithm. The experimental results show that the TTAO algorithm, compared with 10 competitors, obtains 23, 23, and 22 best average results for 29 CEC2017 functions on 30, 50 and 100 dimensions, and its Wilcoxon rank sum test scores still rank first on three different dimensional cases. And more numerical results verify the outstanding optimization performance of the TTAO algorithm effectively. Source codes of TTAO are publicly available at <https://www.mathworks.com/matlabcentral/fileexchange/136029-triangulation-topology-aggregation-optimizer>.

1. Introduction

Design problems for different areas of real life can usually be transformed into optimization problems. As engineering demands increase, optimization problems are also become more and more difficult, which usually have multimodal, multi-objective or large-scale characteristics (Keivanian & Chiong, 2022). In general, optimization methods include deterministic methods and *meta-heuristic* algorithms. Deterministic methods are proposed based on mathematical analysis theories, such as Newton's method, integer programming, and gradient descent (Li, Chen,

Liang, Luo, Zhao, & Dong, 2022). On the one hand, this type of methods is suitable for handling optimization problems with the differentiable objective function and a unique optimal solution. On the other hand, these deterministic methods are usually sensitive to the initial solution vector and can converge to a better accurate position only when starting from a good initial solution (Abualigah, Yousri, Abd Elaziz, Ewees, Al-Qaness, & Gandomi, 2021). Therefore, they suffer from many challenges for addressing complex engineering problems (Braik, Hammouri, Atwan, Al-Betar, & Awadallah, 2022). Fortunately, researchers take inspiration from nature-based phenomena or laws to develop a variety of

* Corresponding author at: Institute of Intelligence Science and Optimization, Liaoning Technical University, Fuxin 123000, China.

E-mail addresses: zhaoshijie@lntu.edu.cn (S. Zhao), ztr20010118@126.com (T. Zhang), service_cl@163.com (L. Cai), yrh_150620@163.com (R. Yang).

meta-heuristic algorithms (MAs). These algorithms do not rely on the concavity and convexity of the problem and are insensitive to the selection of initial values (Jamil & Yang, 2013). In the last few decades, MAs have been developed and become the crucial algorithms to solve complex optimization problems (Gokalp, Tasci, & Ugur, 2020; Esmaeili, Bidgoli, & Hakami, 2022).

Generally speaking, MAs search the solution space by inheriting the superior evolutionary information from the elite individual (Bertsimas and Tsitsiklis, 1993). The randomness of a *meta*-heuristic algorithm is self-organizing and intelligent. According to the current solution information and a certain iterative evolutionary strategy, MAs tune a poor solution or generate a new solution to make it as close as possible to the optimal solution (Blum & Roli, 2003). However, because of the limitations of random search, a *meta*-heuristic algorithm always needs several computations to get a satisfactory set when tackling a majority of practical application problems (Boussaïd, Lepagnot, & Siarry, 2013). More importantly, the exploitation and exploration are two critical concepts for a *meta*-heuristic algorithm (Črepinšek, Liu, & Mernik, 2013). The exploitation mainly focuses on enhancing the exact mining for a certain local solution space, which results in improving convergence accuracy of MAs as much as possible. Comparatively, the exploration is employed to enrich the diversity of the evolutionary population and search the whole solution space. Hence, how to strike a trade-off between the two is an important basic issue for a *meta*-heuristic algorithm with superior optimization performance (Pierezan & Coelho, 2018).

Existing population-based *meta*-heuristic algorithms usually take the whole population or the female-male subpopulations as the evolutionary unit and design some common optimization strategies according to the elite individual of the population or subpopulation. However, limited to the single/dual elite-guided evolution in this large-scale population and the inadequate inheritance of positive evolutionary information, it is easy to induce the high similarity or homogeneity of individuals within populations or subpopulations during iterations. As a result, these approaches still suffer from the biased global search and the poor local extreme avoidance, when solving complex optimization problems. On the other hand, according to the *No Free Lunch* theorem (Wolpert & Macready, 1997), it is recommended to develop more competitive algorithms for specific optimization problems. Discovering the fact in nature, the number three is the minimum combinatorial paradigm in certain closed systems, such as a trio of collaboration teams (e.g., Mathematical Contest in Modeling) and the triangular support in engineering structures. Moreover, triangular similarity, in mathematics (geometry), is a crucial fundamental property, which results in that multiple triangular topologies with the same properties but different sizes can be generated. Currently, MAs inspired by triangular topological similarity have not been developed. As a result, to break the canonical single/dual population-based evolutionary modes and comprehensively use the positive information for complex optimization problems, a novel mathematics-based Triangulation Topology Aggregation Optimizer, is proposed for continuous optimization and engineering applications. In the proposed TTAO algorithm, similar triangular topological units with different sizes as the basic evolutionary unit are formed by aggregation. Different from other MAs, the TTAO algorithm puts forward a new evolutionary guidance pattern, which mainly employs the optimal individual in each triangular unit to guide the evolution of in-unit individuals. Therefore, this kind of evolutionary population does not only rely on the global guided evolution of the elite, but also absorb the promising positive information of the best individual each unit. The main contributions of this study can be summarized as follows.

- The proposed algorithm combines the geometric domain with the *meta*-heuristic domain.
- The mathematical model of the proposed TTAO algorithm is established and explained in detail.

- A novel key individuals guided evolution strategy is proposed, that is, the individual evolution of each triangular topological unit is guided by the best individual within its unit.
- The proposed algorithm is critically compared with 10 state-of-the-art *meta*-heuristics from the aspects of convergence accuracy, stability, dimensional sensitivity on the CEC2017 function suite. Experimental results show that the proposed algorithm outperforms other comparison MAs.
- The proposed TTAO algorithm is applied for solving 8 engineering problems to investigate its better practical value.

The rest of the article is structured as follows. The related works about MAs are sorted out in Section 2. In Section 3, the motivation and mathematical model of the TTAO algorithm are given in detail. The proposed TTAO algorithm with other comparison algorithms is tested in Section 4. And Section 5 gives the application of the proposed TTAO algorithm on real-world engineering problems. Finally, in Section 6, conclusions of this article are summarized and future work is prospected.

2. Related work

In this section, some popular MAs are reviewed, as shown in Table 1. In general, MAs can be classified as four categories, i.e., the evolutionary algorithms, the swarm intelligence algorithms, the physics/mathematics-based algorithms and the life-based algorithms.

Accordingly, the inspirations of the evolutionary algorithms come from the natural evolution. These algorithms mainly encode the next generation of gene expression through genetic, mutation or crossover operators. Moreover, the classic representative is a Genetic Algorithm (GA) (Forrest, 1996). GA mimics natural selection of Darwinian evolution and Genetic principles of biological evolution. For a GA, the global best individual has a higher survival probability, thereby the population gradually approaches the global optimal solution. As another representative, the Differential Evolution (DE) (Storn & Price, 1997) is also motivated by the population evolution, but its evolutionary operation is different from GA. In addition, evolutionary algorithms also contain Evolution Strategy (ES) (Rechenberg, 1973) and Memetic Algorithm (MA) (Moscato, 1989).

Swarm intelligence algorithms are the most numerous *meta*-heuristic algorithms, which mimic the intelligent behaviors of natural organisms (e.g., predation, mating or reproduction). This kind of MAs evolves population through cooperating or fighting between multiple search agents, and then obtains more powerful agents (Zhao, Wang, & Mirjalili, 2022). Particle Swarm Optimization (PSO), the most classic of this category, is inspired by the foraging behavior of birds (Kennedy & Eberhart, 1995). In detail, particles update current positions by borrowing two extreme information positions, i.e., global optimum and individual optimum. Meanwhile, Ant Colony Optimization (ACO) is introduced by simulating the way of ants finding the shortest path (Dorigo, Maniezzo, & Colorni, 1996). In ACO, ants follow paths of high pheromone concentrations during foraging, and each ant releases pheromones as it walks. After several iterations, they can follow the shortest path to the vicinity of the food source. In recent years, other swarm intelligence optimization algorithms have continually been developed, e.g., Harris Hawks Optimization (HHO) (Heidari, Mirjalili, Faris, Aljarah, Mafarja, & Chen, 2019), Chimp Optimization Algorithm (ChOA) (Khishe & Mosavi, 2020), Smell Agent Optimization (SAO) (Salawudeen, Mu'azu, Yusuf, & Adedokun, 2021), Reptile Search Algorithm (RSA) (Abualigah, Abd Elaziz, Sumari, Geem, & Gandomi, 2022), Sand Cat Swarm Optimization (SCSO) (Seyyedabbasi & Kiani, 2023), Beluga Whale Optimization (BWO) (Zhong, Li, & Meng, 2022). In need of attention, human-based algorithms, such as Political Optimizer (PO) (Askari, Younas, & Saeed, 2020) and Forensic-Based Investigation (FBI) (Chou & Nguyen, 2020), can also be classified into this category.

Physics/mathematics-based algorithms, owning good search

Table 1
Reviews of some popular *meta-heuristics*.

Category	Algorithm	Inspiration
Evolutionary algorithms	Genetic Algorithm (GA) (Forrest, 1996)	Darwin's theory of evolution
	Differential Evolution (DE) (Storn & Price, 1997)	Group evolution
	Evolution Strategy (ES) (Rechenberg, 1973)	Biological evolution
	Memetic Algorithm (MA) (Moscato, 1989)	Darwin's natural evolution and Dawkins' cultural evolution
		Foraging behavior of birds
Swarm intelligence algorithms	Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995)	
	Ant Colony Optimization (ACO) (Dorigo et al., 1996)	Ants foraging to find the best path
	Harris Hawks Optimization (HHO) (Heidari et al., 2019)	Cooperative behavior of Harris hawks in rounding up prey
	Chimp Optimization Algorithm (ChOA) (Khishe & Mosavi, 2020)	Social hunting behavior of chimps
	Smell Agent Optimization (SAO) (Salawudeen et al., 2021)	Interactions between agents and an object emitting smell
	Reptile Search Algorithm (RSA) (Abualigah et al., 2022)	Hunting behavior of crocodiles
	Sand Cat Swarm Optimization (SCSO) (Seyyedabbasi & Kiani, 2023)	The survival behavior of sand cats
	Beluga Whale Optimization (BWO) (Zhong et al., 2022)	The behavior of beluga whales
	Political Optimizer (PO) (Askari et al., 2020)	Major political phases
	Forensic-Based Investigation (FBI) (Chou & Nguyen, 2020)	Police investigated, located, and tracked suspects
Physics/Mathematics-based	Artificial Electric Field Algorithm (AEFA) (Yadav, 2019)	Coulomb's law
	Henry Gas Solubility Optimization (HGSO) (Hashim et al., 2019)	Henley's solution theorem
	Archimedes Optimization Algorithm (AOA) (Hashim et al., 2021)	Archimedes' principle
	Atomic Orbital Search (AOS) (Azizi, 2021)	The principles of quantum mechanics
	Gradient-Based Optimizer (GBO) (Ahmadianfar et al., 2020)	Newton's method
	RUNge Kutta optimizer (RUN) (Ahmadianfar et al., 2021)	Runge Kutta method
	Arithmetic Optimization Algorithm (AOA) (Abualigah et al., 2021)	Arithmetic operators
	Football Game Algorithm (FGA) (Fadakar & Ebrahimi, 2016)	Football game
	Stochastic Paint Optimizer (SPO) (Kaveh et al., 2020)	The art and color of painting

performance, are proposed by modelling specific physical/mathematical theorems or certain physical phenomena. For instance, Artificial Electric Field Algorithm (AEFA) is inspired by Coulomb's law ([Yadav, 2019](#)). In AEFA, the charged particle and the charge quantity represent search agent and the fitness value, respectively. The former is updated by gravitational or repulsive forces. For the latter, the strength of a charge determines the optimal individual. Furthermore, some physics-based algorithms keep coming up, including Henry Gas Solubility Optimization (HGSO) ([Hashim, Houssein, Mabrouk, Al-Atabany, & Mirjalili, 2019](#)), Archimedes Optimization Algorithm (AOA) ([Hashim, Hussain, Houssein, Mabrouk, & Al-Atabany, 2021](#)), Atomic Orbital Search (AOS) ([Azizi, 2021](#)). On the other hand, Gradient-Based Optimizer (GBO), as

one of mathematics-based MAs, is based on Newton's method ([Ahmadianfar, Bozorg-Haddad, & Chu, 2020](#)). It uses gradient search rules and local transition operations to emphasize global and local search. More mathematics-based algorithms also include but are not limited to RUNge Kutta Optimizer (RUN) ([Ahmadianfar, Heidari, Gandomi, Chu, & Chen, 2021](#)) and Arithmetic Optimization Algorithm (AOA) ([Abualigah, Diabat, Mirjalili, Abd Elaziz, & Gandomi, 2021](#)).

For the fourth category, life-based algorithms are developed by taking inspiration from real life. For example, Football Game Algorithm (FGA) ([Fadakar & Ebrahimi, 2016](#)) and Stochastic Paint Optimizer (SPO) ([Kaveh, Talatahari, & Khodadadi, 2020](#)) are some instances.

3. Triangulation topology aggregation Optimizer

This section detailly introduces the proposed TTAO algorithm, involving the inspiration, the specific mathematical model and computational complexity.

3.1. Inspiration

A triangle is the most basic graph in plane geometry. In a finite or infinite dimensional space, a triangular topology can be regarded as a graph under its 2-dimensional subspace. The triangle is simpler and more stable than other topologies in certain closed systems. [Fig. 1](#) shows real-life applications of the triangular topology. Because of this advantage, many fields usually divide the research object into triangular topological units, and establish relevant models for identification and analysis ([Jones & Satherley, 2001](#); [Reddy, Padmini, Govindaraj, & Sudhakar, 2022](#)).

Triangular similarity is one of the most important properties for the geometry. A polygon can consist of a union of triangles ([Soifer, 2009](#)), which has important application value for computational geometry and computer science ([Laczkovich, 2021](#)). In addition, the problem of height or distance which is difficult to measure in practical engineering applications can be solved by using similar triangles. For image recognition, it can also judge whether two features are related according to the distance between the corresponding vertexes of similar triangles. In mathematics, similar decision theorems of two triangles are as follows. [Fig. 2\(a\)-\(d\)](#) indicates the schematics about four decision theorems, respectively.

- The newly formed triangle has a side parallel to one side of the original triangle, and the extension line of this line intersects the other sides or the extension lines of both sides, which is similar to the original triangle;
- If the proportions and angles of the corresponding sides of two triangles are equal, the two triangles can be regarded as similar;
- A triangle is similar to another triangle if their corresponding sides are in equal proportion;
- If the angles of two triangles are equal, then two triangles are similar.

The proposed TTAO algorithm is based on similar triangles. With iterative evolution, new vertexes are constantly generated in the search space and used to constitute similar triangles of different sizes. In the proposed TTAO algorithm, each triangle is regarded as a basic evolutionary unit with four agents, i.e., three vertices of the triangle and one inside random vertex. What's more, the core of aggregation is to group vertexes with the superior characteristics. Specifically, the TTAO algorithm employs aggregation to gather the vertexes with good information between or within different topological units. It should be noted that the constructed triangles are all equilateral triangles and derived from the second theorem to construct similar triangles.

3.2. The construction of the mathematical model

This subsection describes the mathematical model about the

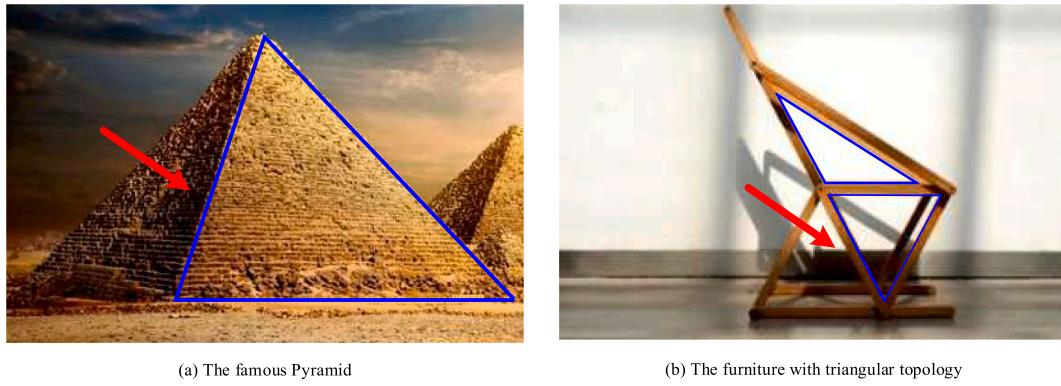


Fig. 1. Real-life applications of triangular topologies.

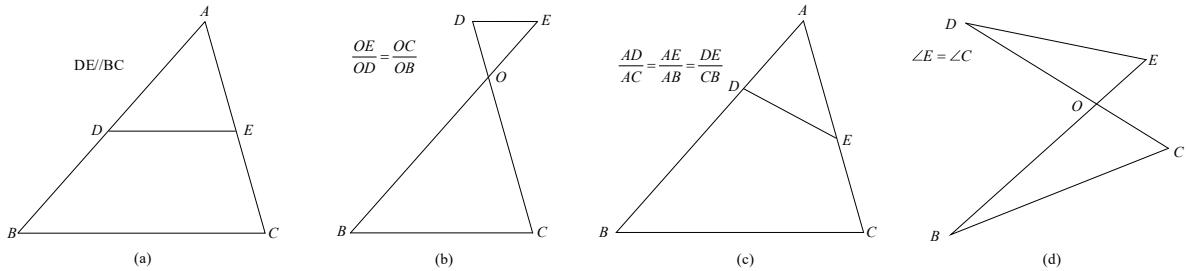


Fig. 2. Similar triangular topologies: (a) Judgment method 1, (b) Judgment method 2, (c) Judgment method 3 and (d) Judgment method 4.

proposed TTAO algorithm. The optimization process of TTAO mainly consists of two stages, i.e., aggregation between different and same units.

3.2.1. Initialization

First, the TTAO algorithm initializes the population to start the optimization process. The population size N and the variable dimension D are given. Each vertex in the triangular topological unit represents a search agent. In the TTAO algorithm, the number of individuals N can be divided into $\lfloor N/3 \rfloor$ triangular topological units, where $\lfloor \cdot \rfloor$ indicates the rounded down. The extra individuals are randomly generated in the search space. In the initialization phase, $\lfloor N/3 \rfloor$ agents are randomly produced in the feasible region, and the mathematical expression generated by each individual is

$$\overrightarrow{X_{i,1}} = r_0 \times (\overrightarrow{UB} - \overrightarrow{LB}) + \overrightarrow{LB} \quad (1)$$

where $\overrightarrow{X_{i,1}}$ represents the first search individual in the i^{th} triangular topological unit, and i is a positive integer between 1 to $\lfloor N/3 \rfloor$. r_0 denotes a random number between $[0, 1]$; \overrightarrow{LB} and \overrightarrow{UB} are lower and upper bounds of the variables, and their mathematical expression can be defined as

$$\begin{aligned} \overrightarrow{LB} &= [lb_1, \dots, lb_D] \\ \overrightarrow{UB} &= [ub_1, \dots, ub_D] \end{aligned} \quad (2)$$

3.2.2. Formation of triangular topological units

Since the dimension of a given problem is usually more than 2, only each two-dimensional surface is guaranteed to be an equilateral triangle in the TTAO algorithm. The formation of the triangular topological units utilizes the transformation between a polar coordinate and an ordinary coordinate system (Jiang, Wang, Hong, & Yen, 2020). A new direction vector with length $l * \vec{f}$ is located using the first vertex as the starting vertex in a spherical coordinate system and converted by trigonometric function into an ordinary coordinate system to form the second vertex.

The generated direction vector with length $l * \vec{f}$ is rotated $\pi/3$ anti-clockwise and then transformed by the coordinate system to obtain the third vertex. The expression of these vertexes can be written as

$$\begin{aligned} \overrightarrow{X_{i,2}} &= \overrightarrow{X_{i,1}} + l * \vec{f}(\theta) \\ \overrightarrow{X_{i,3}} &= \overrightarrow{X_{i,1}} + l * \vec{f}\left(\theta + \frac{\pi}{3}\right) \end{aligned} \quad (3)$$

where l represents the size of a triangular topological unit, which is mathematically expressed as

$$l = 9 * e^{-\frac{t}{T}} \quad (4)$$

In Eq. (4), t represents the current iterations number. T denotes the maximum number of iterations. l decreases with the number of iterations. In the early stages, the unit can generate a larger search range to focus on global search. Then, it develops deeply in a promising direction after gathering in the later stages. In order to ensure that the triangular topological unit can still be produced in the later iterations, l will not decrease to 0. Meanwhile, this setup prevents over-exploitation from getting trapped in a local extremum. $\vec{f}(\vec{\theta})$ and $\vec{f}(\vec{\theta} + \frac{\pi}{3})$ represent the direction vectors of the other two edges that are led by first point. They are calculated by Eq. (5).

$$\begin{aligned} \vec{f}(\vec{\theta}) &= [\cos\theta_1, \cos\theta_2, \dots, \cos\theta_{D-1}, \cos\theta_D] \\ \vec{f}(\vec{\theta} + \frac{\pi}{3}) &= \left[\cos\left(\theta_1 + \frac{\pi}{3}\right), \dots, \cos\left(\theta_{D-1} + \frac{\pi}{3}\right), \cos\left(\theta_D + \frac{\pi}{3}\right) \right] \end{aligned} \quad (5)$$

where $\vec{\theta} = [\theta_1, \dots, \theta_D]$ and $\theta_j (j = 1, \dots, D)$ are the random number between $[0, \pi]$.

Each group of triangular topological units is internally aggregated into fourth vertex. This point is formed in a linearly weighted manner to use individual information. The fourth vertex is defined by

$$\overrightarrow{X_{i,4}} = r_1 * \overrightarrow{X_{i,1}} + r_2 * \overrightarrow{X_{i,2}} + r_3 * \overrightarrow{X_{i,3}} \quad (6)$$

where r_1 , r_2 , and r_3 are random numbers between $[0, 1]$, and $r_1 + r_2 + r_3 = 1$. Thus, the fourth search agent is inside each triangular topological unit.

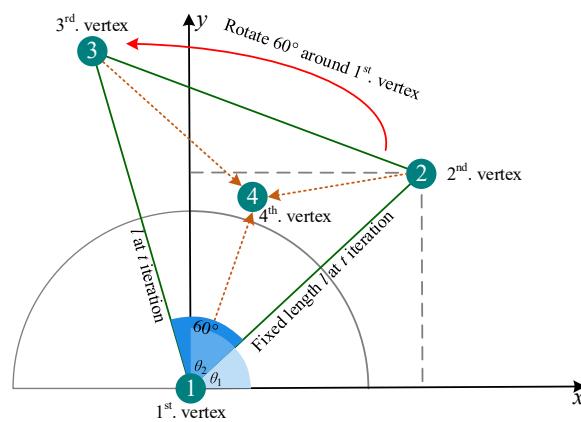
At the beginning of each iteration, the new similar triangular topological unit is obtained from a vertex and two side with same lengths. And the side length changes dynamically during iterations. The best vertex in each unit can be considered as the lead vertex in each iteration, which guides the evolution of other individuals for each unit. Besides, the other vertexes are formed according to the best vertex. Fig. 3 (a) and (b) display the construction of triangular topological units in 2- and 3-dimensional space. In Fig. 3 (a), the first (best) vertex for each unit is located. Transformations between a polar coordinate and an ordinary coordinate system are occurred. Random direction vector $[\theta_1, \theta_2]$ with a fix length l is generated and used different colors to emphasize the angles. Then, the second vertex is converted to the ordinary coordinate system by the trigonometric function. The vector with a fixed length consisting of the first vertex and the second vertex is rotated by 60° , namely $l * [\theta_1 + \pi/3, \theta_2 + \pi/3]$, and converted into an ordinary coordinate system to obtain the third vertex. It is worth noting that the coordinate system is not necessarily the standard coordinate system, and thus, θ is generated arbitrarily and regulated only by the cos function. The first three vertexes are weighted linearly to form the fourth vertex. For the 3-dimensional space, the vectors $l * [\theta_1, \theta_2, \theta_3]$ and $l * [\theta_1 + \pi/3, \theta_2 + \pi/3, \theta_3 + \pi/3]$ in polar coordinate are transformed into an ordinary coordinate system to achieve the second vertex and the third vertex. Fig. 3 (b) depicts the triangular unit by the x - z planes.

Fig. 4 explains states of the similar unit at the same period and different periods for 2-dimensional space. Since the randomness of θ , each triangular topological unit at t time shows the same size and direction, i.e., congruence. For different time, the direction of the triangular topology unit is the consistent after rotation. If the triangular unit of time t is transferred to a vertex in the triangular unit of time $t+1$, the two are rotationally similar.

Next, individuals within each group or between different groups of triangular topological units can exchange information to realize the global search and local mining.

3.2.3. Generic aggregation

Generic aggregation lays emphasis on the exploration phase. In this stage, the information of good individuals in different triangular units is gathered and created to new feasible solutions. Information interaction occurs between the best individual in each triangular topological units and the best individual in any set of randomly selected units. Inspired by



(a) Topological triangulation formed in 2-dimensional space

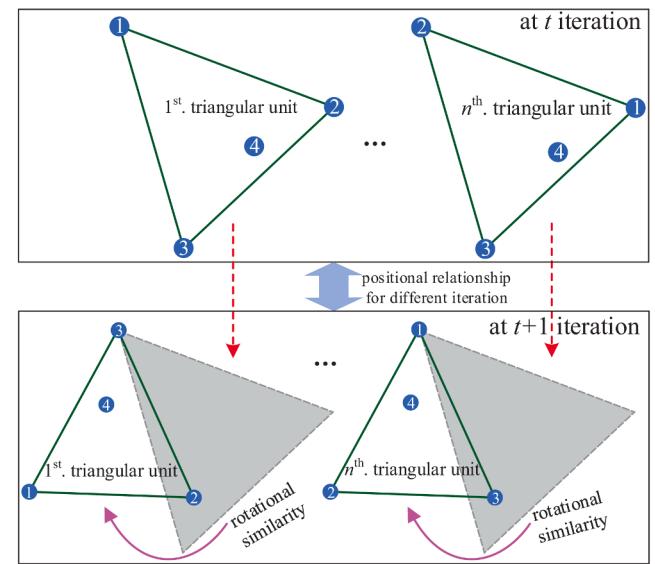


Fig. 4. States of the triangular topology unit in 2-dimensional space.

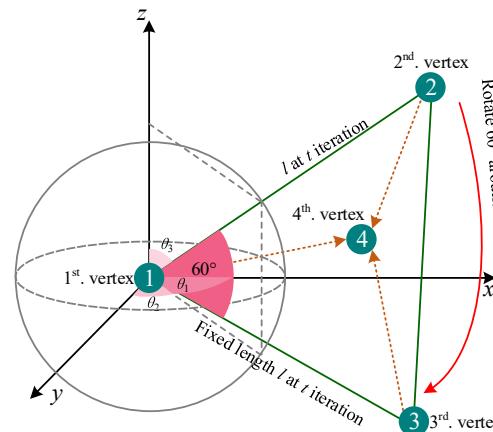
gene crossing in GA (Forrest, 1996), an information interaction mechanism is customized. There is a linear combination of different weights between each dimension variable of two positive individuals. The newly individual is produced in the better two-vertex connection, which is mathematically expressed as

$$\overrightarrow{X_{i,new1}^{t+1}} = r_4 * \overrightarrow{X_{i,best}^t} + (1 - r_4) * \overrightarrow{X_{rand,best}^t} \quad (7)$$

where r_4 is a random number between $[0, 1]$. $\overrightarrow{X_{i,best}^t}$ and $\overrightarrow{X_{rand,best}^t}$ denote the best individual for the unit i and a randomly selected unit at the t^{th} iteration. Further, the fitness value of $\overrightarrow{X_{i,new1}^{t+1}}$ is compared with the optimal or suboptimal search agent and updated the optimal agent. Assuming a minimization problem, the mathematical expressions of the optimal and suboptimal individuals updated at the $(t+1)^{\text{th}}$ iteration are

$$\begin{cases} \overrightarrow{X_{i,best}^{t+1}} = \overrightarrow{X_{i,new1}^{t+1}} & f_{\overrightarrow{X_{i,new1}^{t+1}}} < f_{\overrightarrow{X_{i,best}^t}} \\ \overrightarrow{X_{i,sbest}^{t+1}} = \overrightarrow{X_{i,new1}^{t+1}} & f_{\overrightarrow{X_{i,new1}^{t+1}}} < f_{\overrightarrow{X_{i,sbest}^t}} \end{cases} \quad (8)$$

where $\overrightarrow{X_{i,sbest}^t}$ represents the suboptimal individual at the t^{th} iteration.



(b) Topological triangulation formed in 3-dimensional space

Fig. 3. Schematic diagram of triangular topological units formed in 2- and 3-dimensional space.

$f(\cdot)$ is a function of the given problem.

Fig. 5 describes updating details of the optimal agent and the sub-optimal agent. The number of each vertex in a unit represents the ranking of its fitness value. Three different moving ways are included. The first is where the crossed individuals are better than the original position, e.g., upper left unit and lower left unit, upper right unit and lower left unit. The second is that the crossed individual is better than the suboptimal position, e.g., the upper second unit with the upper right unit, the upper left unit with the lower right unit. The second unit above with the lower left unit represents the worse individual after crossover. The exchange of information between groups is conducive to increase population diversity and enable individuals to adequately explore.

3.2.4. Local aggregation

Local aggregation mainly emphasizes the exploitation phase. In this stage, triangular topological units are aggregated internally. After the previous stage, a triangular topology was temporarily formed between the updated optimal or suboptimal individuals and the two vertexes in the group with good fitness values. In this case, the topology is not necessarily equilateral triangle. The position of the optimal individual is perturbed by a local region (both in terms of direction and step size), based on the movement vector difference constituted by the optimal and suboptimal individuals. Thus, each group is re-searched in a certain local area to realize the exploitation of each topological triangle unit. The new vertex is calculated as

$$\overrightarrow{X_{i,new2}^{t+1}} = \overrightarrow{X_{i,best}^{t+1}} + \alpha * (\overrightarrow{X_{i,best}^{t+1}} - \overrightarrow{X_{i,sbest}^{t+1}}) \quad (9)$$

where α is decreasing, which adjusts the aggregation scope size. α can be calculated as

$$\alpha = \ln\left(\frac{e - e^3}{T - 1} t + e^3 - \frac{e - e^3}{T - 1}\right) \quad (10)$$

The purpose of using suboptimal individual information is to prevent the optimal individual from trapping into a local extremum. After aggregating, the lead point of the temporary triangle unit should be guaranteed to be optimal within the unit. To make the convergence develop in a hopeful direction, the fitness values of the two vertexes after and before local mining are compared to determine the update of

the position. If the new individual outperforms the original one then the updating position, otherwise no update is performed. The corresponding mathematical expression is as follows.

$$\overrightarrow{X_{i,best}^{t+1}} = \begin{cases} \overrightarrow{X_{i,new2}^{t+1}} & f_{\overrightarrow{X_{i,new2}^{t+1}}} < f_{\overrightarrow{X_{i,best}^{t+1}}} \\ \overrightarrow{X_{i,best}^{t+1}} & \text{otherwise} \end{cases} \quad (11)$$

Fig. 6 shows a local aggregation diagram of the triangular topological units. The purpose of local aggregation is to aggregate the positive information inside each topological triangle unit. The dashed triangular topological unit indicates the temporary topology of the combination of the optimal vertex, the suboptimal vertex, and the third vertex after generic aggregation. The optimal individual in each unit is aggregated and updated to other better position within a certain range under the control of parameter α (the area of the gray circle). After the local aggregation of each temporary units, new similar topology units are constructed based on their best positions, formed by purple points. The local aggregation process makes each topological triangle unit to be mined as precisely as possible. Furthermore, for the later iterations, the range of variation becomes gradually smaller.

3.2.5. TTAO algorithm execution flow

First, the number of triangular topological units is determined by $[N/3]$. It should be noted that at each iteration, the remaining 1 or 2 individuals randomly generate new positions in the feasible region. The randomly generated positions are compared with the current individuals for fitness value, which aims to ensure the same number of triangular units in each iteration. Hence, the better top $[N/3]$ individuals are selected as the lead for the next iteration update. They are essentially the optimal individuals for each unit. In the TTAO algorithm, the optimal individual for each triangular topological unit implements three-stage updating per iteration, namely generated triangular topological units, generic aggregation and local aggregation. Algorithm 1 is the execution pseudocode of the TTAO algorithm.

3.3. Computational complexity

The computational complexity of a meta-heuristic algorithm is

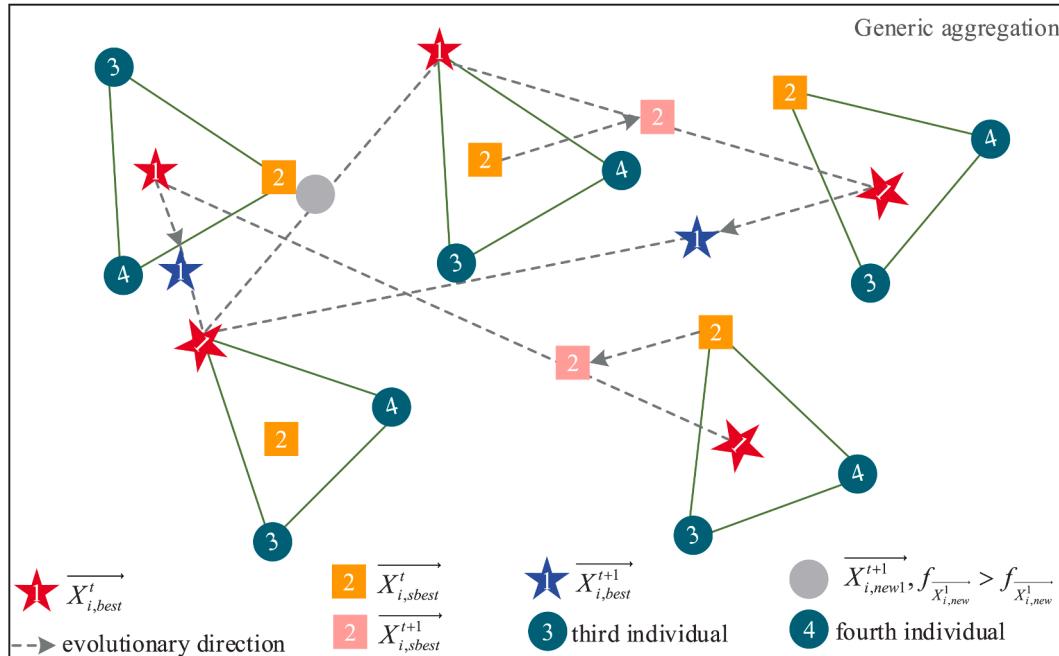


Fig. 5. Generic aggregation of multiple triangular topological units.

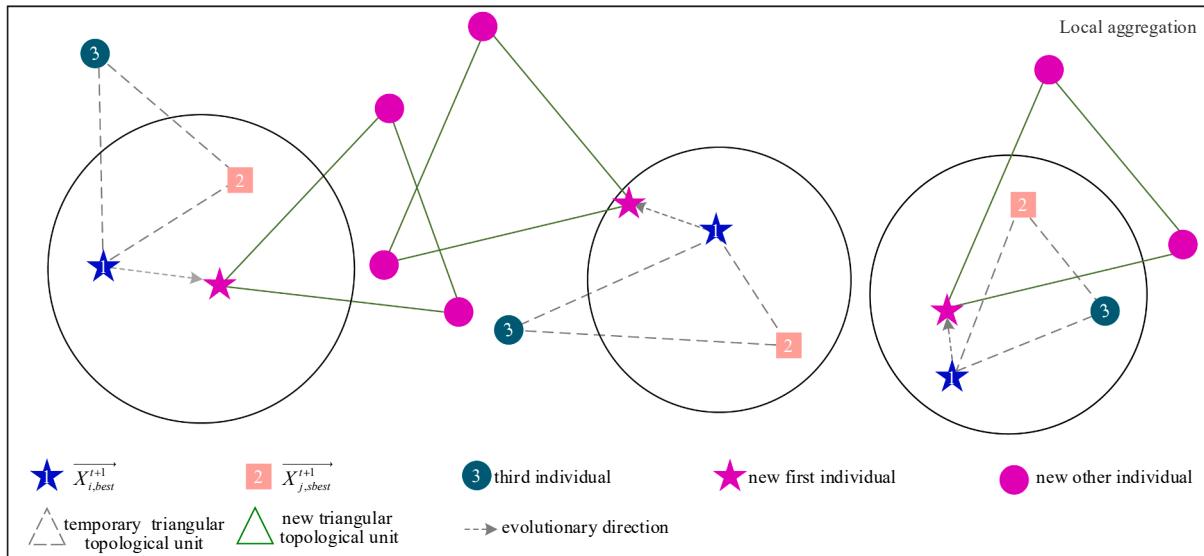


Fig. 6. Local aggregation of the triangular topological units.

connected with population size N , dimension D and maximum iteration times T . The TTAO algorithm has few parameters to be optimized. Therefore, the maximum order rule is used to analyze the computational complexity of the proposed TTAO algorithm. This rule ignores the constant and low order in the formula, and only calculates the magnitude of the largest order.

(1) Time complexity

The time complexity of the TTAO algorithm is analyzed considering the number of functions. It takes $O(N \times D)$ time to initialize the population. In each iteration, it costs $O(N \times D)$ time to update the triangular topological position and $O(N)$ time to evaluate the population. Iterations need to spend $O(T)$ time. In summary, the time complexity of the TTAO algorithm is $O(N \times D \times T)$.

(2) Space complexity

When all triangular topological units are constructed, the space of $O(N \times D)$ is occupied. During whole iterations, it takes $O(D \times T)$ space to save the optimal individual. As a result, TTAO algorithm's space complexity is $\max\{O(D \times T), O(N \times D)\}$.

Algorithm 1: Triangulation Topology Aggregation Optimizer

Input: The population size N , maximum number of iterations T and variable dimension D

Output: The optimal position $\overrightarrow{X_{best}}$ and its fitness value f_{best}

- 1: Initialize the triangle population
- 2: **while** ($t < T$) **do**
- /* Formation of triangular topological units */
- 3: Generate three other positions using Eqs. (3) and (6)
- 4: Rank the positions in each group of triangular topological units according to fitness values
- /* Generic aggregation */
- 5: Update positions using Eq. (7)
- 6: **if** $f_{\overrightarrow{X_{i,new1}}} < f_{\overrightarrow{X_{i,best}}}$ **do**
- $\overrightarrow{X_{i,best}} = \overrightarrow{X_{i,new1}}$
- 8: **elseif** $f_{\overrightarrow{X_{i,new1}}} < f_{\overrightarrow{X_{i,best}}}$ **do**
- $\overrightarrow{X_{i,best}} = \overrightarrow{X_{i,new1}}$
- 10: **end if**
- /* Local aggregation */
- 11: Update positions using Eq. (9)
- 12: **if** $f_{\overrightarrow{X_{i,new2}}} < f_{\overrightarrow{X_{i,best}}}$ **do**

(continued)

Algorithm 1: Triangulation Topology Aggregation Optimizer

- 13: $\overrightarrow{X_{i,best}} = \overrightarrow{X_{i,new2}}$
- 14: **end if**
- 15: Select the optimal population position X_{best} according to fitness values
- 16: **end while**
- 17: **Return** $\overrightarrow{X_{best}}$ and f_{best}

4. Results and discussions

The optimization performance of the proposed TTAO algorithm is analyzed in this section, mainly including four aspects, i.e., exploration and exploitation analysis, general optimization performance analysis, scalability analysis and convergence analysis. Experimental results and analysis are described in detail below.

4.1. Benchmark functions and comparison algorithm

CEC2017 suite (Awad, Ali, Suganthan, Liang, & Qu, 2017) is used to test the proposed TTAO algorithm. F1-F3, F4-F10, F11-F20 and F21-F30 are unimodal, simple multimodal, hybrid and composition functions, which can simulate various situations of real single-objective continuous optimization problems. Fig. 7 depicts the topologies of some CEC2017 functions in 3D space.

Ten state-of-the-art *meta-heuristic* algorithms, Harris Hawks Optimization (Heidari et al., 2019), Chimp Optimization Algorithm (Khishe & Mosavi, 2020), Smell Agent Optimization (Salawudeen et al., 2021), Reptile Search Algorithm (Abualigah et al., 2022) Sand Cat Swarm Optimization (Seyyedabbasi & Kiani, 2023), Beluga Whale Optimization (Zhong et al., 2022), Arithmetic Optimization Algorithm (Abualigah et al., 2021), Atomic Orbital Search (Azizi, 2021), Political Optimizer (Askari et al., 2020), and Reversible Elementary Cellular Automata algorithm (Seck-Tuoh-Mora, Lopez-Arias, Hernandez-Romero, Martinez, & Volpi-Leon, 2022). are selected as comparison algorithms. Table 2 shows the controlled parameters of all algorithms, which set according to original literatures.

To ensure a fair competition for each algorithm, all algorithms were independently performed on each function for 50 times, and the mean (Mean) and standard deviation (Std) are taken as evaluation indicators. The Mean and Std can measure average convergence accuracy and robustness of the algorithm. All algorithms were compared under the premise of the same evolution times 60000. Experiments are run on the

(continued on next column)

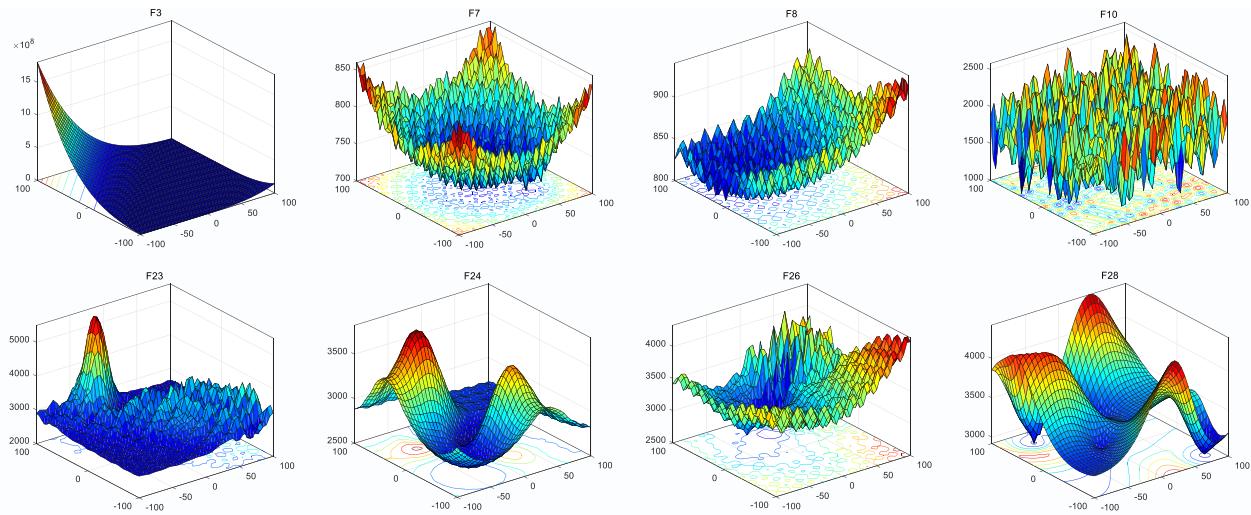


Fig. 7. Topological structure of CEC2017 benchmark functions in 3D space.

Table 2
Parameter settings of all algorithms.

Name	Parameters	Value
All algorithms	Maximum number of iterations (T)	1000
BWO	Population size (N)	60
	Probability of a whale falling (W_f)	[0.1, 0.05]
SCSO	Population size (N)	60
	Sensitivity range (r_G)	[2, 0]
	Control phase transformation (R)	$[-2r_G, 2r_G]$
SAO	Population size (N)	60
	Temperature (Te)	0.825
	Mass (m)	0.175
	Olfaction capacity (Olf)	0.75
	Step length (SL)	0.9
RSA	Population size (N)	60
	Sensitive parameter (α)	0.1
	Sensitive parameter (β)	0.005
AOS	Population size (N)	60
	Maximum number of Layers around nucleus (L_n)	5
	Foton Rate for position determination of electrons (P_n)	0.1
AOA	Population size (N)	60
	Sensitive parameter (α)	5
	Control parameter (μ)	0.5
ChOA	Population size (N)	60
	r_1, r_2	[0, 1]
	m	chaotic
HHO	Population size (N)	60
	Prey escape energy (E)	$[-2, 2]$
PO	Number of political parties (N)	8
	Max limit of party switching rate (λ)	1
	Number of function evaluations (NFEs)	72,000
RECAA	Number of smart-cells (n_s)	12
	Number of neighbors (n_{ne})	6
	Proportion ($prop$)	1.7
	Lower rounding ($lower_r$)	2
	Upper rounding ($upper_r$)	6
TTAO	Population size (N)	60
	r_0, r_1, r_2, r_3, r_4	[0, 1]

MATLAB R2018a software under Windows 11 system in Intel(R) Core (TM) I7-9750H CPU @2.60 GHz 2.59 GHz.

4.2. Exploration and exploitation analysis

In this subsection, the exploration and exploitation capacity of the TTAO algorithm is analyzed by a qualitative approach. For the early iterations, the algorithm conducts a global search to seek more

promising areas. At the later stage of iterations, it pays attention to search in a certain local domain, which aims to mine optimization accuracy. Unfortunately, these two properties are mutually inhibiting. If we only focus on the performance of one, the performance of the other will be weakened. In consequence, keeping a balance between the two is important research for MAs.

To more clearly observe the exploration and exploitation process of the TTAO algorithm for different iteration periods, the dimension diversity measurement method (Hussain, Salleh, Cheng, & Shi, 2019) is employed to analyze it. In this method, the average diversity of the population ($Diversity^t$) is calculated based on the j^{th} dimension average diversity ($Diversity_j$) of the variable. If the value of $Diversity^t$ increases, it means that the population is mainly explored. Otherwise, it is mainly exploited. Population average diversity ($Diversity^t$) and dimensional average diversity ($Diversity_j$) can be calculated as

$$Diversity^t = \frac{1}{D} \sum_{j=1}^D Diversity_j \quad (12)$$

$$Diversity_j = \frac{1}{N} \sum_{i=1}^N median(x_i^j) - x_i^j$$

where D denotes the variable dimension. x_i^j is the j^{th} dimension of the i^{th} individual. $median(x^j)$ is the median of the j^{th} variable in the population. The mathematical expression to measure the percentage of exploration versus exploitation is written as

$$Exploration\% = \frac{Diversity_t}{Diversity_{max}} \times 100 \quad (13)$$

$$Exploitation\% = \frac{|Diversity^t - Diversity_{max}|}{Diversity_{max}} \times 100$$

$$t = 1, 2, \dots, T$$

where T represents the maximum number of iterations. Div_t represents the population diversity at the t^{th} iteration, and Div_{max} is the maximum population diversity in the whole T iterations. The dimension $D = 10$, population size $N = 60$ and the maximum number of iterations $T = 1000$ are set.

Curves of Exploration% and Exploitation% are shown in Fig. 8. As depicted in Fig. 8, although the transformation times of Exploration% and Exploitation% obtained by the TTAO algorithm take distinct states for different functions, all of them show a trend, that is, emphasizing the global search in the early stages and gradually turning to the local mining in the later stages. On the unimodal F3 function, the proposed TTAO algorithm turns from exploration to exploitation earlier than

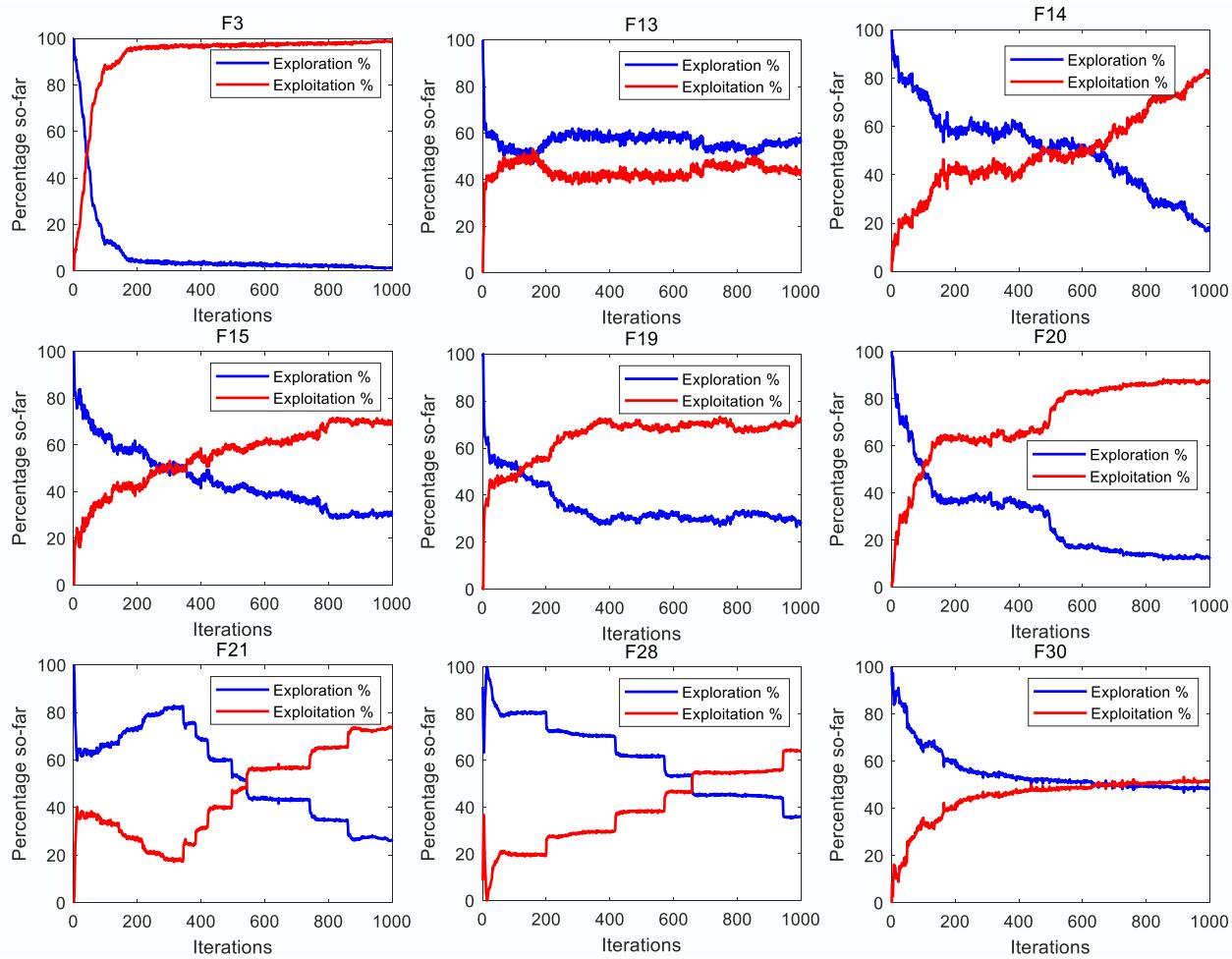


Fig. 8. Change curves of Exploration% and Exploitation%.

other functions. For this reason, the unimodal function has only one extreme point, and more attention is paid to convergence accuracy after exploring promising regions. Nevertheless, the TTAO algorithm requires longer exploration times on the multimodal F13, F14, F15, F19 and F20 functions. Their curve fluctuation amplitude shows more obvious than unimodal functions. On F21, F28 and F30 functions, the TTAO algorithm pays attention to exploitation in the middle and later stages of iterations. For example, on F21 and F28 functions, Exploration% and Exploitation % of the TTAO algorithm appear “stair-stepping”, which reflects the trade-off state of individuals with multiple local extreme values. To sum up, Exploitation% and Exploration% curves visually demonstrate the effectiveness of the TTAO algorithm in different iteration periods.

4.3. General optimization performance analysis

This subsection mainly verifies the general optimization performance of the TTAO algorithm on 30-dimensional CEC2017 functions. The statistical results obtained by all algorithms are reported in Table 3.

As can be seen from Table 3, the TTAO algorithm outperforms other algorithms in 38 out of 58 outcomes, which 23 out of 29 functions are the best average optimization results. The BWO, SCSO, SAO, RSA, AOS, AOA, ChOA, HHO, PO, and RECAA algorithms obtain the best results of 0, 0, 0, 1, 0, 0, 0, 0 and 5 functions. For unimodal functions (F1 and F3), the Mean and Std results of the TTAO algorithm all rank first. The Mean results of the TTAO algorithm on simple multimodal F4, F5, F6, F7, and F8 functions show best. The optimization results of the TTAO algorithm are inferior to RECAA and AOS on F9 and F10 functions. For hybrid functions (e.g., F12, F13, and F18), the TTAO algorithm's

optimization accuracy value is obviously better than other algorithms. And yet other comparison algorithms such as BWO, AOS and RSA have poor convergence accuracy and stability. In addition, the TTAO algorithm's good global optimization capacity is also shown on eight composition functions, namely F21, F23, F24, F25, F27, F28, F29 and F30 functions. The statistical results on multimodal and composition functions illustrate that the TTAO algorithm is better than other existing algorithms to avoid local extreme values. The reason for this result is that using some promising individuals to guide the evolution of each triangular topological units makes the population more fully explored and avoids effectively the local extreme value.

To further verify the optimization stability of the TTAO algorithm, Fig. 9 records the boxplots of all the algorithms on the 30-dimensional CEC2017 functions. It is clear that the TTAO algorithm has a smaller upper and lower bound gap in the 50 experiments while other algorithms have a larger gap. Moreover, the worst case of the TTAO algorithm is less than other algorithms.

4.4. Scalability analysis

The increasing of the optimization dimensions will lead to increase the optimization difficulty. Thus, the sensitivity of the algorithm to dimension analysis is crucial. There are 10, 30, 50, and 100 tested variable dimensions about the CEC2017 functions. Among them, 50-dimensional and 100-dimensional functions are similar to medium-dimensional and high-dimensional optimization problems. So, dimensions are extended to 50 and 100. Table 4 and Table 5 indicate the test results of all algorithms on the 50- and 100- dimensional functions.

Table 3

Statistical results obtained by TTAO and 10 comparison algorithms on 30 dimensional CEC2017 functions.

		BWO	SCSO	SAO	RSA	AOS	AOA	ChOA	HHO	PO	RECAA	TTAO
F1	Mean	4.87E+10	3.36E+09	6.46E+10	4.82E+10	3.20E+07	4.91E+10	2.59E+10	1.73E+07	7.40E+09	1.65E+08	8.29E+03
	Std	4.03E+09	2.11E+09	8.07E+09	7.25E+09	2.38E+07	8.45E+09	5.71E+09	3.82E+06	3.70E+09	1.08E+08	1.32E+04
F3	Mean	7.51E+04	4.51E+04	1.06E+05	7.81E+04	2.70E+04	7.73E+04	7.89E+04	2.23E+04	8.54E+04	4.84E+04	1.19E+04
	Std	5.00E+03	9.90E+03	1.79E+04	5.39E+03	8.23E+03	1.01E+04	8.34E+03	4.16E+03	3.83E+03	1.10E+04	3.43E+03
F4	Mean	1.13E+04	7.06E+02	1.79E+04	1.01E+04	5.49E+02	1.27E+04	4.27E+03	5.43E+02	1.22E+03	5.90E+02	5.04E+02
	Std	1.38E+03	1.85E+02	3.05E+03	3.45E+03	4.45E+01	3.04E+03	2.45E+03	2.87E+01	6.16E+02	2.65E+01	2.21E+01
F5	Mean	9.10E+02	7.36E+02	9.67E+02	9.12E+02	6.69E+02	8.60E+02	8.12E+02	7.38E+02	8.38E+02	6.81E+02	6.52E+02
	Std	1.77E+01	4.23E+01	3.86E+01	2.69E+01	3.88E+01	4.33E+01	2.88E+01	2.65E+01	3.23E+01	1.63E+01	2.33E+01
F6	Mean	6.86E+02	6.58E+02	6.94E+02	6.86E+02	6.44E+02	6.75E+02	6.66E+02	6.63E+02	6.71E+02	6.22E+02	6.22E+02
	Std	5.11E+00	9.33E+00	7.74E+00	7.31E+00	8.79E+00	7.06E+00	8.04E+00	5.41E+00	7.39E+00	2.63E+00	6.93E+00
F7	Mean	1.37E+03	1.12E+03	1.51E+03	1.37E+03	1.09E+03	1.36E+03	1.22E+03	1.25E+03	1.21E+03	9.57E+02	8.67E+02
	Std	2.92E+01	8.73E+01	5.91E+01	3.46E+01	8.10E+01	6.01E+01	4.95E+01	7.37E+01	9.50E+01	2.20E+01	2.66E+01
F8	Mean	1.13E+03	9.87E+02	1.18E+03	1.13E+03	9.36E+02	1.09E+03	1.06E+03	9.69E+02	1.06E+03	9.66E+02	9.24E+02
	Std	1.50E+01	3.06E+01	2.76E+01	2.28E+01	2.73E+01	3.26E+01	2.17E+01	2.45E+01	2.43E+01	1.84E+01	1.61E+01
F9	Mean	1.03E+04	5.56E+03	1.26E+04	1.02E+04	4.46E+03	6.70E+03	7.42E+03	7.50E+03	7.62E+03	1.80E+03	1.96E+03
	Std	8.31E+02	1.04E+03	2.34E+03	9.24E+02	1.33E+03	8.74E+02	1.22E+03	8.68E+02	1.19E+03	2.87E+02	9.50E+02
F10	Mean	8.40E+03	5.76E+03	9.14E+03	8.23E+03	5.10E+03	7.33E+03	8.43E+03	5.54E+03	6.41E+03	6.20E+03	5.34E+03
	Std	3.06E+02	7.64E+02	4.65E+02	4.29E+02	1.00E+03	6.01E+02	2.58E+02	8.24E+02	8.12E+02	4.32E+02	3.28E+02
F11	Mean	6.55E+03	1.89E+03	9.94E+03	9.72E+03	1.37E+03	7.71E+03	3.82E+03	1.26E+03	3.25E+03	1.40E+03	1.21E+03
	Std	9.55E+02	6.08E+02	2.71E+03	2.09E+03	8.41E+01	2.41E+03	8.50E+02	4.41E+01	9.37E+02	5.19E+01	3.55E+01
F12	Mean	9.58E+09	1.41E+08	1.53E+10	1.47E+10	3.82E+08	1.18E+10	5.17E+09	1.81E+07	3.81E+08	9.25E+06	5.77E+05
	Std	2.08E+09	2.19E+08	3.88E+09	3.00E+09	3.05E+08	3.13E+09	2.27E+09	1.57E+07	3.35E+08	5.32E+06	4.56E+05
F13	Mean	5.18E+09	2.20E+07	7.38E+09	1.11E+10	3.89E+05	5.94E+09	3.04E+09	5.75E+05	8.67E+07	1.96E+06	1.02E+04
	Std	1.74E+09	7.24E+07	3.80E+09	5.49E+09	6.52E+05	3.17E+09	2.76E+09	8.37E+05	2.28E+08	1.11E+06	7.57E+03
F14	Mean	2.20E+06	2.42E+05	4.34E+06	7.81E+06	1.18E+05	9.07E+05	6.43E+05	2.80E+05	1.15E+06	3.62E+04	8.38E+03
	Std	1.05E+06	3.08E+05	5.47E+06	8.44E+06	1.65E+05	8.96E+05	7.56E+05	2.74E+05	6.58E+05	2.61E+04	7.77E+03
F15	Mean	2.02E+08	3.38E+05	1.19E+09	7.02E+08	6.85E+04	2.33E+04	1.27E+07	7.11E+04	9.87E+05	4.86E+04	4.10E+03
	Std	9.51E+07	8.75E+05	7.60E+08	4.65E+08	4.56E+04	9.39E+03	1.54E+07	5.81E+04	3.84E+06	2.63E+04	3.37E+03
F16	Mean	5.18E+03	3.15E+03	6.84E+03	5.84E+03	2.93E+03	4.61E+03	3.86E+03	3.32E+03	3.64E+03	2.76E+03	2.66E+03
	Std	3.54E+02	3.60E+02	1.33E+03	1.08E+03	2.56E+02	1.01E+03	3.05E+02	4.40E+02	3.95E+02	1.89E+02	2.18E+02
F17	Mean	3.69E+03	2.37E+03	5.57E+03	6.22E+03	2.31E+03	3.04E+03	2.70E+03	2.62E+03	2.63E+03	1.98E+03	2.01E+03
	Std	3.95E+02	1.97E+02	2.66E+03	4.31E+03	2.19E+02	6.99E+02	2.01E+02	2.74E+02	2.62E+02	6.70E+01	9.93E+01
F18	Mean	2.46E+07	1.86E+06	5.42E+07	3.66E+07	6.45E+05	8.92E+06	3.64E+06	1.84E+06	7.20E+06	2.69E+05	1.52E+05
	Std	1.49E+07	2.10E+06	8.79E+07	2.41E+07	4.56E+05	7.32E+06	2.78E+06	1.84E+06	6.70E+06	1.43E+05	6.36E+04
F19	Mean	2.82E+08	1.56E+06	8.23E+08	7.48E+08	1.27E+06	1.72E+06	1.69E+08	5.08E+05	2.54E+06	6.72E+04	7.05E+03
	Std	1.23E+08	4.38E+06	6.45E+08	3.97E+08	8.90E+05	1.00E+05	2.33E+08	4.41E+05	4.38E+06	3.09E+04	6.54E+03
F20	Mean	2.89E+03	2.64E+03	3.11E+03	3.01E+03	2.52E+03	2.75E+03	3.08E+03	2.76E+03	3.12E+03	2.34E+03	2.44E+03
	Std	1.29E+02	1.78E+02	2.10E+02	1.68E+02	1.87E+02	2.05E+02	1.68E+02	1.77E+02	1.55E+02	6.47E+01	9.34E+01
F21	Mean	2.70E+03	2.51E+03	2.78E+03	2.72E+03	2.45E+03	2.65E+03	2.60E+03	2.56E+03	2.57E+03	2.45E+03	2.41E+03
	Std	3.99E+01	4.62E+01	5.72E+01	4.40E+01	2.32E+01	5.10E+01	3.20E+01	4.64E+01	4.02E+01	7.05E+01	3.17E+01
F22	Mean	8.21E+03	3.80E+03	1.01E+04	8.54E+03	3.56E+03	8.49E+03	9.66E+03	6.20E+03	4.87E+03	2.43E+03	2.75E+03
	Std	6.48E+02	1.80E+03	9.33E+02	9.96E+02	1.91E+03	8.74E+02	2.72E+02	2.03E+03	2.03E+03	2.16E+01	1.36E+03
F23	Mean	3.28E+03	2.91E+03	3.71E+03	3.36E+03	2.93E+03	3.45E+03	3.07E+03	3.18E+03	3.29E+03	2.84E+03	2.78E+03
	Std	4.33E+01	6.32E+01	1.62E+02	1.07E+02	5.87E+01	1.72E+02	5.66E+01	1.05E+02	2.53E+02	4.27E+01	2.71E+01
F24	Mean	3.51E+03	3.05E+03	3.90E+03	3.45E+03	3.11E+03	3.79E+03	3.26E+03	3.41E+03	3.92E+03	3.00E+03	2.94E+03
	Std	6.91E+01	6.17E+01	2.05E+02	2.82E+02	8.96E+01	1.61E+02	4.90E+01	1.34E+02	4.05E+02	1.90E+01	1.94E+01
F25	Mean	4.32E+03	3.07E+03	5.65E+03	4.98E+03	2.94E+03	5.34E+03	4.42E+03	2.93E+03	3.23E+03	2.96E+03	2.91E+03
	Std	1.30E+02	7.76E+01	4.49E+02	6.71E+02	3.53E+01	7.77E+02	4.81E+02	2.13E+01	1.10E+02	1.66E+01	1.94E+01
F26	Mean	1.03E+04	6.20E+03	1.27E+04	1.05E+04	6.28E+03	1.03E+04	6.82E+03	7.25E+03	6.12E+03	3.42E+03	4.74E+03
	Std	5.46E+02	1.08E+03	8.47E+02	9.30E+02	1.11E+03	8.27E+02	4.09E+02	1.28E+03	1.35E+03	5.48E+02	1.20E+03
F27	Mean	3.89E+03	3.36E+03	4.84E+03	4.01E+03	3.39E+03	4.46E+03	3.62E+03	3.42E+03	3.37E+03	3.28E+03	3.25E+03
	Std	1.40E+02	8.41E+01	5.97E+02	4.64E+02	8.83E+01	2.90E+02	8.25E+01	1.12E+02	1.14E+02	1.80E+01	1.67E+01
F28	Mean	6.18E+03	3.51E+03	7.92E+03	6.41E+03	3.33E+03	6.63E+03	4.56E+03	3.30E+03	3.82E+03	3.34E+03	3.24E+03
	Std	2.86E+02	1.11E+02	7.42E+02	7.54E+02	4.74E+01	8.38E+02	5.36E+02	3.21E+01	2.88E+02	2.34E+01	2.33E+01
F29	Mean	6.52E+03	4.49E+03	8.48E+03	7.22E+03	4.47E+03	6.56E+03	4.66E+03	4.65E+03	4.67E+03	3.94E+03	3.86E+03
	Std	4.75E+02	3.83E+02	2.02E+03	1.42E+03	4.63E+02	9.09E+02	1.91E+02	3.75E+02	4.30E+02	1.46E+02	2.01E+02
F30	Mean	7.68E+08	1.08E+07	1.85E+09	2.60E+09	7.51E+06	8.90E+08	5.79E+07	2.89E+06	1.61E+07	2.22E+05	8.85E+03
	Std	3.36E+08	7.80E+06	1.19E+09	1.11E+09	3.85E+06	8.58E+08	2.10E+07	1.67E+06	2.56E+07	1.01E+05	2.36E+03
+/≈/-		29/0/0	29/0/0	29/0/0	29/0/0	28/0/1	29/0/0	29/0/0	29/0/0	29/0/0	29/0/0	23/1/5

Note: +, ≈, and - respectively indicate that TTAO obtain better, almost equal and worse results than other algorithms.

According to Table 4, the TTAO algorithm won 39 out of 58 results. The BWO, SCSO, SAO, RSA, AOS, AOA, ChOA, HHO, PO and RECAA algorithms get the best result of 0, 0, 0, 2, 0, 0, 0, 0, 4 and 23 functions. Compared with the 30-dimensional case, the TTAO algorithm still maintains good performance on the 50-dimensional functions. AOS gains the superior Mean indicators on F5 and F10 functions. RECAA achieves the best outcomes on F6, F9, F22 and F26 functions. And the TTAO algorithm produces better Mean results on the other functions except these functions.

Table 5 reports that TTAO outperforms other comparison algorithms

on 22 functions. Among the 29 mean results obtained by BWO, SCSO, SAO, RSA, AOS, AOA, ChOA, HHO, PO and RECAA algorithms, the numbers of functions ranked first are 0, 2, 0, 0, 2, 0, 0, 1, 0 and 2, respectively. The TTAO algorithm can also keep superior Mean indicators on most 100-dimensional CEC2017 functions. For F3, F5, F6, F9, F10, F22 and F26 functions, the HHO, AOS, RECAA, AOS and SCSO, SCSO and RECAA algorithms obtain best results respectively. The Mean result of the TTAO algorithm is better than 50-dimensional outcome on F30 function. On F12, F18 and F19 functions, the superiority of the TTAO algorithm is particularly remarkable.

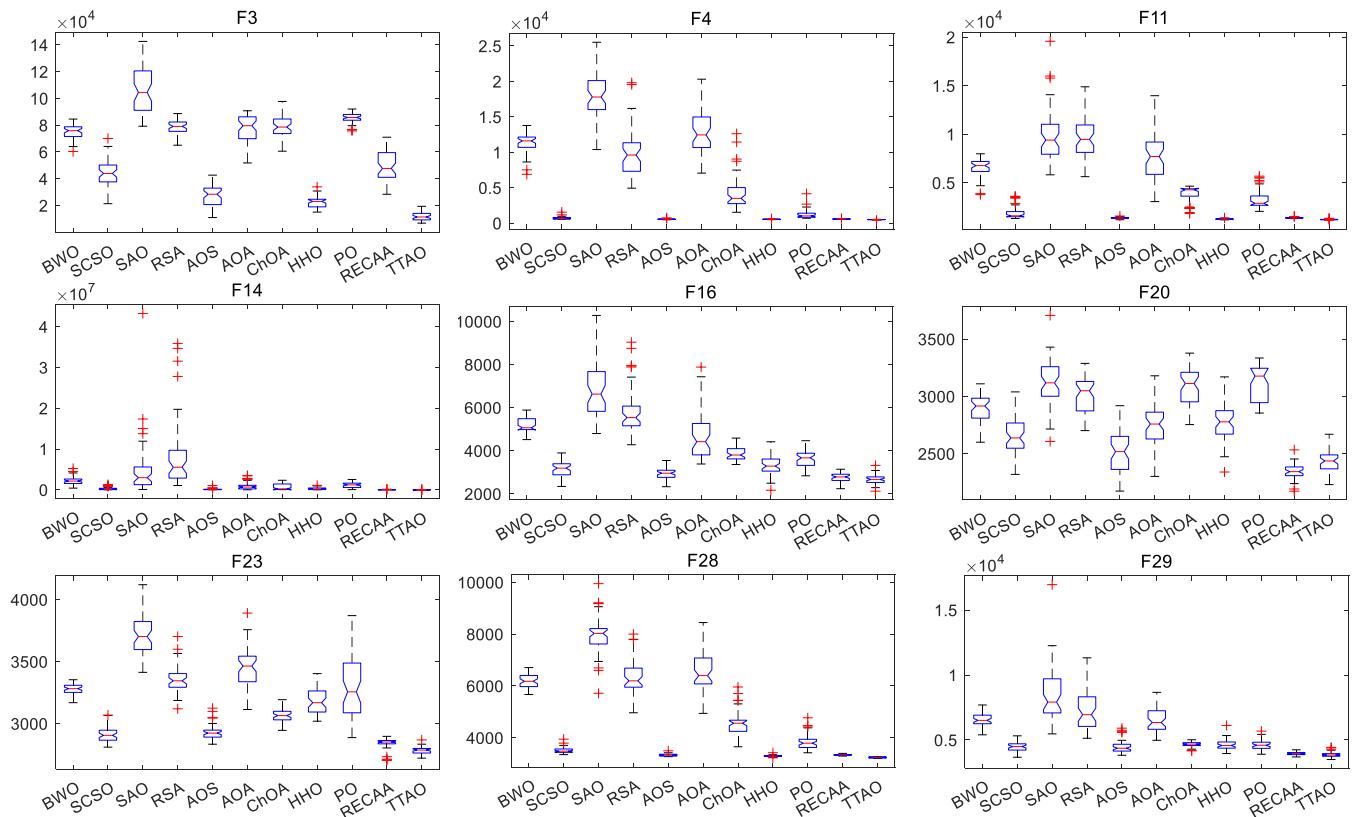


Fig. 9. Boxplots of TTAO and 10 comparison algorithms on 30 dimensional CEC2017 functions.

Overall, the TTAO algorithm is insensitive to dimensional changes, and still has good convergence in the case of medium- and high-dimensional functions.

4.5. Convergence analysis

This subsection analyses the convergence ability of algorithms. Convergence curves of all algorithms are shown in Fig. 10.

In Fig. 10, it appears that the decline rate of the average fitness value of all the algorithms is rapid in the early stages and slow in the late stages. This suggests that algorithms converge to the optimal value with the step of iterations. Specifically, the TTAO algorithm shows the best convergence performance. For F4, F11, F12, F15, F17, F23 and F30 functions, the TTAO algorithm converges faster than the other algorithms in the initial stage of iterations, and gradually transfers to exploitation with the step of iterations. On the other hand, TTAO algorithm continues to search for a better precision value before exploiting on the F18, F23, and F28 functions. Other comparison algorithms, such as AOS, HHO and BWO, are not explored enough, which generate a premature phenomenon. In addition, it is necessary to mention the accuracy values obtained by SAO, RSA and AOA under the preset maximum number of iterations, while the TTAO algorithm can be obtained at the early stage of iterations. The convergence analysis further demonstrates that the TTAO algorithm has excellent convergence performance from the perspective of the change of function fitness value.

4.6. Statistical tests

To verify the significant difference between the TTAO algorithm and comparison algorithms, the Wilcoxon rank sum test (Wilcoxon, 1992) is used for non-parametric test. At the 5 % test level, if $p \leq 0.05$ indicates that two algorithms have obvious differences on a function, otherwise the difference is not obvious. The testing results of the TTAO algorithm

vs. comparison algorithms on the CEC2017 functions with 30D, 50D and 100D are shown in Table 6, Table 7 and Table 8, and algorithms that accept the null hypothesis (H_0) are marked in bold.

From the data reported in Tables 6–8, the significant dominant ratio columns of the TTAO algorithm are 143/145, 148/154 and 139/145 on the 10, 50 and 100 dimensional CEC2017 functions. These results show that algorithms with better results significantly outperform competitive algorithms. Among the competitive results, the highest score is that the TTAO algorithm is better than the BWO, SAO, RSA, AOA and ChOA algorithms on all dimensional CEC2017 functions. While the lowest score is that the TTAO algorithm has obvious advantage over AOS on 22 100-dimensional CEC2017 functions. For F24 and F26 functions, the p -value for the TTAO algorithm vs. the AOS is greater than 0.05, indicating that the optimization performance of the two algorithms is equal. In the case of $D = 50$, although the statistical indicator obtained by the HHO algorithm on the F10 function outperforms the TTAO algorithm, the Wilcoxon rank sum test result accepts the null hypothesis. RECAA is not significantly superior to TTAO on 30-dimensional F6, 30-dimensional F10, 50-dimensional F20, 100-dimensional F6, 100-dimensional F9, and 100-dimensional F26 functions.

In conclusion, the testing results statistically demonstrate that the TTAO algorithm has stronger competitiveness.

4.7. Discussion

It can be concluded from Sections 4.2 to 4.6 that the proposed algorithm has good convergence accuracy and dimensional insensitivity. Below we reveal the main reasons behind TTAO's high performance.

The TTAO algorithm is different from other algorithms, we only save the $\lfloor N/3 \rfloor$ local best points within each triangular topological unit, and the remaining points are generated based on these points. Compared with the full use of the parental population, this strategy adds a certain degree of randomness, and these points cover the necessary information

Table 4

Statistical results obtained by TTAO and 10 comparison algorithms on 50-dimensional CEC2017 functions.

		BWO	SCSO	SAO	RSA	AOS	AOA	ChOA	HHO	PO	PECAA	TTAO
F1	Mean	1.01E+11	1.51E+10	1.16E+11	9.83E+10	5.25E+08	1.08E+11	5.09E+10	1.10E+08	3.29E+10	2.08E+09	9.95E+05
	Std	3.86E+09	5.72E+09	6.37E+09	8.92E+09	4.55E+08	9.40E+09	5.40E+09	3.50E+07	9.99E+09	7.36E+08	1.31E+06
F3	Mean	2.05E+05	1.04E+05	2.17E+05	1.67E+05	1.41E+05	1.70E+05	1.90E+05	8.69E+04	2.64E+05	1.75E+05	8.22E+04
	Std	2.02E+04	1.78E+04	3.12E+04	1.26E+04	2.28E+04	1.95E+04	1.91E+04	1.43E+04	1.72E+04	2.52E+04	1.39E+04
F4	Mean	3.09E+04	2.18E+03	4.50E+04	2.92E+04	8.46E+02	3.07E+04	1.15E+04	7.21E+02	5.25E+03	9.76E+02	6.04E+02
	Std	2.70E+03	8.31E+02	4.96E+03	5.97E+03	1.62E+02	7.85E+03	1.81E+03	6.90E+01	2.16E+03	8.03E+01	5.28E+01
F5	Mean	1.18E+03	9.15E+02	1.26E+03	1.16E+03	8.30E+02	1.14E+03	1.08E+03	8.95E+02	1.06E+03	9.17E+02	8.36E+02
	Std	2.21E+01	4.00E+01	3.94E+01	2.47E+01	3.48E+01	3.47E+01	3.33E+01	3.36E+01	4.66E+01	3.06E+01	5.17E+01
F6	Mean	7.00E+02	6.73E+02	7.10E+02	7.01E+02	6.62E+02	6.92E+02	6.83E+02	6.75E+02	6.83E+02	6.37E+02	6.44E+02
	Std	2.72E+00	6.72E+00	6.26E+00	5.76E+00	6.92E+00	5.35E+00	7.95E+00	4.92E+00	5.75E+00	3.62E+00	9.92E+00
F7	Mean	1.93E+03	1.60E+03	2.17E+03	1.94E+03	1.58E+03	1.94E+03	1.74E+03	1.83E+03	1.81E+03	1.27E+03	1.08E+03
	Std	4.19E+01	1.24E+02	9.37E+01	4.90E+01	1.07E+02	7.36E+01	5.09E+01	1.03E+02	1.34E+02	3.49E+01	6.71E+01
F8	Mean	1.48E+03	1.23E+03	1.56E+03	1.50E+03	1.15E+03	1.48E+03	1.36E+03	1.19E+03	1.34E+03	1.23E+03	1.13E+03
	Std	1.88E+01	4.53E+01	3.98E+01	2.80E+01	3.58E+01	4.44E+01	3.45E+01	3.12E+01	4.45E+01	3.14E+01	4.88E+01
F9	Mean	3.60E+04	1.77E+04	4.19E+04	3.64E+04	1.91E+04	2.77E+04	3.11E+04	2.57E+04	2.65E+04	8.81E+03	1.46E+04
	Std	2.28E+03	3.40E+03	3.81E+03	2.03E+03	4.69E+03	3.58E+03	4.11E+03	2.52E+03	4.45E+03	2.68E+03	4.26E+03
F10	Mean	1.46E+04	9.79E+03	1.55E+04	1.47E+04	9.03E+03	1.33E+04	1.50E+04	9.36E+03	1.18E+04	1.16E+04	9.66E+03
	Std	3.92E+02	9.13E+02	7.16E+02	4.22E+02	9.08E+02	7.74E+02	4.89E+02	9.20E+02	1.52E+03	6.31E+02	5.67E+02
F11	Mean	2.05E+04	4.88E+03	2.85E+04	2.00E+04	1.97E+03	2.21E+04	1.11E+04	1.56E+03	2.32E+04	2.10E+03	1.32E+03
	Std	1.52E+03	1.71E+03	5.18E+03	2.36E+03	2.96E+02	3.11E+03	1.54E+03	1.12E+02	4.71E+03	2.29E+02	4.80E+01
F12	Mean	5.60E+10	1.77E+09	9.65E+10	7.65E+10	4.04E+08	6.82E+10	2.94E+10	1.32E+08	5.50E+09	1.80E+08	4.88E+06
	Std	8.68E+09	1.89E+09	1.88E+10	1.70E+10	2.16E+08	1.83E+10	8.09E+09	5.58E+07	3.46E+09	9.33E+07	2.52E+06
F13	Mean	3.14E+10	2.49E+08	5.86E+10	4.76E+10	3.86E+06	3.14E+10	1.54E+10	3.54E+06	5.81E+08	1.36E+07	9.29E+03
	Std	6.57E+09	2.73E+08	1.65E+10	1.13E+10	3.49E+06	9.14E+09	7.67E+09	2.78E+06	5.24E+08	6.39E+06	4.15E+03
F14	Mean	3.95E+07	1.15E+06	1.16E+08	8.56E+07	8.23E+05	4.47E+07	2.75E+06	1.42E+06	8.40E+06	3.05E+05	1.53E+05
	Std	1.90E+07	1.28E+06	9.29E+07	6.64E+07	7.09E+05	4.49E+07	8.70E+05	1.33E+06	6.89E+06	1.70E+05	1.04E+05
F15	Mean	4.72E+09	2.54E+07	1.11E+10	7.44E+09	2.11E+05	3.56E+09	6.47E+08	5.70E+05	6.72E+07	1.51E+06	8.97E+03
	Std	1.24E+09	7.89E+07	3.39E+09	3.39E+09	9.37E+04	3.05E+09	1.28E+09	2.86E+05	1.18E+08	8.99E+05	5.25E+03
F16	Mean	8.24E+03	2.54E+07	1.13E+04	8.55E+03	4.14E+03	7.58E+03	5.79E+03	4.27E+03	5.77E+03	3.92E+03	3.48E+03
	Std	5.87E+02	7.89E+07	1.55E+03	1.46E+03	4.06E+02	1.23E+03	3.50E+02	5.19E+02	8.58E+02	2.53E+02	3.25E+02
F17	Mean	6.37E+03	3.70E+03	1.97E+04	1.33E+04	3.36E+03	8.44E+03	4.98E+03	3.73E+03	4.43E+03	3.14E+03	2.95E+03
	Std	7.65E+02	4.25E+02	1.72E+04	5.14E+03	3.84E+02	3.26E+03	5.95E+02	3.88E+02	4.30E+02	2.04E+02	2.18E+02
F18	Mean	1.07E+08	8.66E+06	2.33E+08	2.16E+08	3.52E+06	1.05E+08	1.38E+07	4.78E+06	4.96E+07	1.50E+06	9.89E+05
	Std	3.42E+07	1.04E+07	1.18E+08	9.64E+07	2.24E+06	1.07E+08	4.72E+06	3.98E+06	2.77E+07	8.54E+05	6.61E+05
F19	Mean	2.61E+09	6.01E+06	5.55E+09	4.72E+09	4.17E+06	2.52E+09	1.17E+09	1.10E+06	2.29E+07	6.65E+05	1.86E+04
	Std	7.83E+08	8.55E+06	1.62E+09	1.60E+09	2.89E+06	1.05E+09	1.47E+09	7.47E+05	3.47E+07	3.54E+05	7.08E+03
F20	Mean	3.97E+03	3.46E+03	4.36E+03	4.12E+03	3.14E+03	3.61E+03	4.13E+03	3.47E+03	4.00E+03	3.07E+03	3.05E+03
	Std	1.63E+02	3.25E+02	2.72E+02	1.81E+02	3.37E+02	2.75E+02	1.90E+02	2.92E+02	2.87E+02	1.57E+02	1.49E+02
F21	Mean	3.16E+03	2.75E+03	3.31E+03	3.17E+03	2.69E+03	3.06E+03	2.93E+03	2.86E+03	2.91E+03	2.69E+03	2.57E+03
	Std	4.18E+01	7.72E+01	9.37E+01	1.00E+02	6.95E+01	6.68E+01	4.79E+01	8.75E+01	7.78E+01	2.94E+01	4.23E+01
F22	Mean	1.64E+04	1.17E+04	1.75E+04	1.69E+04	1.08E+04	1.57E+04	1.72E+04	1.13E+04	1.39E+04	5.65E+03	1.10E+04
	Std	3.59E+02	1.49E+03	7.43E+02	5.34E+02	9.72E+02	7.00E+02	3.53E+02	8.42E+02	1.20E+03	4.62E+03	2.27E+03
F23	Mean	4.03E+03	3.31E+03	4.82E+03	4.06E+03	3.45E+03	4.37E+03	3.60E+03	3.81E+03	3.65E+03	3.23E+03	3.08E+03
	Std	7.53E+01	1.06E+02	2.10E+02	1.48E+02	1.45E+02	1.82E+02	6.78E+01	1.75E+02	1.83E+02	3.05E+01	7.58E+01
F24	Mean	4.39E+03	3.46E+03	5.12E+03	4.51E+03	3.67E+03	4.84E+03	3.83E+03	4.22E+03	4.94E+03	3.36E+03	3.22E+03
	Std	1.20E+02	1.05E+02	2.93E+02	6.44E+02	1.70E+02	2.35E+02	1.04E+02	2.25E+02	7.51E+02	4.10E+01	4.50E+01
F25	Mean	1.36E+04	4.20E+03	1.64E+04	1.31E+04	3.28E+03	1.53E+04	9.81E+03	3.21E+03	5.99E+03	3.41E+03	3.14E+03
	Std	7.24E+02	4.95E+02	8.93E+02	1.40E+03	1.04E+02	1.65E+03	1.06E+03	4.81E+01	8.22E+02	9.44E+01	3.12E+01
F26	Mean	1.63E+04	1.03E+04	1.81E+04	1.60E+04	1.04E+04	1.66E+04	1.14E+04	1.10E+04	1.10E+04	4.81E+03	8.44E+03
	Std	5.12E+02	1.98E+03	8.72E+02	6.58E+02	9.96E+02	1.21E+03	5.46E+02	1.39E+03	1.44E+03	1.31E+03	2.32E+03
F27	Mean	5.79E+03	4.09E+03	7.73E+03	6.20E+03	4.18E+03	6.69E+03	4.73E+03	4.36E+03	4.24E+03	3.73E+03	3.65E+03
	Std	3.39E+02	2.14E+02	9.58E+02	1.41E+03	3.46E+02	5.62E+02	2.28E+02	4.93E+02	3.83E+02	9.62E+01	8.05E+01
F28	Mean	1.17E+04	4.73E+03	1.55E+04	1.19E+04	3.80E+03	1.18E+04	6.66E+03	3.60E+03	6.30E+03	3.77E+03	3.40E+03
	Std	5.88E+02	4.97E+02	1.54E+03	1.34E+03	3.12E+02	1.39E+03	4.52E+02	7.98E+01	8.43E+02	1.18E+02	4.12E+01
F29	Mean	1.90E+04	6.37E+03	1.73E+05	8.38E+04	6.78E+03	2.99E+04	8.52E+03	6.00E+03	6.83E+03	5.32E+03	4.87E+03
	Std	6.19E+03	7.23E+02	2.09E+05	7.41E+04	8.68E+02	1.82E+04	2.74E+03	7.22E+02	8.07E+02	2.94E+02	3.59E+02
F30	Mean	4.04E+09	1.27E+08	8.97E+09	7.77E+09	1.31E+08	5.52E+09	1.25E+09	4.23E+07	2.17E+08	1.15E+07	9.98E+05
	Std	9.26E+08	4.99E+07	3.56E+09	2.05E+09	5.74E+07	3.13E+09	1.15E+09	1.06E+07	1.61E+08	7.78E+06	1.55E+05
+/-/-		29/0/0	29/0/0	29/0/0	29/0/0	26/0/3	26/0/0	29/0/0	28/0/1	29/0/0	25/0/4	

Note: +, ≈, and - respectively indicate that TTAO obtain better, almost equal and worse results than other algorithms.

of the previous generation, so that the TTAO algorithm can adaptively inherit the validity information and maintain the diversity of the population. It can be well proved from Fig. 10 that the TTAO algorithm shows faster convergence rate than other algorithms. In the iterative process, the proposed algorithm consists of two strategies, namely generic aggregation and local aggregation. Both of these aggregations actually use individuals with positive information to lead the evolution of the current individual, but the search preference is different. Generic aggregation mainly focuses on global exploration. It constructs better information through the exchange of the optimal information of

different triangular topological units to explore more promising positions. Local aggregation enables the information inside each unit to be developed effectively to ensure that the local regions are explored accurately. In the process of aggregation, because the positions of the best and the second best of each unit are constantly changed and updated, the triangular topological unit also gradually moves towards the global optimal direction. This advantage can be more clearly reflected on multimodal and composition functions of different dimensions.

The main reasons why the performance of other algorithms is

Table 5

Statistical results obtained by TTAO and 10 comparison algorithms on 100-dimensional CEC2017 functions.

		BWO	SCSO	SAO	RSA	AOS	AOA	ChOA	HHO	PO	RECAA	TTAO
F1	Mean	2.52E+11	6.76E+10	2.73E+11	2.45E+11	1.19E+10	2.68E+11	1.74E+11	1.96E+09	1.45E+11	2.10E+10	5.62E+07
	Std	5.47E+09	1.23E+10	5.80E+09	7.38E+09	8.64E+09	1.26E+10	1.04E+10	3.24E+08	2.31E+10	5.00E+09	1.47E+07
F3	Mean	3.50E+05	2.83E+05	4.77E+05	3.32E+05	2.98E+05	3.38E+05	5.10E+05	2.76E+05	3.62E+05	3.95E+05	4.07E+05
	Std	1.35E+04	2.29E+04	7.17E+04	1.06E+04	3.06E+04	1.43E+04	9.24E+04	1.59E+04	4.47E+03	4.69E+04	3.83E+04
F4	Mean	9.03E+04	7.61E+03	1.26E+05	8.59E+04	2.64E+03	8.87E+04	3.70E+04	1.63E+03	2.55E+04	3.26E+03	1.00E+03
	Std	7.02E+03	1.89E+03	1.44E+04	1.10E+04	9.47E+02	1.39E+04	6.08E+03	2.22E+02	7.94E+03	5.79E+02	7.04E+01
F5	Mean	2.09E+03	1.54E+03	2.22E+03	2.02E+03	1.41E+03	2.04E+03	1.90E+03	1.55E+03	1.91E+03	1.67E+03	1.49E+03
	Std	2.62E+01	7.05E+01	6.99E+01	4.01E+01	5.57E+01	6.05E+01	5.82E+01	4.15E+01	7.94E+01	4.96E+01	1.14E+02
F6	Mean	7.11E+02	6.79E+02	7.20E+02	7.11E+02	6.75E+02	7.08E+02	6.96E+02	6.85E+02	6.97E+02	6.64E+02	6.67E+02
	Std	2.08E+00	4.79E+00	3.56E+00	3.87E+00	4.48E+00	4.82E+00	4.42E+00	3.78E+00	5.31E+00	6.08E+00	9.37E+00
F7	Mean	3.83E+03	3.27E+03	4.25E+03	3.83E+03	3.19E+03	3.94E+03	3.55E+03	3.70E+03	3.54E+03	2.36E+03	1.85E+03
	Std	4.84E+01	1.69E+02	2.10E+02	6.13E+01	1.15E+02	6.83E+01	1.01E+02	1.31E+02	2.32E+02	9.01E+01	1.63E+02
F8	Mean	2.57E+03	1.98E+03	2.70E+03	2.51E+03	1.86E+03	2.49E+03	2.26E+03	2.01E+03	2.32E+03	2.02E+03	1.83E+03
	Std	3.02E+01	8.12E+01	5.70E+01	4.56E+01	8.11E+01	8.05E+01	6.49E+01	6.11E+01	8.82E+01	6.77E+01	1.23E+02
F9	Mean	7.71E+04	3.78E+04	8.85E+04	7.81E+04	4.73E+04	6.72E+04	7.49E+04	5.73E+04	6.06E+04	5.27E+04	5.59E+04
	Std	2.65E+03	5.05E+03	6.50E+03	3.81E+03	7.08E+03	4.83E+03	5.01E+03	5.30E+03	8.18E+03	8.32E+03	6.37E+03
F10	Mean	3.16E+04	2.03E+04	3.30E+04	3.14E+04	2.19E+04	3.00E+04	3.24E+04	2.22E+04	2.68E+04	2.65E+04	2.35E+04
	Std	6.43E+02	1.81E+03	8.29E+02	7.73E+02	2.03E+03	1.26E+03	4.46E+02	1.73E+03	1.82E+03	1.14E+03	7.38E+02
F11	Mean	2.92E+05	6.43E+04	2.84E+05	1.94E+05	1.33E+05	1.67E+05	1.45E+05	3.18E+04	3.53E+05	9.66E+04	9.78E+03
	Std	5.11E+04	1.43E+04	6.00E+04	4.34E+04	2.85E+04	2.65E+04	1.98E+04	9.54E+03	4.85E+04	2.17E+04	2.06E+03
F12	Mean	1.81E+11	1.53E+10	2.25E+11	1.83E+11	2.57E+09	1.83E+11	8.97E+10	8.49E+08	4.16E+10	3.78E+09	9.37E+07
	Std	9.52E+09	7.79E+09	1.50E+10	1.55E+10	1.13E+09	2.32E+10	1.29E+10	2.89E+08	1.31E+10	8.82E+08	3.98E+07
F13	Mean	4.03E+10	2.08E+09	5.44E+10	4.40E+10	2.19E+07	4.47E+10	2.49E+10	1.12E+07	5.22E+09	8.08E+07	4.71E+04
	Std	2.84E+09	2.02E+09	4.78E+09	6.80E+09	1.38E+07	5.69E+09	6.96E+09	2.32E+06	2.85E+09	5.29E+07	7.93E+04
F14	Mean	5.98E+07	7.70E+06	6.38E+07	9.61E+07	4.69E+06	7.62E+07	1.45E+07	4.03E+06	2.14E+07	4.15E+06	1.04E+06
	Std	1.72E+07	4.09E+06	3.94E+07	5.29E+07	2.41E+06	4.28E+07	3.65E+06	1.37E+06	7.98E+06	1.58E+06	3.96E+05
F15	Mean	2.05E+10	3.75E+08	2.84E+10	2.28E+10	3.07E+06	2.17E+10	8.77E+09	3.00E+06	1.08E+09	1.11E+07	6.38E+03
	Std	2.42E+09	5.04E+08	3.93E+09	5.22E+09	4.00E+06	4.70E+09	2.86E+09	1.77E+06	1.13E+09	6.42E+06	2.78E+03
F16	Mean	2.14E+04	9.37E+03	2.80E+04	2.16E+04	9.37E+03	1.98E+04	1.42E+04	8.10E+03	1.25E+04	9.23E+03	6.73E+03
	Std	1.41E+03	1.21E+03	2.61E+03	2.54E+03	1.19E+03	2.67E+03	1.02E+03	8.56E+02	1.34E+03	5.52E+02	1.11E+03
F17	Mean	2.74E+06	9.92E+03	1.65E+07	7.66E+06	7.21E+03	5.26E+06	2.14E+04	6.60E+03	1.66E+04	6.80E+03	5.40E+03
	Std	1.60E+06	1.84E+04	1.59E+07	5.87E+06	7.73E+02	4.42E+06	1.28E+04	6.05E+02	1.37E+04	4.55E+02	4.35E+02
F18	Mean	1.46E+08	7.74E+06	2.63E+08	1.65E+08	4.80E+06	1.31E+08	2.39E+07	5.14E+06	5.38E+07	4.51E+06	2.06E+06
	Std	4.44E+07	4.25E+06	1.40E+08	7.98E+07	2.28E+06	7.54E+07	8.30E+06	2.69E+06	3.42E+07	1.88E+06	8.70E+05
F19	Mean	2.00E+10	3.35E+08	2.95E+10	2.35E+10	2.73E+07	2.16E+10	6.54E+09	1.08E+07	9.88E+08	1.12E+07	1.23E+04
	Std	2.29E+09	5.94E+08	3.69E+09	4.94E+09	1.89E+07	5.18E+09	3.48E+09	4.62E+06	7.09E+08	7.11E+06	1.74E+04
F20	Mean	7.51E+03	5.98E+03	8.14E+03	7.59E+03	5.37E+03	7.11E+03	7.72E+03	5.97E+03	6.85E+03	5.92E+03	5.31E+03
	Std	2.40E+02	5.40E+02	3.47E+02	2.17E+02	5.86E+02	3.90E+02	4.43E+02	4.69E+02	3.72E+02	3.09E+02	3.58E+02
F21	Mean	4.66E+03	3.61E+03	5.03E+03	5.04E+03	3.71E+03	4.63E+03	4.21E+03	4.10E+03	4.12E+03	3.47E+03	3.17E+03
	Std	7.44E+01	1.63E+02	2.15E+02	2.54E+02	1.67E+02	1.68E+02	1.02E+02	1.68E+02	2.15E+02	4.52E+01	1.36E+02
F22	Mean	3.42E+04	2.40E+04	3.58E+04	3.42E+04	2.49E+04	3.32E+04	3.50E+04	2.51E+04	3.11E+04	2.68E+04	2.66E+04
	Std	5.59E+02	1.91E+03	1.34E+03	6.24E+02	2.24E+03	1.22E+03	5.15E+02	1.34E+03	2.11E+03	7.15E+03	8.13E+02
F23	Mean	5.98E+03	4.28E+03	6.97E+03	5.60E+03	4.90E+03	7.03E+03	5.10E+03	5.28E+03	6.96E+03	4.16E+03	3.69E+03
	Std	1.38E+02	1.91E+02	3.80E+02	1.95E+02	3.26E+02	4.13E+02	1.83E+02	2.91E+02	9.21E+02	8.26E+01	1.19E+02
F24	Mean	8.81E+03	5.30E+03	1.13E+04	9.51E+03	7.10E+03	1.12E+04	6.69E+03	7.15E+03	1.10E+04	5.05E+03	4.27E+03
	Std	3.67E+02	2.54E+02	9.57E+02	2.72E+03	8.44E+02	8.46E+02	2.80E+02	5.00E+02	2.17E+03	1.42E+02	1.57E+02
F25	Mean	2.64E+04	7.66E+03	3.06E+04	2.52E+04	4.49E+03	2.79E+04	1.59E+04	4.10E+03	1.41E+04	5.41E+03	3.68E+03
	Std	1.25E+03	1.13E+03	2.11E+03	2.25E+03	5.31E+02	2.62E+03	8.82E+02	1.45E+02	2.24E+03	2.34E+02	5.78E+01
F26	Mean	4.97E+04	2.89E+04	5.50E+04	4.93E+04	2.92E+04	5.16E+04	2.85E+04	2.79E+04	3.18E+04	1.85E+04	2.04E+04
	Std	1.38E+03	3.95E+03	1.91E+03	2.89E+03	2.64E+03	3.60E+03	1.26E+03	1.50E+03	2.89E+03	4.77E+03	3.90E+03
F27	Mean	1.15E+04	5.17E+03	1.57E+04	1.23E+04	5.10E+03	1.26E+04	6.73E+03	5.09E+03	5.14E+03	4.47E+03	4.37E+03
	Std	9.10E+02	3.71E+02	1.85E+03	2.96E+03	4.56E+02	1.50E+03	3.91E+02	6.25E+02	8.00E+02	1.67E+02	2.63E+02
F28	Mean	2.69E+04	1.05E+04	2.99E+04	2.96E+04	5.90E+03	3.34E+04	1.54E+04	4.50E+03	1.58E+04	6.51E+03	3.78E+03
	Std	8.65E+02	1.62E+03	8.58E+02	2.21E+03	1.44E+03	2.84E+03	1.07E+03	4.06E+02	2.28E+03	5.57E+02	5.44E+01
F29	Mean	2.99E+05	1.29E+04	1.03E+06	6.58E+05	1.28E+04	4.22E+05	3.89E+04	1.05E+04	1.81E+04	1.07E+04	8.30E+03
	Std	1.32E+05	2.00E+03	7.70E+05	3.51E+05	1.50E+03	3.24E+05	2.09E+04	5.74E+02	7.99E+03	5.90E+02	6.38E+02
F30	Mean	3.58E+10	1.66E+09	4.61E+10	4.16E+10	5.10E+08	3.82E+10	1.67E+10	9.42E+07	3.78E+09	1.31E+08	5.52E+05
	Std	4.70E+09	1.51E+09	6.24E+09	4.33E+09	2.63E+08	7.19E+09	3.13E+09	3.53E+07	2.74E+09	1.35E+08	3.39E+05
+/-/-		29/0/0	27/0/2	29/0/0	29/0/0	24/0/5	29/0/0	29/0/0	28/0/1	29/0/0	25/0/4	

Note: +, ≈, and - respectively indicate that TTAO obtain better, almost equal and worse results than other algorithms.

inferior to the TTAO algorithm are as follows. The BWO, SAO, RSA and HHO algorithms all set parameters to regulate exploration and exploitation, and update the current individual based on the randomly selected individual in the exploration stage and the best individual in the exploitation stage. However, the different strategies designed by each algorithm result in different results. It can be seen from the results that randomly selected individuals may carry information that is detrimental to the evolution of the population. The AOS algorithm considers the combination of several individuals to construct the movement vector, which is superior to other algorithms to some extent. The AOA algorithm

emphasizes the exploration by multiplying or dividing random positions regulated by math optimizer accelerated (MOA), and the addition and subtraction operations focus on the exploitation based on the best individual. Although the execution efficiency of the algorithm is fast, if the results of multiplication and division or addition and subtraction are too different in each iteration, it is likely to lead to too much exploration or exploitation of the algorithm. The parameter control range used by the ChOA algorithm is insufficient, and it is easy to fall into local extreme value. The PO algorithm assigns dual roles to the search agent, party members and election candidates, and searches optimization through

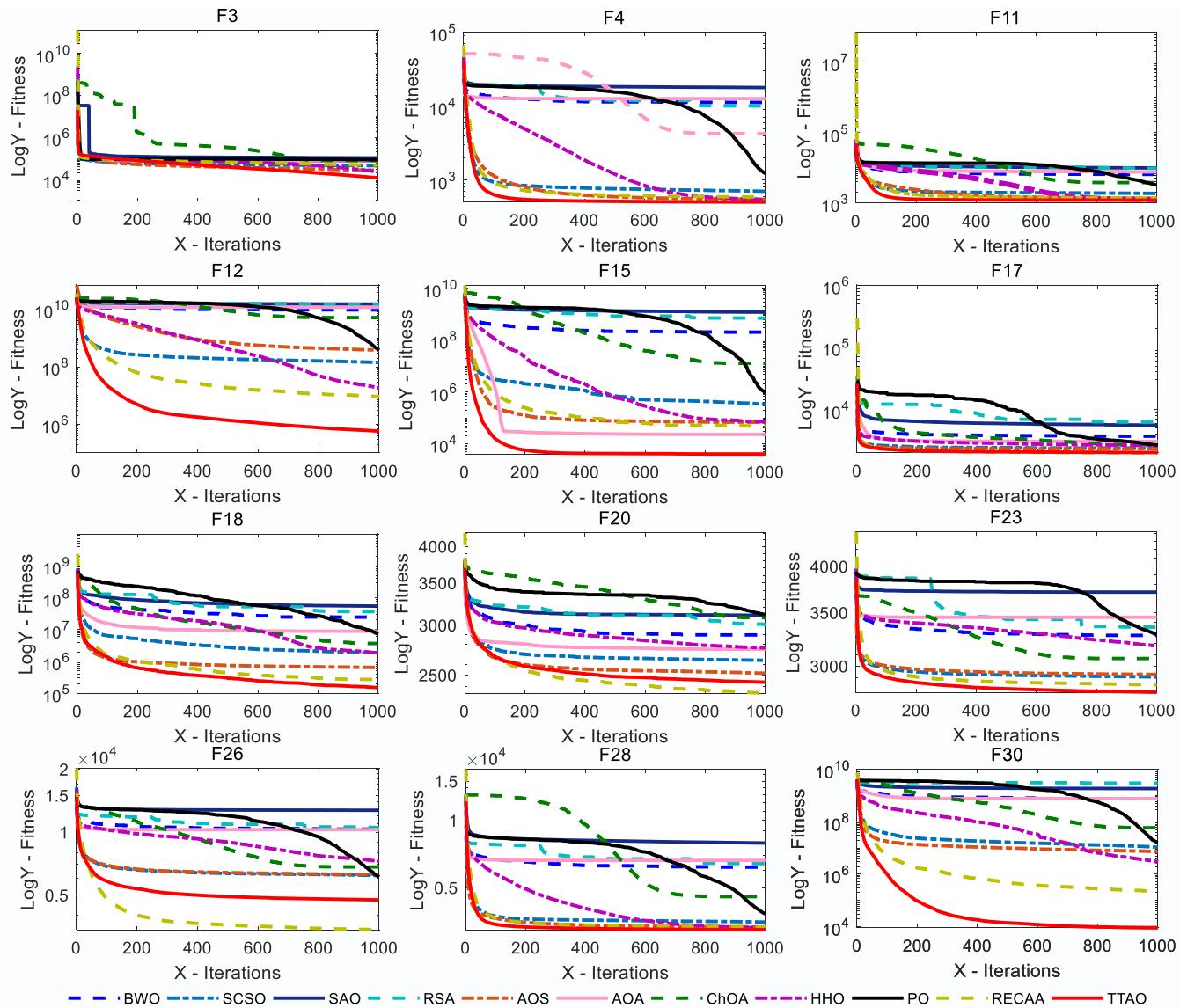


Fig. 10. Convergence curves from TTAO and 10 comparison algorithms on 30 dimensional CEC2017 functions.

the stages of campaigning, party switching, inter-party elections, and parliamentary business. However, being confined to a simpler setup of the total number of political parties, the number of party members, and the total number of elections leads to the algorithm inability to effectively mine promising positions and produce weak search optimization accuracy for complex optimization problems. The RECAA algorithm develops search strategies for population optimization by elementary cellular automata evolutionary rules. The utilization of the information from the intelligent units enables the algorithm to obtain good results, but performs poorly on F1, F12, F14, F19 and F30 functions.

Although the TTAO algorithm has good optimization ability on most functions, there is still room for improvement on F6, F9 and F10 functions. Some parameters can be introduced into the algorithm to perturb the individual moving vector, such as Levy function, Cauchy mutation operator, etc.

5. Engineering applications

The proposed TTAO algorithm is implemented to address some real-world optimization problems to verify planning and applicability. Table 9 describes the details of the engineering problems and population size set by the TTAO algorithm on different engineering problems. The

external penalty function method is used to transform the constrained problem into an unconstrained one. The external penalty function method is described as follows.

For a general nonlinear optimization problem $f(\vec{x})$, the general expression is

$$\begin{aligned} & \min f(\vec{x}) \\ & s.t. g_i(\vec{x}) \leq 0, i = 1, \dots, q \\ & h_j(\vec{x}) = 0, j = 1, \dots, p \end{aligned}$$

The external penalty function can be expressed as

$$\phi(\vec{x}) = f(\vec{x}) \pm \left[\sum_{i=1}^q r_i \times G_i + \sum_{j=1}^p c_j \times L_j \right]$$

where $\phi(\vec{x})$ is the unconstrained function needs to optimize. r_i and c_j are penalty factors. G_i and L_j are functions of the constraints $g_i(\vec{x})$ and $h_j(\vec{x})$, respectively, their general forms are

$$G_i = \max[0, g_i(\vec{x})]^\beta$$

$$L_j = |h_j(\vec{x})|^\gamma$$

Table 6

Wilcoxon rank sum test results from TTAO and 10 comparison algorithms on 30 dimensional CEC2017 functions.

	BWO	SCSO	SAO	RSA	AOS	AOA	ChOA	HHO	PO	RECAA
F1	7.07E-18									
F3	7.07E-18	7.07E-18	7.07E-18	7.07E-18	2.26E-14	7.07E-18	7.07E-18	2.54E-16	7.07E-18	7.07E-18
F4	7.07E-18	3.32E-17	7.07E-18	7.07E-18	8.31E-10	7.07E-18	7.07E-18	1.85E-10	7.07E-18	8.46E-18
F5	7.07E-18	2.85E-16	7.07E-18	7.07E-18	2.00E-02	7.07E-18	7.07E-18	1.37E-17	7.07E-18	6.07E-09
F6	7.07E-18	7.07E-18	7.07E-18	7.07E-18	3.79E-16	7.07E-18	7.07E-18	7.07E-18	7.07E-18	8.93E-01
F7	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.50E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	1.37E-17
F8	7.07E-18	2.05E-15	7.07E-18	7.07E-18	3.29E-02	7.07E-18	7.07E-18	1.95E-13	7.07E-18	7.27E-15
F9	7.07E-18	2.62E-17	7.07E-18	7.07E-18	2.95E-14	7.07E-18	7.50E-18	7.07E-18	7.07E-18	5.42E-01
F10	7.07E-18	1.85E-03	7.07E-18	7.07E-18	1.76E-02	7.97E-18	7.07E-18	1.30E-01	9.92E-13	1.64E-14
F11	7.07E-18	7.50E-18	7.07E-18	7.07E-18	2.71E-15	7.07E-18	7.07E-18	3.42E-08	7.07E-18	9.54E-18
F12	7.07E-18									
F13	7.07E-18	8.99E-18	7.07E-18							
F14	7.07E-18	3.38E-15	7.07E-18	7.07E-18	7.92E-16	4.46E-17	7.97E-18	5.32E-17	7.50E-18	2.51E-14
F15	7.07E-18	2.07E-17	7.07E-18	7.07E-18	8.99E-18	7.07E-18	7.07E-18	7.07E-18	7.97E-18	7.07E-18
F16	7.07E-18	1.12E-10	7.07E-18	7.07E-18	6.11E-07	7.07E-18	7.07E-18	1.04E-12	4.73E-17	5.18E-03
F17	2.63E-23									
F18	7.07E-18	6.69E-16	7.07E-18	7.07E-18	2.09E-12	7.07E-18	7.07E-18	5.05E-12	1.01E-16	7.05E-07
F19	7.07E-18	7.07E-18	7.07E-18	7.07E-18	1.08E-17	7.07E-18	7.07E-18	9.54E-18	7.07E-18	1.73E-17
F20	1.45E-17	2.81E-08	9.54E-18	7.50E-18	2.76E-02	8.19E-12	7.07E-18	5.02E-14	7.07E-18	1.08E-06
F21	7.07E-18	7.08E-16	7.07E-18	7.07E-18	4.92E-10	7.07E-18	7.07E-18	7.50E-18	7.50E-18	4.92E-11
F22	1.29E-17	6.98E-13	7.50E-18	7.97E-18	1.80E-12	1.84E-17	7.07E-18	2.56E-15	2.27E-13	5.56E-12
F23	7.07E-18	2.02E-16	7.07E-18	7.07E-18	1.01E-17	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.44E-12
F24	7.07E-18	1.31E-15	7.07E-18	7.07E-18	1.91E-16	7.07E-18	7.07E-18	7.07E-18	7.07E-18	5.64E-17
F25	7.07E-18	7.07E-18	7.07E-18	7.07E-18	1.42E-06	7.07E-18	7.07E-18	2.35E-04	7.07E-18	5.24E-15
F26	7.07E-18	2.60E-08	7.07E-18	7.07E-18	6.12E-10	7.07E-18	5.04E-16	1.39E-14	2.45E-06	9.02E-04
F27	7.07E-18	8.48E-17	7.07E-18	7.07E-18	9.92E-16	7.07E-18	7.07E-18	5.98E-17	1.94E-15	6.80E-11
F28	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.50E-18	7.07E-18	7.07E-18	1.10E-13	7.07E-18	7.07E-18
F29	7.07E-18	6.31E-13	7.07E-18	7.07E-18	3.65E-14	7.07E-18	1.95E-17	1.27E-16	2.69E-16	4.36E-03
F30	7.07E-18									

Table 7

Wilcoxon rank sum test results from TTAO and 10 comparison algorithms on 50 dimensional CEC2017 functions.

	BWO	SCSO	SAO	RSA	AOS	AOA	ChOA	HHO	PO	RECAA
F1	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F3	7.07E-18	2.05E-09	7.07E-18	7.07E-18	1.13E-16	7.07E-18	7.07E-18	1.34E-01	7.07E-18	8.46E-18
F4	7.07E-18	7.07E-18	7.07E-18	7.07E-18	8.94E-14	7.07E-18	7.07E-18	1.48E-12	7.07E-18	7.07E-18
F5	7.07E-18	8.19E-12	7.07E-18	7.07E-18	3.95E-01	7.07E-18	7.07E-18	2.21E-08	7.07E-18	1.34E-12
F6	7.07E-18	3.97E-17	7.07E-18	7.07E-18	6.53E-14	7.07E-18	1.01E-17	1.95E-17	7.07E-18	2.00E-06
F7	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	1.21E-17
F8	7.07E-18	4.51E-14	7.07E-18	7.07E-18	3.91E-03	7.07E-18	7.50E-18	9.37E-11	1.08E-17	5.24E-15
F9	7.07E-18	2.29E-04	7.07E-18	7.07E-18	6.85E-06	1.54E-17	7.97E-18	2.62E-17	6.72E-17	1.54E-10
F10	7.07E-18	2.99E-01	7.07E-18	7.07E-18	6.86E-05	7.07E-18	7.07E-18	6.93E-02	2.42E-12	2.47E-17
F11	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	1.01E-17	7.07E-18	7.07E-18
F12	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F13	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F14	7.07E-18	4.28E-14	7.07E-18	7.07E-18	1.22E-13	7.07E-18	7.07E-18	5.32E-17	7.07E-18	3.56E-08
F15	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F16	7.07E-18	8.11E-13	7.07E-18	7.07E-18	4.20E-13	7.07E-18	7.07E-18	2.31E-12	7.07E-18	5.14E-10
F17	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23
F18	7.07E-18	1.50E-13	7.07E-18	7.07E-18	9.01E-12	7.07E-18	7.07E-18	8.97E-13	7.07E-18	9.48E-04
F19	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F20	7.07E-18	6.20E-11	7.07E-18	7.07E-18	1.11E-01	9.92E-16	7.07E-18	1.68E-11	7.07E-18	4.46E-01
F21	7.07E-18	5.32E-17	7.07E-18	7.07E-18	3.11E-14	7.07E-18	7.97E-18	7.07E-18	7.07E-18	1.73E-17
F22	7.07E-18	6.62E-02	7.07E-18	7.07E-18	2.17E-04	7.07E-18	7.07E-18	1.18E-01	5.02E-17	2.23E-04
F23	7.07E-18	5.97E-16	7.07E-18	7.07E-18	2.95E-17	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.68E-15
F24	7.07E-18	2.47E-17	7.07E-18	7.07E-18	7.97E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	4.21E-17
F25	7.07E-18	7.07E-18	7.07E-18	7.07E-18	2.27E-13	7.07E-18	7.07E-18	1.15E-11	7.07E-18	7.07E-18
F26	7.07E-18	1.36E-04	7.07E-18	7.07E-18	2.14E-06	7.07E-18	5.24E-15	7.12E-11	3.69E-09	3.61E-10
F27	7.07E-18	8.99E-17	7.07E-18	7.07E-18	1.91E-16	7.07E-18	7.07E-18	5.02E-17	9.53E-17	1.62E-05
F28	7.07E-18	7.07E-18	7.07E-18	7.07E-18	2.65E-14	7.07E-18	7.07E-18	2.20E-17	7.07E-18	7.07E-18
F29	7.07E-18	3.02E-16	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	6.53E-14	9.54E-18	7.15E-09
F30	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18

where β and γ are usually 1 or 2. The classical engineering applications do not involve equality constraints. Therefore, just consider the inequality constraints, and the parameters β and r in the penalty function are set to 2 and 10^{10} , respectively.

Then, the solution to the unconstrained problem will converge to the solution of the original problem. Meanwhile, the experimental environment of this section is the same as that of Section 4.

5.1. Speed reducer design problem

The reducer is an important part for the gearbox, whose function is to run the engine and propeller, seen Fig. 11. The weight of the reducer is minimized under 11 constraints for this problem. In total 7 variables determine this problem, which are the face width b , the teeth module m , the number of teeth of the pinion p , the length of the first shaft between

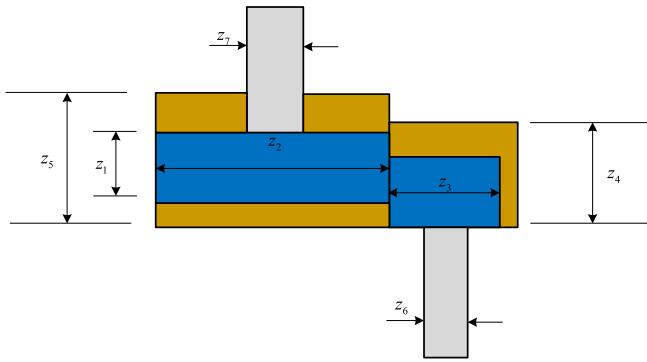
Table 8

Wilcoxon rank sum test results from TTAO and 10 comparison algorithms on 100 dimensional CEC2017 functions.

	BWO	SCSO	SAO	RSA	AOS	AOA	ChOA	HHO	PO	RECAA
F1	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F3	6.80E-11	2.47E-17	8.32E-06	5.29E-14	3.19E-16	7.34E-13	1.61E-08	1.21E-17	2.23E-09	3.70E-02
F4	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.97E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F5	7.07E-18	3.07E-02	7.07E-18	7.07E-18	1.56E-04	7.07E-18	7.07E-18	4.97E-03	7.07E-18	1.29E-13
F6	7.07E-18	5.40E-11	7.07E-18	7.07E-18	1.64E-07	7.07E-18	7.07E-18	2.29E-15	7.07E-18	5.07E-02
F7	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	9.54E-18
F8	7.07E-18	3.84E-09	7.07E-18	7.07E-18	1.53E-01	7.07E-18	7.07E-18	3.39E-11	7.07E-18	6.43E-12
F9	7.50E-18	4.46E-17	7.07E-18	7.97E-18	1.49E-08	1.41E-12	1.51E-16	1.00E-01	4.27E-03	1.49E-01
F10	7.07E-18	2.56E-15	7.07E-18	7.07E-18	1.32E-06	7.07E-18	7.07E-18	3.27E-05	6.69E-16	1.54E-17
F11	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	1.14E-17	7.07E-18	7.07E-18
F12	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F13	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F14	7.07E-18	7.07E-18	7.07E-18	7.07E-18	5.02E-17	7.07E-18	7.07E-18	1.21E-17	7.07E-18	1.54E-17
F15	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F16	7.07E-18	3.57E-15	7.07E-18	7.07E-18	1.39E-14	7.07E-18	7.07E-18	6.86E-09	7.07E-18	3.97E-17
F17	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23
F18	7.07E-18	1.58E-13	7.07E-18	7.07E-18	3.39E-11	7.07E-18	7.07E-18	2.95E-12	7.07E-18	4.42E-13
F19	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F20	7.07E-18	4.12E-10	7.07E-18	7.07E-18	7.85E-01	1.21E-17	7.07E-18	6.80E-11	7.97E-18	6.13E-12
F21	7.07E-18	3.53E-17	7.07E-18	7.07E-18	1.45E-17	7.07E-18	7.07E-18	7.07E-18	7.07E-18	9.53E-17
F22	7.07E-18	3.24E-11	7.07E-18	7.07E-18	1.76E-11	7.07E-18	7.07E-18	8.77E-09	1.95E-17	1.23E-09
F23	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F24	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F25	7.07E-18	7.07E-18	7.07E-18	7.07E-18	5.33E-16	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F26	7.07E-18	7.64E-14	7.07E-18	7.07E-18	1.60E-16	7.07E-18	5.64E-17	2.27E-16	1.37E-17	6.03E-02
F27	7.07E-18	4.21E-15	7.07E-18	7.07E-18	6.53E-14	7.07E-18	7.07E-18	2.33E-11	2.27E-13	4.83E-02
F28	7.07E-18	7.07E-18	7.07E-18	7.07E-18	3.02E-16	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F29	7.07E-18	9.54E-18	7.07E-18	7.07E-18	7.50E-18	7.07E-18	7.07E-18	3.32E-17	7.07E-18	2.20E-17
F30	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18

Table 9Description of engineering application problems (*Dim* represents the number of variables; *Nc* is the number of constraints; *N* is population size).

No	Name	Dim	Nc	Objective	N
1	Speed reducer problem	7	11	Min	50
2	Tension/Compression Spring Design	3	4	Min	50
3	Three-Bar Truss Design Problem	2	3	Min	50
4	Welded Beam Design	4	7	Min	50
5	Pressure Vessel	4	4	Min	50
6	Tubular Column Design	2	6	Min	30
7	Piston lever	4	3	Min	30
8	Design of gear train	4	0	Min	30

**Fig. 11.** Speed reducer problem.

bearings l_1 , the length of the second shaft between bearings l_2 , the first bearing diameter shafts d_1 , and the second bearing diameter shafts d_2 . The mathematical expression of this problem is defined as follows.

$$\vec{z} = [z_1 z_2 z_3 z_4 z_5 z_6 z_7] = [bmpl_1 l_2 d_1 d_2]$$

$$\min(\vec{z}) = 0.7854z_1^2(3.333z_3^2 + 14.9334z_3 - 43.0934) \\ - 1.508z_1(z_6^2 + z_7^2) + 7.4777(z_6^3 + z_7^3) + 0.7854(z_4z_6^2 + z_5z_7^2) \quad s.t.$$

$$g_1(\vec{z}) = \frac{27}{z_1 z_2^2 z_3} - 1 \leq 0$$

$$g_2(\vec{z}) = \frac{397.5}{z_1 z_2^2 z_3^2} - 1 \leq 0$$

$$g_3(\vec{z}) = \frac{1.93z_4^3}{z_2 z_6^4 z_3} - 1 \leq 0$$

$$g_4(\vec{z}) = \frac{1.93z_5^3}{z_2 z_7^4 z_3} - 1 \leq 0$$

$$g_5(\vec{z}) = \frac{\sqrt{(745(z_4/z_2 z_3))^2 + 16.9 \times 10^6}}{110 z_6^3} - 1 \leq 0$$

$$g_6(\vec{z}) = \frac{\sqrt{(745(z_5/z_2 z_3))^2 + 157.5 \times 10^6}}{85 z_7^3} - 1 \leq 0$$

$$g_7(\vec{z}) = \frac{z_2 z_3}{40} - 1 \leq 0$$

$$g_8(\vec{z}) = \frac{5z_2}{z_1} - 1 \leq 0$$

$$g_9(\vec{z}) = \frac{z_1}{12z_2} - 1 \leq 0$$

$$g_{10}(\vec{z}) = \frac{1.5z_6 + 1.9}{z_4} - 1 \leq 0$$

$$g_{11}(\vec{z}) = \frac{1.1z_7 + 1.9}{z_5} - 1 \leq 0$$

$$2.6 \leq z_1 \leq 3.6, 0.7 \leq z_2 \leq 0.8, 17 \leq z_3 \leq 287, 3 \leq z_4 \leq 8.3$$

$$7.3 \leq z_5 \leq 8.3, 2.9 \leq z_6 \leq 3.95, 0 \leq z_7 \leq 5.5$$

Note: NA indicates that relevant statistical indicators are not given in original literatures.

The TTAO algorithm is compared with the SC (Ray & Liew, 2003), HHA-ACT (Wang, Cai, Zhou, & Fan, 2009), PSO-DE (Liu, Cai, & Wang, 2010), $(\mu + \lambda)$ ES (Mezura-Montes & Coello, 2005), CS (Gandomi, Yang, & Alavi, 2013), SHO (Dhiman & Kumar, 2017), MRFO (Zhao, Zhang, & Wang, 2020), EO (Hashim et al., 2021), AOA (Hashim et al., 2021), AO (Abualigah et al., 2021) and CSA (Feng, Niu, & Liu, 2021) algorithms. Table 10 shows the results that all algorithms achieve the optimal value. From Table 10, the TTAO algorithm outperforms previously proposed algorithms on this engineering problem. Table 11 records statistical results of all the algorithms. According to Table 11, the TTAO algorithm can achieve the best Mean, Worst and Std deviation results with lower evolution times. It is further proved that the TTAO algorithm has good applicability and high efficiency on this problem.

5.2. Tension/compression spring design problem

Minimizing the weight of the spring is the primary aim of the tension/compression spring design problem. Fig. 12 is a schematic of the problem. This problem is constrained by minimum deflection, shear stress, surge frequency, and outer diameter limit. Three decision variables are wire diameter d , mean coil diameter D , the number of active coils N . The mathematical description of this problem is as follows.

$$\vec{z} = [z_1 z_2 z_3] = [dDN]$$

$$\min f(\vec{z}) = (z_3 + 2)z_2 z_1^2$$

s.t.

$$g_1(\vec{z}) = 1 - \frac{z_2^2 z_3}{71785 z_1^4} \leq 0$$

$$g_2(\vec{z}) = \frac{4z_2^2 - z_1 z_2}{12566(z_2 z_1^3 - z_1^4)} + \frac{1}{5108 z_1^2} \leq 0$$

$$g_3(\vec{z}) = 1 - \frac{140.45 z_1}{z_2^2 z_3} \leq 0$$

$$g_4(\vec{z}) = \frac{z_1 + z_2}{1.5} - 1 \leq 0$$

$$0.05 \leq z_1 \leq 2.00, 2.25 \leq z_2 \leq 1.32, 0 \leq z_3 \leq 15.0$$

The GA2 (Coello & Montes, 2002), CA (Coello Coello & Becerra, 2004), CPSO (He & Wang, 2007), $(\mu + \lambda)$ ES (Mezura-Montes & Coello, 2005), SHO (Dhiman & Kumar, 2017), EO (Hashim et al., 2021), MRFO (Zhao et al., 2020), AOA (Hashim et al., 2021), BWO (Zhong et al., 2022)

algorithms have solved this problem. We compare these algorithms with the proposed TTAO algorithm. Tables 12 and 13 show optimal results and statistical results obtained by all the algorithms on this problem. The Best result of the TTAO ranks first among these algorithms. The evolutionary efficiency of the TTAO algorithm is higher than that of classical algorithms GA2, CA and CPSO. On the other hand, the TTAO algorithm can also outperform the novel proposed MRFO, AOA and BWO algorithms in recent years. Besides, the Mean, Worst and Std indicators of the TTAO algorithm are second only to SHO which has evolved 30,000 times. In general, the TTAO algorithm provides a strong competitiveness for this problem.

5.3. Three-bar truss design problem

The three-bar truss design problem is mainly to optimize the weight of the three-bar truss. Fig. 13 displays the planar construction of a three-bar truss. This problem is controlled by the cross-sectional areas of the two trusses (A_1, A_2). The mathematical formulation of the engineering problem is

$$\vec{z} = [z_1 z_2] = [A_1 A_2]$$

$$\min f(\vec{z}) = (2\sqrt{2}z_1 + z_2) \times l$$

s.t.

$$g_1(\vec{z}) = \frac{\sqrt{2}z_1 + z_2}{\sqrt{2z_1^2 + 2z_1 z_2}} p - \sigma \leq 0$$

$$g_2(\vec{z}) = \frac{z_2}{\sqrt{2z_1^2 + 2z_1 z_2}} p - \sigma \leq 0$$

$$g_3(\vec{z}) = \frac{1}{\sqrt{2z_2 + z_1}} p - \sigma \leq 0$$

$$l = 100\text{cm}, p = 2 \frac{kN}{cm^2}, \sigma = 2 \frac{kN}{cm^2}$$

$$0 \leq z_1, z_2 \leq 1$$

Taking GA1 (Kaveh & Eslamloo, 2020), PSO (Kaveh & Eslamloo, 2020), GWO (Kaveh & Eslamloo, 2020), MFO (Kaveh & Eslamloo, 2020), WSA (Kaveh & Eslamloo, 2020) and CSA (Feng et al., 2021) as the comparison algorithms, the effectiveness of the TTAO algorithm for solving three-bar truss design problem is verified. The optimal results and statistical results obtained by all the algorithms on this problem are shown in Tables 14 and 15. Combined with the analysis of the two tables, it can be seen that the TTAO algorithm can achieve more suitable results under the premise of less evolution times (25,000). However, other algorithms have to evolve 50,000 times to get comparable results to the TTAO algorithm. Furthermore, the Mean indicator of the TTAO algorithm also ranks first among these algorithms. In a nutshell, the

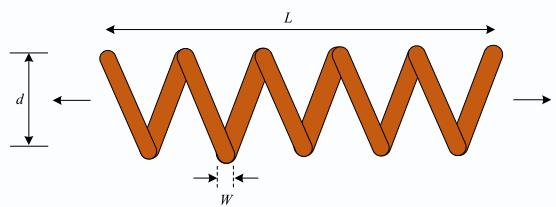
Table 10
Optimal results from different algorithms on speed reducer problem.

Algorithms	z_1	z_2	z_3	z_4	z_5	z_6	z_7	f_{best}
SC(Ray & Liew, 2003)	3.50000	0.70000	17	7.327602	7.715321	3.350267	5.286655	2994.7442410
HHA-ACT(Wang et al., 2009)	3.50002	0.70000	17.000013	7.300428	7.715377	3.350231	5.286664	2994.499107
PSO-DE(Liu et al., 2010)	3.50000	0.70000	17	7.300000	7.800000	3.350215	5.286683	2996.3481670
$(\mu + \lambda)$ ES(Mezura-Montes & Coello, 2005)	3.499999	0.699999	17	7.300000	7.800000	3.350215	5.286683	2996.348094
CS(Gandomi et al., 2013)	3.5015	0.7000	17.0000	7.6050	7.8181	3.3520	5.2875	3000.9810
SHO(Dhiman & Kumar, 2017)	3.50159	0.7	17	7.3	7.8	3.35127	5.28874	2998.5507
MRFO(Zhao et al., 2020)	3.5000000	0.7000000	17.0000000	7.3000000	7.7153199	3.3502147	5.2866545	2994.4710667
EO(Hashim et al., 2021)	3.4976	0.7	17	7.3	7.8	3.3501	5.2857	3.00E+03
AOA(Hashim et al., 2021)	3.4976	0.7	17	7.3	7.8	3.3501	5.2857	3.00E+03
AO(Abualigah et al., 2021)	3.5021	0.7000	17.0000	7.3099	7.7476	3.3641	5.2994	3007.7328
CSA(Feng, Niu, & Liu, 2021)	3.5	0.7	17	7.3	7.8	3.35021467	5.28668323	2996.348165
TTAO	3.5	0.7	17	7.3	7.7153199	3.3502147	5.2866545	2994.4710661

Table 11

Statistical results from different algorithms on speed reducer problem.

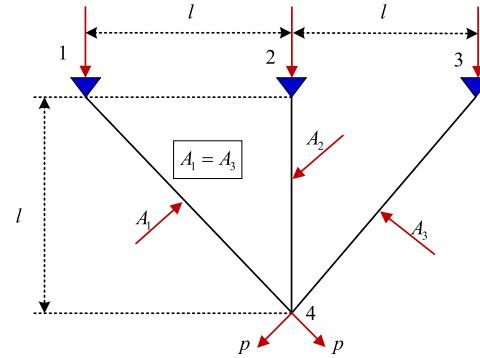
Algorithms	Mean	Worst	Best	Std	Evals
SC(Ray & Liew, 2003)	3001.7582640	3009.9647360	2994.7442410	4	54,456
HHA-ACT(Wang et al., 2009)	2994.6133680	2994.7523110	2994.4991070	7.0E-02	40,000
PSO-DE(Wang et al., 2009)	2996.3481740	2996.3482040	2996.3481670	6.4000E-06	54,350
($\mu + \lambda$)ES(Mezura-Montes & Coello, 2005)	2996.348094	2996.348094	2996.348094	0	30,000
CSC(Gandomi et al., 2013)	3007.1997	3009.0	3007.1997	4.9634E+00	50,000
SHO(Gandomi et al., 2013)	2999.640	3003.889	2998.5507	1.93193E+00	30,000
MRFO(Gandomi et al., 2013)	2994.4710662	2994.4710663	2994.4710667	3.6508E-08	30,000
EO(Hashim et al., 2021)	3.00E+03	3.00E+03	3.00E+03	1.37E-12	30,000
AOA(Hashim et al., 2021)	3.00E+03	3.00E+03	3.00E+03	1.22E-12	30,000
AO(Abuligah et al., 2021)	NA	NA	3007.7328	NA	50,000
CSA(Feng, Niu, & Liu, 2021)	2996.348164968529	2996.348164968505	2996.34816496853	9.90E-13	50,000
TTAO	2994.4710661	2994.4710661	2994.4710661	9.59E-13	25,000

**Fig. 12.** Tension/compression spring design problem.

TTAO algorithm has good applicability in this problem.

5.4. Welded beam design problem

The goal of the welded beam design problem is to minimize the

**Fig. 13.** Three-bar truss design problem.**Table 12**

Optimal results from different algorithms on tension/compression spring design problem.

Algorithms	z_1	z_2	z_3	f_{best}
GA2(Coello & Montes, 2002)	0.051989	0.363965	10.890522	0.012681
CA(Coello Coello & Becerra, 2004)	0.050000	0.317395	14.031795	0.012721
CPSO(He & Wang, 2007)	0.051728	0.357644	11.244543	0.0126747
($\mu + \lambda$)ES(Mezura-Montes & Coello, 2005)	0.052836	0.384942	9.807729	0.012689
SHO(Dhiman & Kumar, 2017)	0.051144	0.343751	12.0955	0.012674000
EO(Hashim et al., 2021)	0.0512	0.3445	12.0455	0.012682
MRFO(Zhao et al., 2020)	0.0523734	0.3733461	10.3831265	0.0126813
AOA(Hashim et al., 2021)	0.0508	0.3348	11.7020	0.012681
BWO(Zhong et al., 2022)	0.0517	0.3568	11.3132	0.012703
TTAO	0.051674	0.356352	11.31044	0.012665

Table 14

Optimal results from different algorithms on three-bar truss design problem.

Algorithms	z_1	z_2	f_{best}
GAI(Kaveh & Eslamloo, 2020)	0.788915	0.407569	263.8958857
PSO(Kaveh & Eslamloo, 2020)	0.788669	0.408265	263.89584341
GWO(Kaveh & Eslamloo, 2020)	0.788648	0.408325	263.89600631
MFO(Kaveh & Eslamloo, 2020)	0.788601	0.408458	263.89584742
WSA(Kaveh & Eslamloo, 2020)	0.788683	0.408227	263.89584340
CSA(Feng et al., 2021)	0.788638976	0.408350573	263.895844337
TTAO	0.788688	0.408213	263.8958431

Table 13

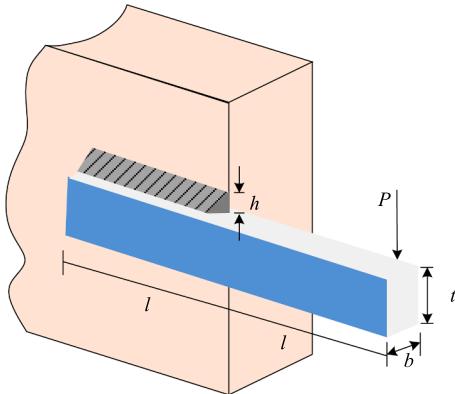
Statistical results from different algorithms on tension/compression spring design problem.

Algorithms	Mean	Worst	Best	Std	Evals
GA2(Coello & Montes, 2002)	0.012742	0.012973	0.012681	9.5E-05	80,000
CA(Coello Coello & Becerra, 2004)	0.013568	0.0151156	0.012721	8.4E-04	50,000
CPSO(He & Wang, 2007)	0.0127300	0.0129240	0.0126747	5.1985E-04	200,000
($\mu + \lambda$)ES(Mezura-Montes & Coello, 2005)	0.013165	0.014078	0.012689	3.9E-04	30,000
SHO(Dhiman & Kumar, 2017)	0.012684106	0.012715185	0.012674000	2.7E-05	30,000
EO(Hashim et al., 2021)	0.013536	0.015711	0.012682	2.20E-04	30,000
MRFO(Zhao et al., 2020)	0.0127007	0.0131811	0.0126757	2.1378E-04	50,000
AOA(Hashim et al., 2021)	0.013369	0.015625	0.012681	7.44E-04	30,000
BWO(Zhong et al., 2022)	NA	NA	0.012703	NA	50,000
TTAO	0.0126967	0.0128470	0.012665	3.22E-05	25,000

Table 15

Statistical results from different algorithms on three-bar truss design problem.

Algorithms	Mean	Worst	Best	Std	Evals
GA1(Kaveh & Eslamou, 2020)	263.96803663	264.82080546	263.89588573	1.66862E-01	50,000
PSO(Kaveh & Eslamou, 2020)	263.95741428	264.58490296	263.89584341	1.3689749E-01	50,000
GWO(Kaveh & Eslamou, 2020)	263.89795501	263.90421778	263.89600631	1.61422E-03	50,000
MFO(Kaveh & Eslamou, 2020)	263.91806578	264.16411946	263.89584742	4.703090E-02	50,000
WSA(Kaveh & Eslamou, 2020)	263.89606687	263.89743217	263.89584340	3.1196E-04	50,000
CSA(Feng et al., 2021)	NA	NA	263.895844337	NA	50,000
TTAO	263.8960254	263.8981915	263.8958431	3.56E-04	25,000

**Fig. 14.** Welded beam design problem.

manufacturing cost of welded beams, which is illustrated in Fig. 14. This problem has seven constraints for this problem, which is affected by shear stress, bending stress, buckling load and end deflection. The thickness of the weld h , the length of the clamping bar l , the height of the bar t and the thickness of the bar b are the decision variables of this problem. The mathematical description of the problem is

$$\vec{z} = [z_1 z_2 z_3 z_4] = [hltb]$$

$$\min f(\vec{z}) = 1.1047z_1^2 z_2 + 0.04811z_3 z_4(14.0 + z_2)$$

s.t.

$$g_1(\vec{z}) = \tau(\vec{z}) - 13600 \leq 0$$

$$g_2(\vec{z}) = \sigma(\vec{z}) - 30000 \leq 0$$

$$g_3(\vec{z}) = \delta(\vec{z}) - 0.25 \leq 0$$

$$g_4(\vec{z}) = z_1 - z_4 \leq 0$$

$$g_5(\vec{z}) = 6000 - P_c(\vec{z}) \leq 0$$

$$g_6(\vec{z}) = 0.125 - z_1 \leq 0$$

$$g_7(\vec{z}) = 1.10471z_1^2 + 0.04811z_3 z_4(14.0 + z_2) - 5.0 \leq 0$$

$$\tau(\vec{z}) = \sqrt{(z'_1)^2 + (z''_1)^2 + \frac{(lt'z''_1)}{\sqrt{0.25(l^2 + (h+t)^2)}}}$$

$$z'_1 = \frac{6000}{\sqrt{2hl}}, \sigma(\vec{z}) = \frac{504000}{t^2 b}, \delta(\vec{z}) = \frac{65856000}{(30 \times 10^6)bt^3}$$

$$z''_1 = \frac{6000(14 + 0.5l)\sqrt{0.25(l^2 + (h+t)^2)}}{2[0.707hl\left(\frac{l}{12} + 0.25(h+t)^2\right)]}$$

$$P_c(\vec{z}) = 64746.022(1 - 0.0282346t)tb^3$$

$$0.1 \leq z_1 \leq 2, 0.1 \leq z_2 \leq 10, 0.1 \leq z_3 \leq 10, 0.1 \leq z_4 \leq 2$$

The SC (Ray & Liew, 2003), CPSO (He & Wang, 2007), GA2 (Coello & Montes, 2002), HHA-ACT (Wang et al., 2009), SHO (Dhiman & Kumar, 2017), HGSO (Hashim et al., 2019), EO (Hashim et al., 2021), MRFO (Zhao et al., 2020), MPA (Faramarzi, Heidarnejad, Mirjalili, & Gandomi, 2020), LFD (Houssein, Saad, Hashim, Shaban, & Hassaballah, 2020), AOA (Hashim et al., 2021) and AOS (Azizi, 2021) algorithms are used for comparison. The optimal results and statistical results from all algorithms on welded beam design problem are listed in Tables 16 and 17. From these results, it can be clearly seen that although both the TTAO algorithm and MRFO algorithm get better results, the TTAO algorithm uses fewer iterations. The Mean, Worst and Std Deviation indicators of the TTAO algorithm also manifest obvious advantages in these comparison algorithms. Overall, the TTAO algorithm has excellent optimization capacity in solving this problem.

5.5. Pressure vessel design

Fig. 15 describes pressure vessel and the optimization objective of this problem is to minimize the total cost. There are four decision variables in this problem, which are the thickness of shell T_s , the thickness of head T_h , the inner radius R and the length of cylinder L . The corresponding mathematical description of the problem is as follows.

$$\vec{z} = [z_1 z_2 z_3 z_4] = [T_s T_h RL]$$

$$\min f(\vec{z}) = 0.6224z_1 z_3 z_4 + 1.7781z_2 z_3^2 + 3.1661z_1^2 z_4 + 19.84z_1^2 z_3$$

s.t.

$$g_1(\vec{z}) = -z_1 + 0.0193z_3 \leq 0$$

$$g_2(\vec{z}) = -z_3 + 0.00954z_3 \leq 0$$

$$g_3(\vec{z}) = -\pi z_3^2 z_4 - \frac{4}{3}\pi z_3^3 + 1296000 \leq 0$$

$$g_4(\vec{z}) = z_4 - 240 \leq 0$$

Tables 18 and 19 list the optimal results and statistical results from the proposed TTAO algorithm and the GA2 (Coello & Montes, 2002), CPSO (He & Wang, 2007), $(\mu + \lambda)$ ES (Mezura-Montes & Coello, 2005), EO (Hashim et al., 2021), MPA (Faramarzi et al., 2020), LFD (Houssein et al., 2020), AO (Abualigah et al., 2021) and BWO (Zhong et al., 2022) algorithms. The TTAO algorithm is obviously better than the GA2, CPSO, AO, BWO algorithm with more evolutionary times and MPA algorithm with same evolutionary times. In addition, other indicators of the TTAO algorithm ranked second among these algorithms. Consequently, the results demonstrate that the TTAO algorithm has better applicability for this problem and can surpass the comparison algorithms to obtain a more suitable solution.

Table 16

Optimal results from different algorithms on welded beam design problem.

Algorithms	z_1	z_2	z_3	z_4	f_{best}
SC(Ray & Liew, 2003)	0.244438276	6.237967234	8.288576143	0.244566182	2.3854347
CPSO(He & Wang, 2007)	0.202369	3.544214	9.048210	0.205723	1.728024
GA2(Coello & Montes, 2002)	0.205986	3.471328	9.020224	0.206480	1.728226
HHA-ACT(Wang et al., 2009)	0.2443688943	6.2175179741	8.2914773014	0.2443689510	2.38095723
SHO(Dhiman & Kumar, 2017)	0.205563	3.474846	9.035799	0.205811	1.72566
HGSO(Hashim et al., 2019)	0.2054	3.4476	9.0269	0.2060	1.7260
EO(Hashim et al., 2021)	0.2057	3.4705	9.0366	0.2057	1.7449
MRFO(Zhao et al., 2020)	0.2057296	3.4704887	9.0366239	0.2057296	1.7248523
MPA(Faramarzi et al., 2020)	0.205728	3.470509	9.036624	0.205730	1.724853
LFD(Houssein et al., 2020)	0.1857	3.9070	9.1552	0.2051	1.77E+00
AOA(Hashim et al., 2021)	0.2057	3.4705	9.0366	0.2057	1.7249
AOS(Azizi, 2021)	0.012665233	0.051689535	0.356729145	11.288297130	1.724852309
TTAO	0.20573	3.470489	9.036624	0.20573	1.7248523

Table 17

Statistical results from different algorithms on welded beam design problem.

Algorithms	Mean	Worst	Best	Std	Evals
SC(Ray & Liew, 2003)	3.0025883	6.3996780	2.3854347	9.600E-01	33,095
CPSO(He & Wang, 2007)	1.748831	1.782143	1.728024	1.2926E-02	240,000
GA2(Coello & Montes, 2002)	1.792654	1.993408	1.728226	7.4713E-02	80,000
HHA-ACT(Wang et al., 2009)	2.38097103	2.38102095	2.38095723	1.3E-05	30,000
SHO(Dhiman & Kumar, 2017)	1.725828	1.726064	1.725661	2.87E-04	30,000
HGSO(Hashim et al., 2019)	1.7265	1.7325	1.7260	7.66E-03	30,000
EO(Hashim et al., 2021)	1.7555	1.8849	1.7449	1.86E-03	30,000
MRFO(Zhao et al., 2020)	1.7248547	1.7248648	1.7248523	3.8320E-06	30,000
MPA(Faramarzi et al., 2020)	1.724861	1.724873	1.724853	6.41E-06	25,000
LFD(Houssein et al., 2020)	2.30E+00	3.04E+00	1.77E+00	3.16E-01	35,000
AOA(Hashim et al., 2021)	1.7304	1.8716	1.7249	2.67E-02	30,000
AOS(Azizi, 2021)	1.725673538	1.732516334	1.724852309	2.4786984E-02	200,000
TTAO	1.7248524	1.7248534	1.7248523	2.24E-07	25,000

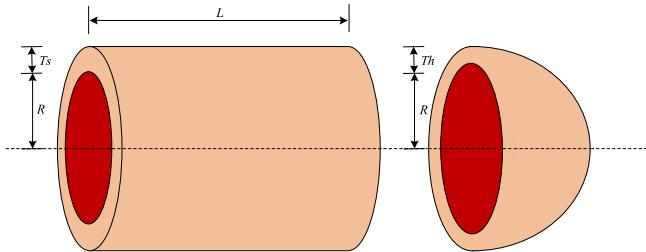


Fig. 15. Pressure vessel design problem.

5.6. Tubular column design problem

The tubular column design problem requires the design of a uniform column section to bear the compressive load, and the optimization goal is to minimize the cost of the tubular column, seen in Fig. 16. The problem is subject to six conditions. The decision variables are the average diameter d and average thickness t of the column. The

corresponding mathematical model is given as.

$$\vec{z} = [z_1 z_2] = [dt]$$

s.t.

$$\min f(\vec{z}) = 9.82z_1 z_2 + 2z_1$$

$$g_1(\vec{z}) = \frac{P}{\pi z_1 z_2 \sigma_y} - 1 \leq 0$$

$$g_2(\vec{z}) = \frac{8PL}{\pi^3 E dt(z_1^2 + z_2^2)} - 1 \leq 0$$

$$g_3(\vec{z}) = \frac{2}{z_1^2} - 1 \leq 0$$

$$g_4(\vec{z}) = \frac{z_1^2}{14} - 1 \leq 0$$

$$g_5(\vec{z}) = \frac{0.2}{z_2^2} - 1 \leq 0$$

Table 18

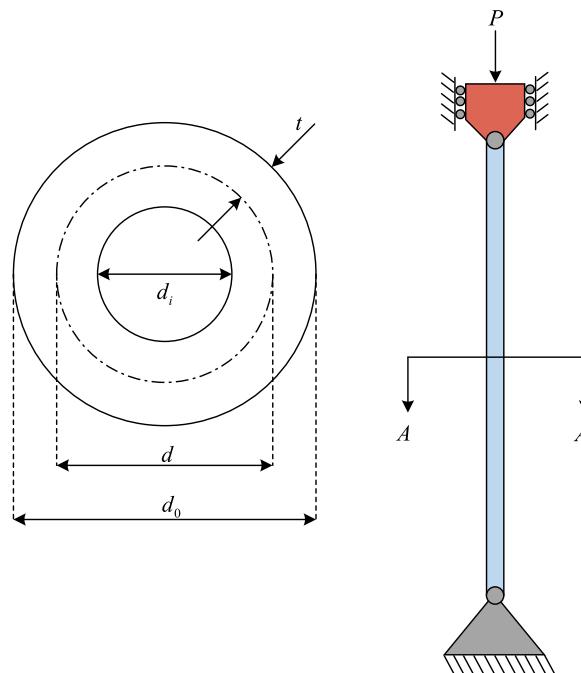
Optimal results from different algorithms on pressure vessel design problem.

Algorithms	z_1	z_2	z_3	z_4	f_{best}
GA2(Coello & Montes, 2002)	0.812500	0.437500	42.097398	176.654047	6059.946341
CPSO(He & Wang, 2007)	0.8125	0.4375	42.098445596	176.636595842	6059.714335048
(μ + λ)ES(Mezura-Montes & Coello, 2005)	0.8125	0.4375	42.098446	176.636596	6059.701610
EO(Hashim et al., 2021)	0.7929	0.3914	41.1773	188.3950	5.91E+03
MPA(Faramarzi et al., 2020)	0.8125	0.4375	42.098445	176.636607	6059.7144
LFD(Houssein et al., 2020)	0.8777	0.4339	45.4755	139.0654	6.08E+03
AO(Abualigah et al., 2021)	1.0540	0.182806	59.6219	38.8050	5949.2258
BWO(Zhong et al., 2022)	0.7796	0.3921	40.3598	199.4567	5912.114
TTAO	0.20573	3.470489	9.036624	0.20573	5907.851851

Table 19

Optimal results from different algorithms on pressure vessel design problem.

Algorithms	Mean	Worst	Best	Std	Evals
GA2(Coello & Montes, 2002)	6177.253268	6469.322010	6059.946341	1.30929702E+02	80,000
CPSO(He & Wang, 2007)	6059.714335	6059.714335	6059.714335	1.0E-10	42,100
($\mu + \lambda$)ES(Mezura-Montes & Coello, 2005)	6379.938037	6820.397461	6059.701610	2.1E+02	30,000
EO(Hashim et al., 2021)	6.53E+03	7.30E+03	5.91E+03	3.98E+02	30,000
LFD(Faramarzi et al., 2020)	1.60E+04	3.62E+04	6.08E+03	8.02E+03	35,000
MPA(Faramarzi et al., 2020)	6102.8271	6410.0929	6059.7144	1.0661E+02	25,000
AO(Abualigah et al., 2021)	NA	NA	5949.2258	NA	50,000
BWO(Zhong et al., 2022)	NA	NA	5912.114	NA	50,000
TTAO	6292.545138	6866.25039	5904.242695	2.17E+02	25,000

**Fig. 16.** Tubular column design problem.

$$g_6(\vec{z}) = \frac{z_1}{0.8} - 1 \leq 0$$

$$\sigma_y = 500 \text{ kgf/cm}^2, E = 0.85 \times 10^6 \text{ kgf/cm}^2, 2 \leq z_1 \leq 14, 0.2 \leq z_2 \leq 0.8 \\ P = 2500 \text{ kgf}, L = 250 \text{ cm}$$

On this problem, the CS (Gandomi et al., 2013), AOS (Azizi, Tahbari, & Giaralis, 2021) and CSA (Feng et al., 2021) algorithms are selected as the comparison algorithms. The optimal results and statistical results of the four algorithms on this problem are listed in Tables 20 and 21. Based on the analysis of the two tables, the TTAO algorithm gains better results than the CS algorithm with the same number of evolutions for all statistical indicators. For another, the TTAO algorithm converges to higher optimization accuracy in less evolution times than the AOS and CSA algorithms. Moreover, it can be seen from Mean and Std Deviation that the TTAO has good robustness.

Table 20

Optimal results from different algorithms on tubular column design problem.

Algorithms	z_1	z_2	f_{best}
CS(Gandomi et al., 2013)	5.45139	0.29196	26.53217
AOS(Azizi et al., 2021)	5.451152962	0.291966716	26.53137828
CSA(Feng et al., 2021)	5.451163397	0.291965509	26.531364472
TTAO	5.452181	0.291626	26.51816147

5.7. Piston lever design

The piston lever design problem is also one of the classical real-world optimization problems. As shown in Fig. 17, as the piston lever rises from 0 to $\pi/4$, the oil volume is minimized to determine the piston components H , B , D , and X . The mathematical expression of the problem can be written as

$$\vec{z} = [z_1 z_2 z_3 z_4] = [HBDX]$$

$$\min(\vec{z}) = \frac{1}{4} \pi z_3^2 (L_2 - L_1)$$

s.t.

$$g_1(\vec{z}) = QL\cos\theta - R \times F \leq 0$$

$$g_2(\vec{z}) = Q(L - z_4) - M_{max} \leq 0$$

$$g_3(\vec{z}) = 1.2(L_2 - L_1) - L_1 \leq 0$$

$$g_4(\vec{z}) = \frac{z_3}{2} - z_2 \leq 0$$

$$R = \frac{| -z_4(z_4\sin\theta + z_1) + z_1(z_2 - z_4\cos\theta) |}{\sqrt{(z_4 - z_2)^2 + z_1^2}}, F = \frac{\pi P z_3^2}{4}$$

$$L_1 = \sqrt{(z_4 - z_2)^2 + z_1^2}, \theta = \frac{\pi}{4}, Q = 10000 \text{ lbs}, L = 240 \text{ in} \\ L_2 = \sqrt{(z_4\sin\theta - z_1)^2 + (z_2 - z_4\cos\theta)^2}$$

$$M_{max} = 1.8 \times 10^6 \text{ lbsin}, P = 1500 \text{ psi}$$

$$0.05 \leq z_1, z_2, z_4 \leq 500, 0.05 \leq z_3 \leq 120$$

To prove the applicability of the TTAO algorithm on this problem, the GA (Gandomi et al., 2013), PSO (Gandomi et al., 2013), HPSO (Gandomi et al., 2013), DE (Gandomi et al., 2013), CS (Gandomi et al., 2013), SSA (Seyyedabbasi & Kiani, 2023) and AOS (Azizi et al., 2021) are used as comparison algorithms. Tables 22 and 23 record the results obtained by the different algorithms. The TTAO algorithm provides a suitable solution, which is more efficient than the GA, PSO, HPSO and DE algorithms with 50,000 evolution times. Hence, the TTAO algorithm can effectively solve piston lever design problem.

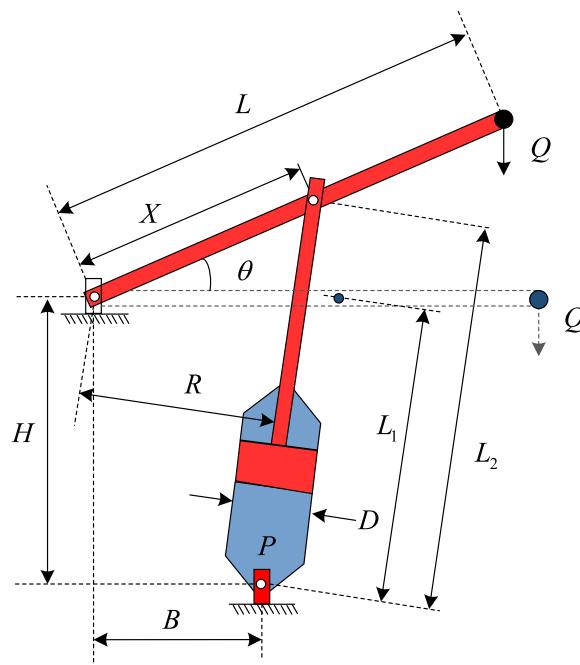
5.8. Gear train design problem

This problem is an unconstrained engineering application problem, as shown in Fig. 18. The design goal of this problem is to minimize the cost of the train gear ratio. Different from the other problems, decision variables of the gear train design problem are composed of four discrete integers, which are the number of teeth of the gear. The mathematical expression is described as.

Table 21

Statistical results from different algorithms on tubular column design problem.

Algorithms	Mean	Worst	Best	Std	Evals
CS(Gandomi et al., 2013)	26.53504	26.53972	26.53217	1.93E-02	15,000
AOS(Azizi et al., 2021)	26.53161399	26.60831361	26.53137828	1.0300E-03	100,000
CSA(Feng et al., 2021)	26.5315766	26.5319781	26.5313645	1.66E-04	50,000
TTAO	26.51816147	26.51816147	26.51816147	3.59E-15	15,000

**Fig. 17.** Piston lever design problem.

$$\min f(\vec{z}) = \left(\frac{1}{6.931} - \frac{z_3 z_2}{z_1 z_4} \right)^2$$

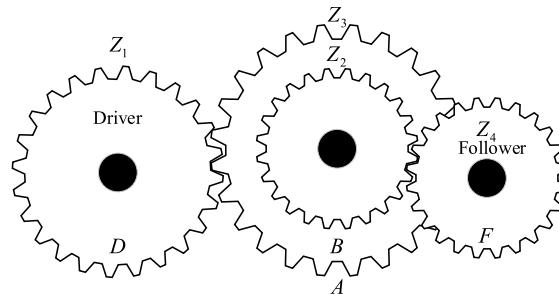
$$z_1, z_2, z_3, z_4 \in \{12, 13, 14, \dots, 60\}$$

The algorithms used to address this problem include the PSO (Kaveh

& Eslamloo, 2020), MFO (Kaveh & Eslamloo, 2020), BBO (Kaveh & Eslamloo, 2020), CBO (Kaveh & Eslamloo, 2020), WOA (Kaveh & Eslamloo, 2020), WSA (Kaveh & Eslamloo, 2020) and CSA (Feng et al., 2021) algorithms. By comparing the TTAO algorithm with these algorithms, the optimal results and statistical results obtained by all the algorithms are shown in Tables 24 and 25. From the optimal results, the TTAO, WOA, WSA and CSA algorithms can get the optimal value for this problem. Statistical results of TTAO are better than other comparison algorithms under the premise of using less evolution times. In summary, the TTAO algorithm also has certain advantages in solving this unconstrained engineering problem.

6. Conclusions and future work

To break the canonical evolutionary paradigm and comprehensively mine the positive information for complex optimization problems, a novel *meta-heuristic* algorithm, Triangulation Topology Aggregation Optimizer (TTAO), is proposed in this study, which is based on similar triangles in mathematics. In detail, vertexes in each triangular

**Fig. 18.** Gear train design problem.**Table 22**

Optimal results from different algorithms on piston lever design problem.

Algorithms	z_1	z_2	z_3	z_4	f_{best}
GA(Gandomi et al., 2013)	NA	NA	NA	NA	161
PSO(Gandomi et al., 2013)	NA	NA	NA	NA	122
HPSO(Gandomi et al., 2013)	NA	NA	NA	NA	162
DE(Gandomi et al., 2013)	NA	NA	NA	NA	159
CS(Gandomi et al., 2013)	0.0500	2.0430	120.000	4.0851	8.4271
SSA(Seyyedababasi & Kiani, 2023)	0.050	2.073	4.145	116.32	8.80243254
AOS(Azizi et al., 2021)	0.05	2.042112482	119.951727	4.084004492	8.419142742
TTAO	0.05	2.041514	4.083027	120	8.412698323

Table 23

Statistical results from different algorithms on piston lever design problem.

Algorithms	Mean	Worst	Best	Std	Evals
GA(Gandomi et al., 2013)	185	216	161	1.82E+01	50,000
PSO(Gandomi et al., 2013)	166	294	122	5.17E+01	50,000
HPSO(Gandomi et al., 2013)	187	197	162	1.34E+01	50,000
DE(Gandomi et al., 2013)	187	199	159	1.42E+01	50,000
CS(Gandomi et al., 2013)	40.2319	168.5920	8.4271	5.91E+01	50,000
SSA(Seyyedababasi & Kiani, 2023)	NA	NA	8.80243254	NA	15,000
AOS(Azizi et al., 2021)	33.7412759	60.66498628	8.419142742	9.35E+01	100,000
TTAO	123.1365161	167.4727301	8.412698323	7.18E+01	15,000

Table 24

Optimal results from different algorithms on gear train design problem.

Algorithms	z_1	z_2	z_3	z_4	f_{best}
PSO(Kaveh & Eslamou, 2020)	34	13	20	53	2.3078E-11
MFO(Kaveh & Eslamou, 2020)	51	30	13	53	2.3078E-11
BBO(Kaveh & Eslamou, 2020)	53	26	15	51	2.3078E-11
CBO(Kaveh & Eslamou, 2020)	53	13	20	34	2.3078E-11
WOA(Kaveh & Eslamou, 2020)	43	19	16	49	2.7009E-12
WSA(Kaveh & Eslamou, 2020)	43	16	19	49	2.7009E-12
CSA(Feng et al., 2021)	19	16	43	49	2.701E-12
TTAO	43	16	19	49	2.70E-12

Table 25

Statistical results from different algorithms on gear train design problem.

Algorithms	Mean	Worst	Best	Std	Evals
PSO(Kaveh & Eslamou, 2020)	7.9383E-08	1.0222E-06	2.3078E-11	1.8147E-07	50,000
MFO(Kaveh & Eslamou, 2020)	7.5337E-09	2.7265E-08	2.3078E-11	9.3539E-09	50,000
BBO(Kaveh & Eslamou, 2020)	4.5418E-08	4.2018E-07	2.3078E-11	7.2953E-08	50,000
CBO(Kaveh & Eslamou, 2020)	2.1032E-09	1.1173E-08	2.3078E-11	2.4025E-09	50,000
WOA(Kaveh & Eslamou, 2020)	9.6633E-10	6.5123E-09	2.7009E-12	1.1296E-09	50,000
WSA(Kaveh & Eslamou, 2020)	1.6800E-10	1.3616E-09	2.7009E-12	3.8265E-10	50,000
CSA(Feng et al., 2021)	NA	NA	2.701E-12	NA	50,000
TTAO	1.01E-10	9.92E-10	2.70E-12	2.52E-10	15,000

topological unit carried out generic aggregation and local aggregation during iterations, to emphasize the goals of global exploration and local exploitation, respectively. The optimization performance of the TTAO algorithm is evaluated on CEC2017 functions and 8 engineering problems.

Some conclusions are drawn from experimental results. Qualitative analysis clearly indicates the iterative searching process of the TTAO algorithm, which proves the effectiveness of the two strategies. On the 30-D CEC2017 functions, the TTAO shows superior convergence and stability. Moreover, the superior optimization performance of the TTAO algorithm is insensitive to the dimensional change of optimization problems, and it still obtains better convergence accuracy than other algorithms. In addition, the Wilcoxon rank sum test results further demonstrate the superior searching ability of the TTAO algorithm under statistical significance. For practical engineering problems, the TTAO algorithm achieves suitable solutions than comparison algorithms.

From the structure and experimental results of the TTAO algorithm, it is necessary to clearly explain the main findings of this article. (1) The properties of trigonometric topological similarity can be introduced into the meta-heuristic, thus a mathematic-based TTAO algorithm is developed. (2) Individual evolution based on triangular topological units manifests better avoidance of local extreme values than other competitors. (3) Generic aggregation and local aggregation form multiple similar triangular topological units to maintain effectively the balance between the exploration and exploitation. (4) The TTAO algorithm has few controlled parameters.

Future researches can further investigate the optimization capacity of similar triangles with different rotation angles, different side-length scales and non-equilateral structures in the same or different iteration. Some factors such as reverse learning mechanism, Cauchy mutation operator and chaotic map can be introduced into the TTAO algorithm to

further improve its searching ability. The TTAO algorithm's evolutionary optimization strategy based on triangular topological units can be combined with other algorithms to improve the performance. With the propose of wide applications in multi-objective, many multi-objective, dynamic multi-objective and classification problems, multi-objective and binary versions of the TTAO algorithm will be developed.

CRediT authorship contribution statement

Shijie Zhao: Conceptualization, Methodology, Writing – review & editing, Software, Investigation. **Tianran Zhang:** Methodology, Formal analysis, Writing – original draft, Writing – review & editing, Software, Visualization, Investigation. **Liang Cai:** Writing – review & editing, Supervision. **Ronghua Yang:** Writing – review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgement

This work was supported in part by the Natural Science Foundation of Liaoning Province (Grant No. 2023-MS-317), the China Postdoctoral Science Foundation (Grant No. 2021 M701537), and the Basic Research Foundation of Liaoning Educational Committee (Grant No. LJKMZ20220694).

References

- Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z. W., & Gandomi, A. H. (2022). Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications*, 191, Article 116158.
- Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The arithmetic optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*, 376, Article 113609.
- Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A., & Gandomi, A. H. (2021). Aquila optimizer: A novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, 157, Article 107250.
- Ahmadianfar, I., Bozorg-Haddad, O., & Chu, X. (2020). Gradient-based optimizer: A new metaheuristic optimization algorithm. *Information Sciences*, 540, 131–159.
- Ahmadianfar, I., Heidari, A. A., Gandomi, A. H., Chu, X., & Chen, H. (2021). RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Systems with Applications*, 181, Article 115079.
- Askari, Q., Younas, I., & Saeed, M. (2020). Political Optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowledge-based Systems*, 195, Article 105709.
- Awad, N. H., Ali, M. Z., Suganthan, P. N., Liang, J. J., & Qu, B. Y. (2017). Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization. *2017 IEEE congress on evolutionary computation (CEC)*.
- Azizi, M. (2021). Atomic orbital search: A novel metaheuristic algorithm. *Applied Mathematical Modelling*, 93, 657–683.
- Azizi, M., Talatahari, S., & Giaralis, A. (2021). Optimization of engineering design problems using atomic orbital search algorithm. *IEEE Access*, 9, 102497–102519.
- Bertsimas, D., & Tsitsiklis, J. (1993). Simulated annealing. *Statistical Science*, 8(1), 10–15.
- Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3), 268–308.
- Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, 237, 82–117.
- Braik, M., Hammouri, A., Atwan, J., Al-Betar, M. A., & Awadallah, M. A. (2022). White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowledge-Based Systems*, 243, Article 108457.
- Chou, J. S., & Nguyen, N. M. (2020). FBI inspired meta-optimization. *Applied Soft Computing*, 93, Article 106339.
- Coello Coello, C. A., & Becerra, R. L. (2004). Efficient evolutionary optimization through the use of a cultural algorithm. *Engineering Optimization*, 36(2), 219–236.
- Coello, C. A. C., & Montes, E. M. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 16(3), 193–203.

- Črepinský, M., Liu, S. H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM computing surveys (CSUR)*, 45(3), 1–33.
- Dhiman, G., & Kumar, V. (2017). Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Advances in Engineering Software*, 114, 48–70.
- Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1), 29–41.
- Esmaeili, H., Bidgoli, B. M., & Hakami, V. (2022). CMML: Combined metaheuristic-machine learning for adaptable routing in clustered wireless sensor networks. *Applied Soft Computing*, 118, Article 108477.
- Fadakar, E., & Ebrahimi, M. (2016, March). A new metaheuristic football game inspired algorithm. In *2016 1st conference on swarm intelligence and evolutionary computation (CSIEC)* (pp. 6–11). IEEE.
- Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications*, 152, Article 113377.
- Feng, Z. K., Niu, W. J., & Liu, S. (2021). Cooperation search algorithm: A novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems. *Applied Soft Computing*, 98, Article 106734.
- Forrest, S. (1996). Genetic algorithms. *ACM Computing Surveys (CSUR)*, 28(1), 77–80.
- Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2013). Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Engineering with Computers*, 29(1), 17–35.
- Gokalp, O., Tasci, E., & Ugur, A. (2020). A novel wrapper feature selection algorithm based on iterated greedy metaheuristic for sentiment classification. *Expert Systems with Applications*, 146, Article 113176.
- Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W., & Mirjalili, S. (2019). Henry gas solubility optimization: A novel physics-based algorithm. *Future Generation Computer Systems*, 101, 646–667.
- Hashim, F. A., Hussain, K., Houssein, E. H., Mabrouk, M. S., & Al-Atabany, W. (2021). Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems. *Applied Intelligence*, 51(3), 1531–1551.
- He, Q., & Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence*, 20(1), 89–99.
- Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, 97, 849–872.
- Houssein, E. H., Saad, M. R., Hashim, F. A., Shaban, H., & Hassaballah, M. (2020). Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 94, Article 103731.
- Hussain, K., Salleh, M. N. M., Cheng, S., & Shi, Y. (2019). On the exploration and exploitation in popular swarm-based metaheuristic algorithms. *Neural Computing and Applications*, 31(11), 7665–7683.
- Jamil, M., & Yang, X. S. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2), 150–194.
- Jiang, M., Wang, Z., Hong, H., & Yen, G. G. (2020). Knee point-based imbalanced transfer learning for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 25(1), 117–129.
- Jones, M. W., & Satherley, R. A. (2001). Shape representation using space filled sub-voxel distance fields. In *Proceedings international conference on shape modeling and applications* (pp. 316–325). IEEE.
- Kaveh, A., & Eslamlou, A. D. (2020, June). Water strider algorithm: A new metaheuristic and applications. In *Structures* (Vol. 25, pp. 520–541). Elsevier.
- Kaveh, A., Talatahari, S., & Khodadadi, N. (2020). Stochastic paint optimizer: Theory and application in civil engineering. *Engineering with Computers*, 1–32.
- Keivanian, F., & Chiong, R. (2022). A novel hybrid fuzzy-metaheuristic approach for multimodal single and multi-objective optimization problems. *Expert Systems with Applications*, 195, Article 116199.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (pp. 1942–1948). IEEE.
- Khishe, M., & Mosavi, M. R. (2020). Chimp optimization algorithm. *Expert Systems with Applications*, 149, Article 113338.
- Laczkovich, M. (2021). Irregular tilings of regular polygons with similar triangles. *Discrete & Computational Geometry*, 66(4), 1239–1261.
- Li, C., Chen, G., Liang, G., Luo, F., Zhao, J., & Dong, Z. Y. (2022). Integrated optimization algorithm: A metaheuristic approach for complicated optimization. *Information Sciences*, 586, 424–449.
- Liu, H., Cai, Z., & Wang, Y. (2010). Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing*, 10(2), 629–640.
- Mezura-Montes, E., & Coello, C. A. C. (2005). Useful infeasible solutions in engineering optimization with evolutionary algorithms. In *Mexican international conference on artificial intelligence* (pp. 652–662). Berlin, Heidelberg: Springer.
- Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826, 1989.
- Pierezan, J., & Coelho, L. D. S. (2018). Coyote optimization algorithm: A new metaheuristic for global optimization problems. In *2018 IEEE congress on evolutionary computation (cec)* (pp. 1–8).
- Ray, T., & Liew, K. M. (2003). Society and civilization: An optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation*, 7(4), 386–396.
- Rechenberg, I. (1973). Evolution Strategy: Optimization of Technical systems by means of biological evolution. *Fromman-Holzboog Stuttgart*, 104, 15–16.
- Reddy, P. V. N., Padmini, G. R., Govindaraj, P., & Sudhakar, M. S. (2022). Robust Feature Descriptor Employing Square Triangle Tessellation for Shape Retrieval. *Wireless Personal Communications*, 123(3), 2923–2936.
- Salawudeen, A. T., Mu'azu, M. B., Yusuf, A., & Adedokun, A. E. (2021). A Novel Smell Agent Optimization (SAO): An extensive CEC study and engineering application. *Knowledge-Based Systems*, 232, Article 107486.
- Seck-Tuoh-Mora, J. C., Lopez-Arias, O., Hernandez-Romero, N., Martínez, G. J., & Volpi-Leon, V. (2022). A New Algorithm Inspired on Reversible Elementary Cellular Automata for Global Optimization. *IEEE Access*, 10, 112211–112229.
- Seyyedababasi, A., & Kiani, F. (2023). Sand Cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. *Engineering with Computers*, 39(4), 2627–2651.
- Soifer, A. (2009). How Does One Cut a Triangle? I. In *How Does One Cut a Triangle?* (pp. 15–23). New York, NY: Springer.
- Storn, R., & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341–359.
- Wang, Y., Cai, Z., Zhou, Y., & Fan, Z. (2009). Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Structural and Multidisciplinary Optimization*, 37(4), 395–413.
- Wilcoxon, F. (1992). Individual comparisons by ranking methods. In *Breakthroughs in statistics* (pp. 196–202). New York, NY: Springer.
- Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on Evolutionary Computation*, 1(1), 67–82.
- Yadav, A. (2019). AEFA: Artificial electric field algorithm for global optimization. *Swarm and Evolutionary Computation*, 48, 93–108.
- Zhao, W., Wang, L., & Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering*, 388, Article 114194.
- Zhao, W., Zhang, Z., & Wang, L. (2020). Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Engineering Applications of Artificial Intelligence*, 87, Article 103300.
- Zhong, C., Li, G., & Meng, Z. (2022). Beluga whale optimization: A novel nature-inspired metaheuristic algorithm. *Knowledge-Based Systems*, 251, 109215.