

Full length article

Genghis Khan shark optimizer: A novel nature-inspired algorithm for engineering optimization

Gang Hu^{a,*}, Yuxuan Guo^a, Guo Wei^b, Laith Abualigah^{c,d,e,f,g}^a Department of Applied Mathematics, Xi'an University of Technology, Xi'an 710054, PR China^b University of North Carolina at Pembroke, Pembroke, NC 28372, USA^c Computer Science Department, Al al-Bayt University, Mafraq 25113, Jordan^d Department of Electrical and Computer Engineering, Lebanese American University, Byblos 13-5053, Lebanon^e Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan^f MEU Research Unit, Middle East University, Amman 11831, Jordan^g Applied science research center, Applied science private university, Amman 11931, Jordan

ARTICLE INFO

Keywords:
 Optimization
 Genghis Khan shark optimizer
 Meta-heuristic algorithm
 Exploration and exploitation
 Real-world constrained optimization problem

ABSTRACT

This study tenders a new nature-inspired metaheuristic algorithm (MA) based on the behavior of the Genghis Khan shark (GKS), called GKS optimizer (GKSO), which is used for numerical optimization and engineering design. The inspiration for GKSO comes from the predation and survival behavior of GKS, and the entire optimization process is achieved by simulating four different activities of GKS, including hunting (exploration), movement (exploitation), foraging (switch from exploration to exploitation), and self-protection mechanism. These operators are mimicked using various mathematical models to efficiently perform optimization tasks of agents in different regions of the search space. In an effort to validate this method's viability and superiority, an in-depth analysis of the proposed GKSO is carried out from both qualitative and quantitative perspectives. Qualitative analysis verifies that GKSO has good exploration and exploitation (ENE) capability. Simultaneously, GKSO is quantitatively analyzed with eight existing fish optimization algorithms and the other nine well-known MAs on CEC2019 and CEC2022, respectively. Among them, a series of experimental scenarios are conducted to validate the applicability and robustness of GKSO by exploring its performance for CEC2022 at different dimensions and maximum fitness evaluation quantity. Statistical results indicate that GKSO has a strong advantage in the competition between two different types of algorithms. Furthermore, five different kinds of real-world constrained optimization problems (OPs) in CEC2020 benchmark constrained optimization functions, including 50 engineering case suites, are selected to evaluate GKSO's performance and the other seven optimizers, further validating GKSO's extensive usefulness and validity in solving practical complex problems.

1. Introduction

Optimization target is to sniff out the optimal solution to a problem under given constraints and objectives. OPs have characteristics of discrete or continuous search space, non-derivative objective function, high dimensionality, non-convexity, etc. However, traditional methods have cumbersome processes and high computational complexity, which cannot be solved in a short time [1]. Therefore, better optimization techniques are demanded to address increasingly complex OPs. Optimization techniques can be roughly divided into two types: deterministic and non-deterministic methods.

Deterministic methods are relative to randomized methods, which

can be classified into linear and nonlinear methods. Generally speaking, deterministic methods are gradient based algorithms that help tackle linear, nonlinear, and differentiable OPs, but they often fall into local optima during the optimization process [2,3]. Heuristic algorithms are also a type of deterministic method, which usually relies on the specificity of the current problem and does not require any random factors for implementation. As long as an input is given, a fixed output can be obtained, and the execution steps of the algorithm are fixed. Therefore, heuristic algorithms, like other deterministic methods, often fall into local optima due to their overly mechanical implementation, and therefore cannot obtain the global optimum [4]. As the heuristic algorithm's amelioration, MA is a non-deterministic method due to its

* Corresponding author.

E-mail address: hugang@xaut.edu.cn (G. Hu).

Table 1

A brief review of MAs in the last three years.

Algorithms	Inspiration	Year
Mayfly Algorithm (MA) [16]	Flight behavior and mating process of mayflies	2020
Equilibrium Optimizer (EO) [17]	Physics-based source and sink models	2020
Sparrow Search Algorithm (SSA) [18]	Foraging behavior of sparrow	2020
Marine Predators Algorithm (MPA) [19]	Foraging behavior of Marine Predator	2020
Gradient-Based Optimizer (GBO) [20]	Gradient-based Newton's method	2020
Capuchin Search Algorithm (CapSA) [21]	Hunting behavior of capuchin	2020
Rat Swarm Optimizer (RSO) [22]	Chasing and attacking behaviors of rats	2020
Generalized Normal Distribution Optimization (GNDO) [23]	Generalized normal distribution model	2020
Adolescent Identity Search Algorithm (AISA) [24]	Adolescent identity development/search process	2020
Human Urbanization Algorithm (HUS) [25]	Human behavior for urbanization and improving life situations	2020
Aquila Optimizer (AO) [26]	Foraging behavior of Aquila	2021
Chameleon Swarm Algorithm (CSA) [27]	Hunting behavior of chameleon	2021
Hunger Games Search (HGS) [28]	Animal hunger concept	2021
Elephant Clan Optimization (ECO) [29]	Individual and collective behaviors of elephants	2021
Smell Agent Optimization (SAO) [30]	Odor source	2021
Wild Geese Algorithm (WGA) [31]	Behavior of wild geese	2021
Flow Direction Algorithm (FDA) [32]	Water flow direction	2021
Arithmetic Optimization Algorithm (AOA) [33]	Distribution behavior of arithmetic operators in mathematics	2021
Honey Badger Algorithm (HBA) [34]	Foraging behavior of honey badger	2021
Firebug Swarm Optimization (FSO) [35]	Reproductive swarming behaviors of Firebugs	2021
Poplar Optimization Algorithm (POA) [36]	Sexual and asexual propagation mechanism of poplar	2022
Aphid-Ant Mutualism (AAM) [37]	A unique relationship between aphids and ants' species	2022
Orca Predation Algorithm (OPA) [38]	Hunting behavior of orca	2022
Fire Hawk Optimizer (PHO) [39]	Foraging behavior of fire hawk	2022
Trees Social Relations (TSR) Optimization Algorithm [40]	Hierarchical and collective life of trees in the jungle	2022
Single Candidate Optimizer (SCO) [41]	A unique set of equations	2022
Snake Optimizer (SO) [42]	Foraging and reproduction behaviors of snakes	2022
weighted meaN oF vectOrs (INFO) [43]	Weighted mean idea	2022
Termite Life Cycle Optimizer (TLCO) [44]	Behavior of termite colonies	2022
Homonuclear Molecules Optimization (HMO) [4]	Bohr atomic model and homonuclear molecules structure	2022
Nutcracker Optimization Algorithm (NOA) [45]	Clark's nutcrackers	2023
Young's Double-Slit Experiment (YDSE) Optimizer [46]	Young's double-slit experiment	2023
Special Relativity Search (SRS) [13]	Special Theory of Relativity	2023
Osprey Optimization Algorithm (OOA) [47]	Behavior of osprey	2023
Exponential Distribution Optimizer (EDO) [48]	Exponential probability distribution model	2023
Spider Wasp Optimizer (SWO) [49]	Behavior of female spider wasp	2023
Chernobyl Disaster Optimizer (CDO) [50]	Nuclear reactor core explosion of Chernobyl	2023
Energy Valley Optimizer (EVO) [51]	Physics principles on stability and different particle decay modes	2023

inclusion of random factors. MA is a problem independent technique that does not exploit any specificity of the problem. It is the product of a combination of the randomized algorithm and the local search algorithm. This non-deterministic method utilizes stochastically created variables to ferret about the approximate optimal solution in the problem space [5]. MA relies on its independence and stochasticity, which adds an extra logic layer to heuristic algorithm that jumps out of local optima. Therefore, compared to traditional and deterministic methods, MAs are simple, efficient, gradient free, and non-derivation [6]. Driven by these advantages, MAs are increasingly favored by researchers.

In the decades long history of modern optimization, hundreds or thousands of MAs have emerged for application in various fields, and most of the inspiration for MAs comes from nature. Since the birth of genetic algorithm (GA) [7] in the 1960 s, it has gone through a period from the embryonic stage of proposing concepts, to the growth stage for OPs, and then to the mature stage of developing towards depth. GA, as the groundbreaking work of MA, is still active in various fields today. In the past twenty years, with the introduction of classic MAs such as particle swarm optimization (PSO) [8] and artificial fish swarm algorithm (AFSA) [9], it has been found that there are countless behaviors among organisms through hunting, migration, reproduction, or growth. Various biomimetic optimizers have emerged day by day, and a metaheuristic storm has immediately swept the entire optimization field. In recent years, with the COVID-19 pandemic and the continuous development of globalization, the progress of MAs is also inseparable from the guidance of various social and natural elements. The changes in paradigms have inspired people to create various magical optimizers. For some researchers attempting to develop new MAs have gradually shifted their focus to some medical methods and their implementation processes. For example, inspired by human self-protection behavior, coronavirus mask protection algorithm (CMPA) [10] was proposed. Inspired by concepts of the herd immunity and the social distance, coronavirus herd immunity optimizer (CHIO) [11] was born. Inspired by a treatment method called immune plasma, researchers have proposed the immune plasma algorithm (IPA) [12].

As one of the most powerful tools for dealing with real-world OPs, MAs have attracted a high degree of interest. Some people may ask why fresh MAs are still being developed, albeit so many existing algorithms. Firstly, just as there is no better analytic method to address all OPs, MAs vary in their ability to solve problems [13]. According to the NFL theorem [14], no optimizer is universal across all problem domains. In addition, although existing MAs have shown a remarkable ability to address many complicated OPs, they still have many shortcomings such as premature convergence, inefficient exploration of the search space, use of too many parameters, and complicated fitness functions, which postpone obtaining the desired solution [15]. Finally, as the constantly emerging challenging OPs get trickier, and novel algorithms continue to achieve better results, they collectively drive the amelioration of existing algorithms and the proposal of fresh algorithms. Table 1 shows some emerging MAs that have received much attention in recent years, providing more options for researchers and experts from a wide range of fields. Therefore, in the field of intelligent optimization, there is a need for more fresh MAs that can tackle various types of OPs, which energizes us to study this work.

Several studies have shown that most of the proposed MAs are inspired by the behavior of animals searching and capturing prey in nature [52]. However, so far, there is no research that can imitate the behavior of GKS in searching for prey. This inspires us to study the unique aspects of GKS behavior and develop mathematical models to mimic it. Therefore, in order to develop a more adaptable and efficient cluster optimizer, we draw ideas from the hunting behavior of GKS, and design a new optimization model called GKS optimizer.

GKS is a fierce marine predator with a keen sense of smell, unique hunting style and strategy, and cooperates with a high level of intelligence. The inspiration for designing this model comes from the hunting and survival behavior of GKS, which mainly includes hunting,

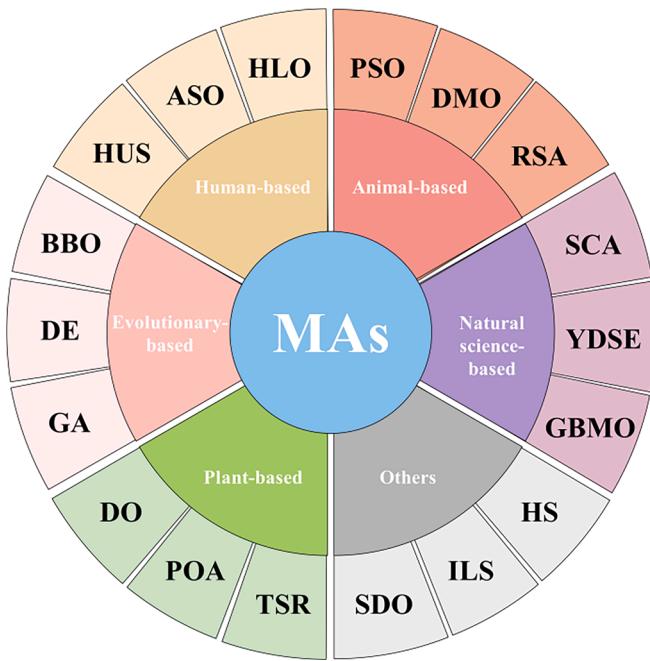


Fig. 1. A brief classification of MAs.

movement, foraging, and color changing escape. Both for the hunting and survival behaviors of GKS can be expanded into a rich theme, leading to the derivation of a powerful optimization technique that can be used to solve a variety of tricky OPs.

In this study, in order to verify the effectiveness and competitiveness of GKS, we compare GKS with eight existing fish optimization algorithms and the other nine MAs, and use CEC2019 and CEC2022 test suite to evaluate them respectively. In addition, to further validate the potential of GKS in engineering optimization problems, GKS and the other seven optimizers are evaluated on 50 engineering cases in the CEC2020 benchmark constrained optimization function. The results indicate that GKS can efficiently solve real-world problems. This study's main contributions are as below:

1. A new nature-inspired MA termed GKS is proposed, and mathematical modeling is conducted on the different behaviors of GKS in the four stages mentioned above.
2. Qualitative analysis is conducted on the proposed GKS through CEC2019 and CEC2022, verifying its excellent ENE capabilities.
3. By introducing statistical metrics to quantitatively analyze the performance of GKS with eight existing fish optimization algorithms on CEC2019, it is verified that GKS has certain competitiveness and advantages in similar comparison algorithms. The theme of “same species’ comparison between algorithms” has not yet been studied by scholars. Therefore, this is also the most prominent highlight in this study.
4. A set of experimental schemes is designed to explore the differences in performance between GKS and the other nine MAs on CEC2022 for different dimensions and maximum fitness evaluation quantities. Meanwhile, statistical metrics are introduced to quantitatively analyze each algorithm’s performance, verifying the superiority of GKS in different optimization algorithms.
5. GKS and the other seven optimizers are tested on 50 engineering case suites in the CEC2020 benchmark constrained optimization function, further verifying the effectiveness of GKS in addressing practical complex problems.

The rest of this article is structured as below: Section 2 describes the related works on MAs. Section 3 discusses the current research gap in

MAs. Section 4 describes the proposed GKS’s inspiration and mathematical model. Section 5 conducts numerical experiments and discussion, including an introduction to two sets of algorithms, sensitivity analysis of parameters, and qualitative and quantitative analysis on CEC2019 and CEC2022. In Section 6, 50 engineering case suites are selected to verify GKS’s validity. Finally, Section 7 presents conclusions and prospects for future research.

2. Related works

State of the art (SOTA) [53] is a technical evaluation indicator used to measure the impact of current technologies in a field, and a good SOTA classification method can help people quickly grasp the cutting-edge dynamics of the field. Currently, with the development of computational methods and the demand in related technical fields, various new optimizers have been proposed and used to address complex OPs that have emerged in various applied disciplines. Most MAs come from simple real-world phenomena and natural inspirations, which can be roughly classified into six categories: animal-based, plant-based, human-based, natural science-based (mathematics, physics and chemistry), evolutionary-based and others algorithms. Fig. 1 is a brief classification of MAs. The following will be explained around Fig. 1.

Firstly, animal-based algorithms’ inspirations come from group behaviors in nature. The most common and well-known algorithm is particle swarm optimization (PSO) [8], which designs a kind of massless particle to simulate birds’ behaviors. The other two examples are dwarf mongoose optimization (DMO) [54] and reptile search algorithm (RSA) [55], which simulate the grouping mechanism of dwarf mongoose family and the hunting behavior of crocodiles, respectively.

Since ancient times, animals and plants have been interdependent and share weal and woe. In plant-based algorithms, dandelion optimizer (DO) [56] simulates the long-distance flight process of dandelion seeds relying on wind. Poplar optimization algorithm (POA) [36] simulates the sexual reproduction and asexual reproduction mechanism of poplar. The inspiration for the tree social relations optimization algorithm (TSR) [40] comes from the hierarchical system and trees’ collective life in the jungle. Trees prevent forest destruction by cooperating collectively for their survival and development needs.

Considering that human groups have a highly conscious tendency to cooperate, many researchers have focused on studying human groups to explore new approaches. Therefore, human-based algorithms are developed based on various behavioral characteristics related to humans, such as learning, society, and emotions [57]. The most popular among them is human learning optimization (HLO) [58], which is inspired by human learning mechanisms and imitates human learning processes. Human urbanization algorithm (HUS) [25] simulates the improvement of human urbanization and living conditions. Alpine skiing optimization (ASO) [59] mainly simulates the behavior of skiers competing for championships.

It has long been known that mathematics, physics, and chemistry are indivisible because these three disciplines have strong logic and interconnected ways of thinking. Among them, mathematics-based algorithms’ representative is sine cosine algorithm (SCA) [60], which utilizes mathematical models of sine and cosine functions to channel generated candidate solutions towards the optimal solution direction or in the opposite direction. A typical physics-based algorithm is Young’s double slit experiment (YDSE) optimizer [46], inspired by the famous Young’s double slit experiment, which reveals the light’s fluctuating nature. Chemistry-based algorithms’ main representative is gases Brownian motion optimization (GBMO) [61], which is enlightened by gases Brownian motion and turbulent rotational motion.

As the earliest type of MAs, evolution-based algorithms primarily mimic the evolution procedure in nature. Among them, the most well-known algorithms are genetic algorithm (GA) [7] and differential evolution algorithm (DE) [62], which are also exemplary among numerous MAs. GA mimics a Darwinian natural selection. DE is a

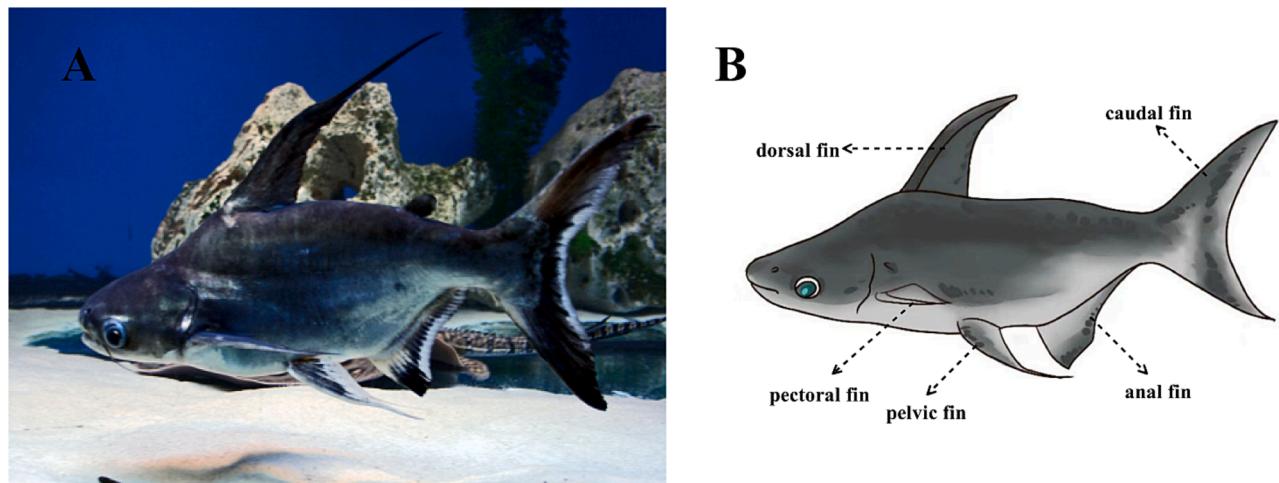


Fig. 2. (A) Realistic appearance of GKS. (B) External organ structure of GKS.

stochastic technique that mimics biological evolution, and its approach is similar to GA, with only differences in specific definitions and operations. In addition, inspired by the principles of biogeography, biogeography-based optimization (BBO) [63] was proposed later. It is based on GA and PSO's evolution, mainly simulating the movement and adaptation process of species in different environments.

The inspiration for the vast majority of MAs comes from the above five categories, but there are also some distinctive MAs. In other categories, it mainly includes iterative local search (ILS) [64], harmony search (HS) algorithm [65], and supply-demand-based optimization (SDO) [66]. Their inspiration comes from special search methods, economics, and music, among others. There are many unique algorithms such as fireworks algorithm (FWA) enlightened by fireworks explosion [67], and transit search (TS) method for detecting exoplanets in astronomy [68].

The ultimate goal of MAs is to optimize practical problems. For the different types of MAs mentioned above, it is necessary to test their respective abilities and demonstrate their own strength in practical applications. Meanwhile, in order to achieve higher solution accuracy and rate, some MAs still need to be appropriately improved to further improve their performance in the field of engineering optimization. For example, Yuan et al. [69] proposed a learning-simulation strategy to assist alpine skiing optimization (LISASO) to find the optimal solution of semi-submersible platform arm. In addition, they also applied the original ASO algorithm to solve multidisciplinary design OPs [70]. Aslan and Demirci [71] introduced a new variant called regional immune plasma algorithm (rIPA) to minimize the measurement noise of electroencephalography signals. Amin et al. [72] proposed a nonlinear chaotic Harris Hawk optimizer (NCHHO) to solve an Internet of Vehicles (IoV) optimization problem. Ahmed et al. [73] introduced the binary simulated normal distribution optimizer (BSNDO) to detect COVID-19 cases. Yuan et al. [74] proposed a dragonfly algorithm based on adaptive resistance and endurance strategies to solve engineering OPs. Yuan et al. [75] also proposed an ameliorated grey wolf optimizer (EOCSGWO) based on elite opposition-based learning and chaotic k-best gravitational search strategy for optimizing the design of automatic drum brakes. Arslan and Aslan [76] proposed a lattice based artificial bee colony (LBABC) algorithm for minimizing electroencephalography signal noise.

3. Research gap

Above MAs play a vital role in the intelligent optimization realm, but they also have certain limitations. In this regard, we make a summary and provide corresponding ideas:

1. The optimization process of MAs revolves around ENE. During the exploration phase, search agents should ferret about unknown areas throughout the entire problem space as much as possible. The exploitation phase involves conducting in-depth searches of potential areas [77]. Any imbalance may result in poor convergence stability or premature convergence of the algorithm. Therefore, how MAs remain a more appropriate balance between ENE is a topic worth discussing. While improvements to the design of specific algorithms may be a shortcut to balancing ENE.
2. An algorithm's performance often depends on parameters, but actually, it is hard to identify a precise parameter to reconcile all OPs. For instance, fine-tuning PSO's parameters c_1 and c_2 will affect its optimization ability. Therefore, how to adjust the tuning parameters of each algorithm is also a challenge in expanding the use of these methods [13]. The design of flexible and variable adaptive parameters is a hot research orientation in this domain.
3. Some algorithms have a zero deviation trend, which signifies that efficiency is high when the theoretical optimum (TO) is zero. When the TO is not zero, these algorithms' efficiency is not ideal [36]. However, in complex OPs in reality, the TO is usually not zero. Therefore, designing simple and efficient algorithms for non-zero optimal solution problems is another noteworthy research direction in this field.
4. The success level of MAs is directly proportional to the number of fitness evaluations most of the time. However, not all fitness evaluations can lead to successful fitness updates. In addition, the more fitness evaluations are conducted, the more time it takes, and the running time of each fitness evaluation may vary depending on the characteristics of OPs [78]. Therefore, how to obtain the optimal answer in the shortest time or with the least fitness evaluation in MAs is a challenge faced by researchers. In the future, perhaps we should pay more attention to the model itself, choose better update methods and condition judgments, and improve the quality of candidate solutions while reducing computational costs.

4. Genghis Khan shark optimizer

This section provides a detailed introduction to the proposed GKS. By simulating the biological mechanism and survival strategy of GKS, the mathematical model and the complexity analysis of GKS are given.

4.1. Inspiration

This MA proposed in this article is based on GKS' dynamic behaviors. We discover some intrinsic characteristics in this fish, so we develop the

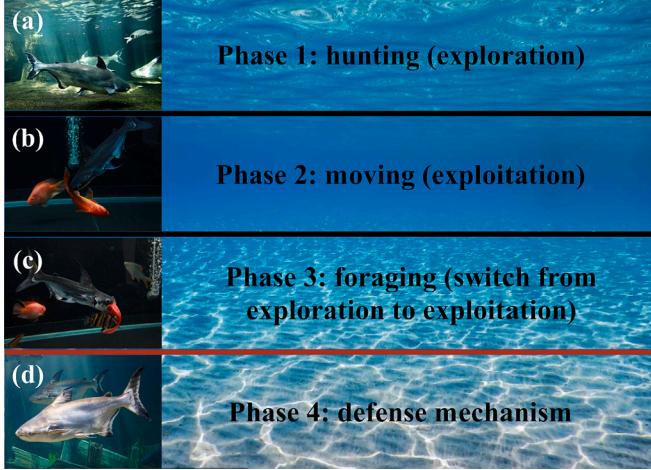


Fig. 3. Four stage behaviors of GKS.

proposed GKS. Genghis Khan shark, scientifically known as *Pangasius sanitwongsei*, is also known as Genghis Khan fish. GKS is a freshwater migratory fish that mainly inhabits larger freshwater rivers such as Chao Phraya River and Mekong River in Asia. GKSs like to cluster and inhabit the middle and lower layers of water bodies. They are lively and often move up the water surface. Fig. 2 (A) and (B) show the realistic appearance of GKS and its external organs' structural diagrams, respectively. The black body, white belly, wide snout, and tall and straight dorsal fin are prominent features shared by GKS. In addition, the fin bone on the dorsal fin of GKS is very thick, and there are obvious silk markings on the dorsal, abdominal, and pectoral fins [79]. The body length of adult GKS generally exceeds 50 cm, and the largest individual can grow to about 3 m, weighing over 300 kg [80]. Under the same body length, the comparison between two GKSs showed that the flatter the head and the stronger the body is the female. On the contrary, the slimmer and more elongated the body is the male. In addition, in most cases, males' dorsal fin wire drawing is significantly longer than females, and many female GKSs' dorsal fin wire drawing will bend backwards abruptly, while males have a relatively flat curve that bends back and up [81]. The sexual maturity of GKS is relatively late. It needs to be kept for 3 to 4 years and can only breed when its weight is more than 3 kg. The breeding season is from April to September. It is a freshwater shark that lays eggs once a year [82].

GKS has a strong tolerance to low oxygen, and can still live normally when cyprinid fish in the same water body have severe floating heads. However, GKS has weak tolerance to low temperatures, and the most suitable water temperature for their survival is 20 to 30 °C. When the water temperature drops to 14 to 18 °C, GKS will experience reduced activity and weak breathing; Shock or even death may occur when the water temperature is between 12 and 13 °C. Although GKS does not have the massive body of a shark, its diet is diverse and large, mainly feeding on fish and animals, as well as various decaying crustaceans and plant debris in the water [83]. Due to GKS's fast swimming speed, its body structure is almost streamlined, and few fish in the Mekong River can match its speed. GKS has a strong physique, and its muscle is very strong with relatively low fat content, making it easy to hunt down other fish. Of course, GKS also has natural enemies. When encountering larger predators, GKS's tail and skin color become lighter, scaring off predators and then fleeing.

Therefore, inspired by the hunting, movement, foraging, and color changing escape behaviors of GKS, we establish a mathematical model for the entire optimization process of GKS. Simulating these natural behaviors is the basic inspiration for modelling the GKS method. Fig. 3 shows the different behaviors of GKS in four stages.

4.2. Mathematical model

This section provides a minute description of GKS's various phases and provides specific mathematical expressions for each stage. Fig. 6 shows the details of the entire optimization process of GKS.

4.2.1. Wandering hunting stage (exploration)

GKSs are powerful predators in the freshwater river region. They often wander around near the bottom of the water, as if patrolling something. In fact, they do this to ensure that there are no more powerful predators than themselves in the entire region. In this way, while ensuring their own safety, they can also expand the search range of the entire space to hunt a location where the best prey is located. As shown in Fig. 6 (a), GKSs patrol in all directions until the population finds the optimal location where the target is located. GKSs do not suddenly attack their prey, but once their prey is targeted, they will wait for the best opportunity to launch a total attack on the target. Fig. 3 (a) reflects a scene of an adult GKS roaming underwater for food.

In GKS, in order to simulate GKS's such natural behavior, a new random position is calculated as the "optimal hunting position" by searching space's upper and lower bounds (ULBs). GKS's position is updated as below:

$$\begin{aligned} X_i^j(t+1) &= X_i^j(t) + \frac{\mathbf{ub}_j + r_1^*(\mathbf{ub}_j - \mathbf{lb}_j)}{it}, i = 1, 2, \dots, N, j = 1, 2, \dots, D, it \\ &= 1, 2, \dots, T. \end{aligned} \quad (4.1)$$

Among them, $X_i^j(t+1)$ represents the i -th member's position on the j -th dimension at time $t+1$, \mathbf{ub}_j and \mathbf{lb}_j represent ULBs on the j -th dimension, respectively. r_1 is a stochastic digit on the interval $[0, 1]$, N indicates the population size, D denotes the problem dimension, it stands for the present iteration count, and T notes iterations totality.

4.2.2. Moving towards the best hunting position (exploitation)

In order to capture higher quality prey, as shown in Fig. 3 (b) and Fig. 6 (b), GKSs often rely on their sensitive sense of smell to constantly approach the best prey. The specific mathematical model for this behavior is as below:

$$\hat{X}_i^j(t+1) = s^*(X_{best}^j(t) - X_i^j(t)). \quad (4.2)$$

Among them, $X_{best}^j(t)$ denotes the known optimal hunting position on the j -th dimension at time t , and s represents GKSs' olfactory intensity as they move towards the optimal prey, which depends on the concentration of odors emitted by the prey. This attractive model is established as follows:

$$s = mI^r, \quad (4.3)$$

wherein, r is a stochastic digit on the interval $[0, 1]$, which reflects the absorption of prey odor by the search agent. For two extreme cases:

- When $r = 0$, this means that the scent emitted by the prey is completely imperceptible to GKS, and the search agent will re-enter the exploration phase.
- When $r = 1$, this odor is completely absorbed by GKS, so the algorithm may easily reach an optimal value (which may be local).

Therefore, parameter r controls the behavior of GKS. I is an attribute strength, which hinges on the population individual's capacity, i.e., each agent current fitness value. Another important parameter is m , which is a non-negative constant that needs to be dictated by OP's characteristics. In addition, m is a key parameter that affects GKS's convergence rate. After multiple experiments, we find that the algorithm performs best when m is taken as 1.5. As seen in Eq. (4.1), although s can guide

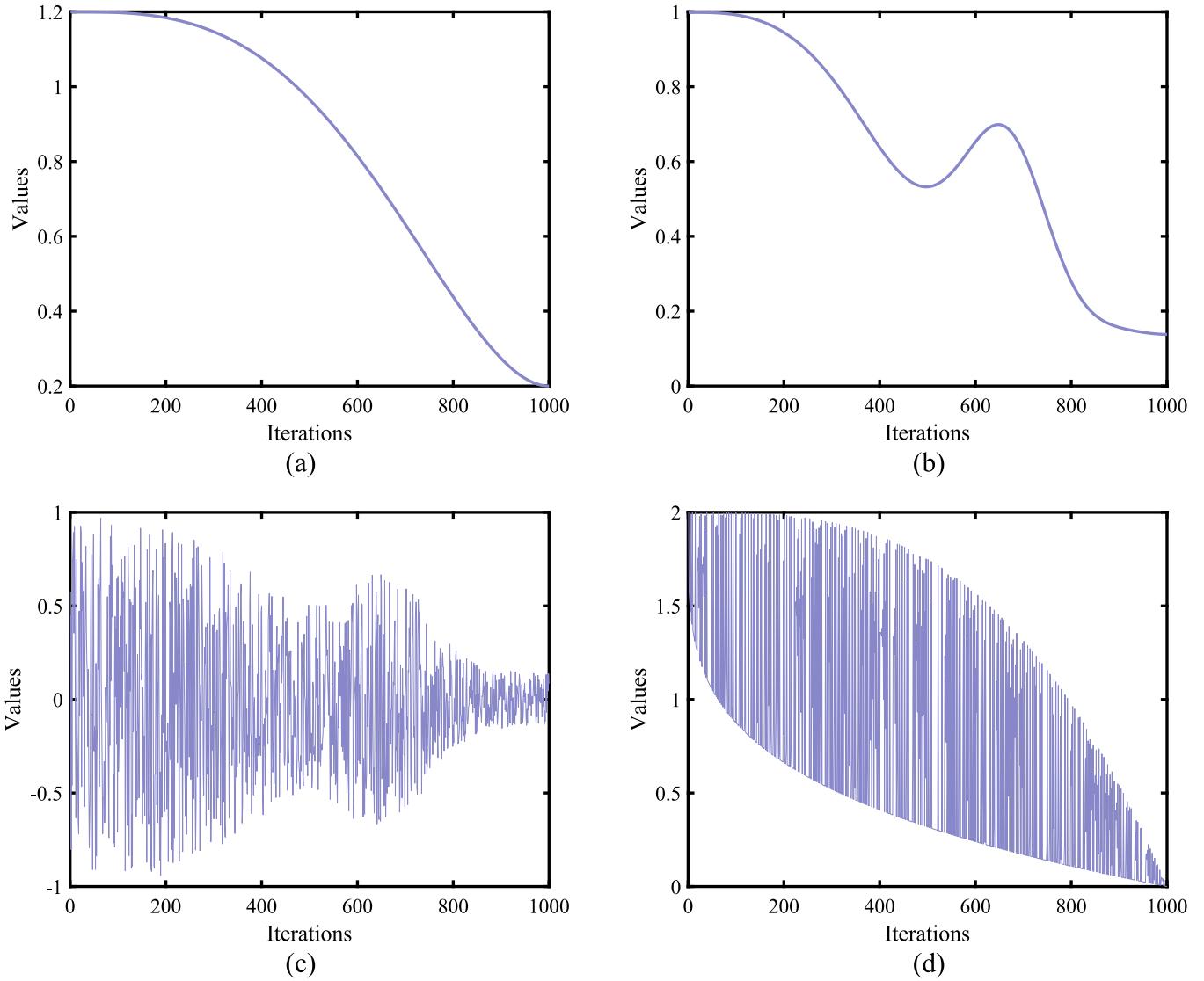


Fig. 4. Convergence curve graphs with increasing number of iterations (a) β value (b) α value (c) ρ value (d) p value.

GKS to move towards the optimal prey position, but bigger s is not better for two main reasons:

- Excessive s values may lead to a relative weakening of the exploration ability of the equation.
- An excessively high s value may drive new candidate solutions beyond the global optimum, which in turn weakens the equation's exploitation ability.

Furthermore, we retain first two optimal agents' positions and renew other agents' positions based on these two optimal positions, making them as close to the optimal position as possible. GKS's position is updated as below:

$$\mathbf{X}_i^j(t+1) = \frac{\hat{\mathbf{X}}_i^j(t+1) + \mathbf{X}_{i-1}^j(t)}{2}. \quad (4.4)$$

4.2.3. Parabolic foraging

Fish use many unexpected ways or strategies to express themselves during their foraging process. For example, in TSO [84], tunas choose to arrange themselves in a parabolic shape for cooperative feeding. GKS is no exception, after approaching the best prey, as shown in Fig. 3 (c) and Fig. 6 (c), GKS uses the prey as a reference point and quickly swims in

front of the prey with its own speed advantage. It uses its wide mouth to launch a fatal blow on the head of the prey, and the entire process is parabolic. Based on the behavior of GKS, we improve the position update formula in TSO and apply it to GKS. The concrete formula is as below:

$$\mathbf{X}_i^j(t+1) = \mathbf{X}_{\text{best}}^j(t) + r_2 * (\mathbf{X}_{\text{best}}^j(t) - \mathbf{X}_i^j(t)) + \lambda * p^2 * (\mathbf{X}_{\text{best}}^j(t) - \mathbf{X}_i^j(t)), \quad (4.5)$$

among them, r_2 is a stochastic digit on $[0, 1]$, λ is a stochastic digit of 1 or -1 , and p is a parameter that controls GKS's movement step size during its activity. This parameter is a nonlinear convergence factor with perturbations. When the p value is large, GKS mainly focuses on global exploration; As the p value decreases, local exploitation will gradually dominate. Fig. 4 (d) shows the curve of p as iteration counts increase. The calculation formula for p is as below:

$$p = 2 * \left\{ 1 - \left(\frac{t}{T} \right)^{\frac{1}{4}} + |\omega(t+1)| * \left[\left(\frac{t}{T} \right)^{\frac{1}{4}} - \left(\frac{t}{T} \right)^3 \right] \right\}. \quad (4.6)$$

In the formula, $|\omega(t+1)|$ is the weight coefficient at time $t+1$, calculated as below:

$$|\omega(t+1)| = 1 - 2\omega^4(t). \quad (4.7)$$

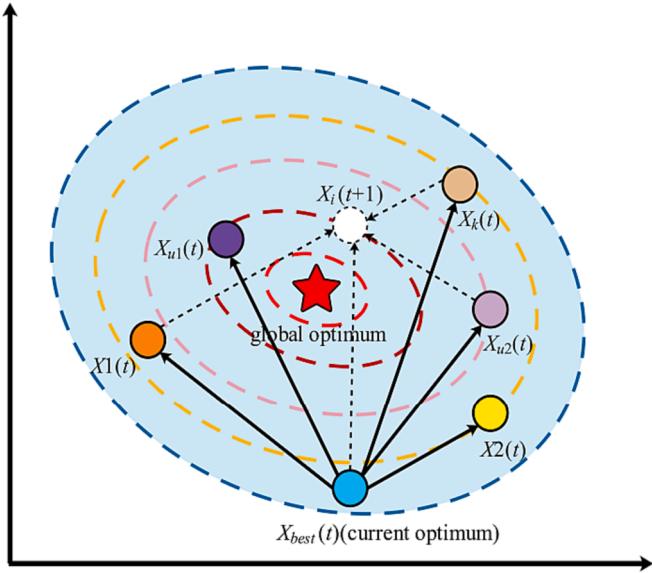


Fig. 5. A local update scheme with neighborhood mutations.

Here is the initial value $\omega(0) = 0.1$.

4.2.4. Self protection mechanism

In the long process of natural environment evolution, freshwater sharks of the same genus have formed some reflection phenomena to adapt to the environment. The discoloration is generally affected by external factors such as being startled or abrupt alterations in water temperature affected by light pollution [85,86]. During the foraging process of GKSs, they often encounter predators that threaten their own safety or compete with their prey. Therefore, in order to get rid of these natural enemies, GKS also has a color changing behavior mechanism similar to that of the cuttlefish, as shown in Fig. 3 (d) and Fig. 6 (d). When GKS is startled, its tail and body color turn light, thereby deterring predators and quickly fleeing. In order to mathematically mimic GKS's behavior, we will perform this process in the fourth stage of GKS, and the specific mathematical model is represented as below:

$$\begin{cases} X_i^j(t+1) = X_i^j(t) + k_1(a_1 X_{best}^j(t) - a_2 X_k^j(t)) + k_2 \rho (a_3 (X2_i^j(t) - X1_i^j(t))) & \text{if } a_1 < 0.5 \\ & + a_2 (X_{u1}^j(t) - X_{u2}^j(t))/2, \\ X_i^j(t+1) = X_{best}^j(t) + k_1(a_1 X_{best}^j(t) - a_2 X_k^j(t)) + k_2 \rho (a_3 (X2_i^j(t) - X1_i^j(t))) & \text{otherwise} \\ & + a_2 (X_{u1}^j(t) - X_{u2}^j(t))/2, \end{cases} \quad (4.8)$$

where k_1 is a uniform distribution stochastic digit between [-1, 1], k_2 is a normal distribution stochastic number with mean value of 0 and standard deviation of 1. a_1 , a_2 , and a_3 are three random numbers, which are calculated by the following formula:

$$\begin{cases} a_1 = l_1 * 2 * rand + (1 - l_1) \\ a_2 = l_1 * rand + (1 - l_1) \\ a_3 = l_1 * rand + (1 - l_1) \end{cases}, \quad (4.9)$$

where $rand$ is a stochastic digit on the range [0, 1], and l_1 is a binary parameter with a value of 0 or 1. Additionally, ρ is an adaptive

coefficient. Fig. 4 (a), (b), and (c) show the curves of β , α , and ρ with raising count of iterations, respectively. They are calculated as follows:

$$\rho = \alpha^*(2 * rand - 1), \quad (4.10)$$

$$\alpha = \left| \beta^* \sin\left(\frac{3\pi}{2} + \sin\left(\frac{3\pi}{2}\beta\right)\right) \right|, \quad (4.11)$$

$$\beta = \beta_{min} + (\beta_{max} - \beta_{min}) * (1 - (\frac{it}{T})^3)^2. \quad (4.12)$$

In Eq. (4.12), β_{min} and β_{max} are taken as 0.2 and 1.2, respectively. it and T are interpreted in the identical way as Eq. (4.1). In addition, the randomly generated solutions $X1_i^j(t)$ and $X2_i^j(t)$ at two t moments can be expressed as:

$$X1_i^j(t) = lb_j + rand * (ub_j - lb_j), \quad (4.13)$$

$$X2_i^j(t) = lb_j + rand * (ub_j - lb_j), \quad (4.14)$$

The calculation solution $X_k^j(t)$ is calculated as below:

$$X_k^j(t) = l_2 * (X_p^j(t) - X_r^j(t)) + X_r^j(t), \quad (4.15)$$

Among them, $X_r^j(t)$ represents a set of solutions generated by random initialization at time t , $X_p^j(t)$ ($p \in \{1, 2, \dots, N\}$) is a stochastically chosen solution at time t . l_2 is taken in the same way as l_1 .

According to Eq. (4.8), the defense mechanism of GKS searches for the space around the current optimal solution by using multiple solutions, such as the current optimum $X_{best}^j(t)$, two stochastically create solutions $X1_i^j(t)$ and $X2_i^j(t)$, two stochastically choose solutions $X_{u1}^j(t)$ and $X_{u2}^j(t)$, a novel stochastically create solution $X_k^j(t)$ and the create current solution $X_i^j(t+1)$. This method is actually a neighbourhood mutation. So as to represent the connection between these variables more simply and intuitively, as shown in Fig. 5, position mutation is performed on the neighborhood of $X_{best}^j(t)$ to generate three mutation vectors $X1_i^j(t)$, $X2_i^j(t)$ and $X_k^j(t)$. Using the minimization problem as an instance of neighborhood updates, if the individual fitness value gained after mutation is less than $X_{best}^j(t)$, then the $X_{best}^j(t)$ will be updated as the

$X_i^j(t+1)$. Otherwise, the individual position with the smallest fitness value in $X1_i^j(t)$, $X_k^j(t)$ and $X_{u2}^j(t)$ is updated to $X_i^j(t+1)$.

The self-protection mechanism of GKS, as a special stage, plays a transitional and transfer role. During the optimization process, this method generates multiple candidate solution positions through neighborhood mutations to detect potential new regions in the problem space, significantly changing the position of the current solution $X_i^j(t+1)$ and improving population diversity. Exploring new and better positions can be said to be a process of reverse exploitation, which helps GKS break away from local optima and avoid search stagnation, achieving good

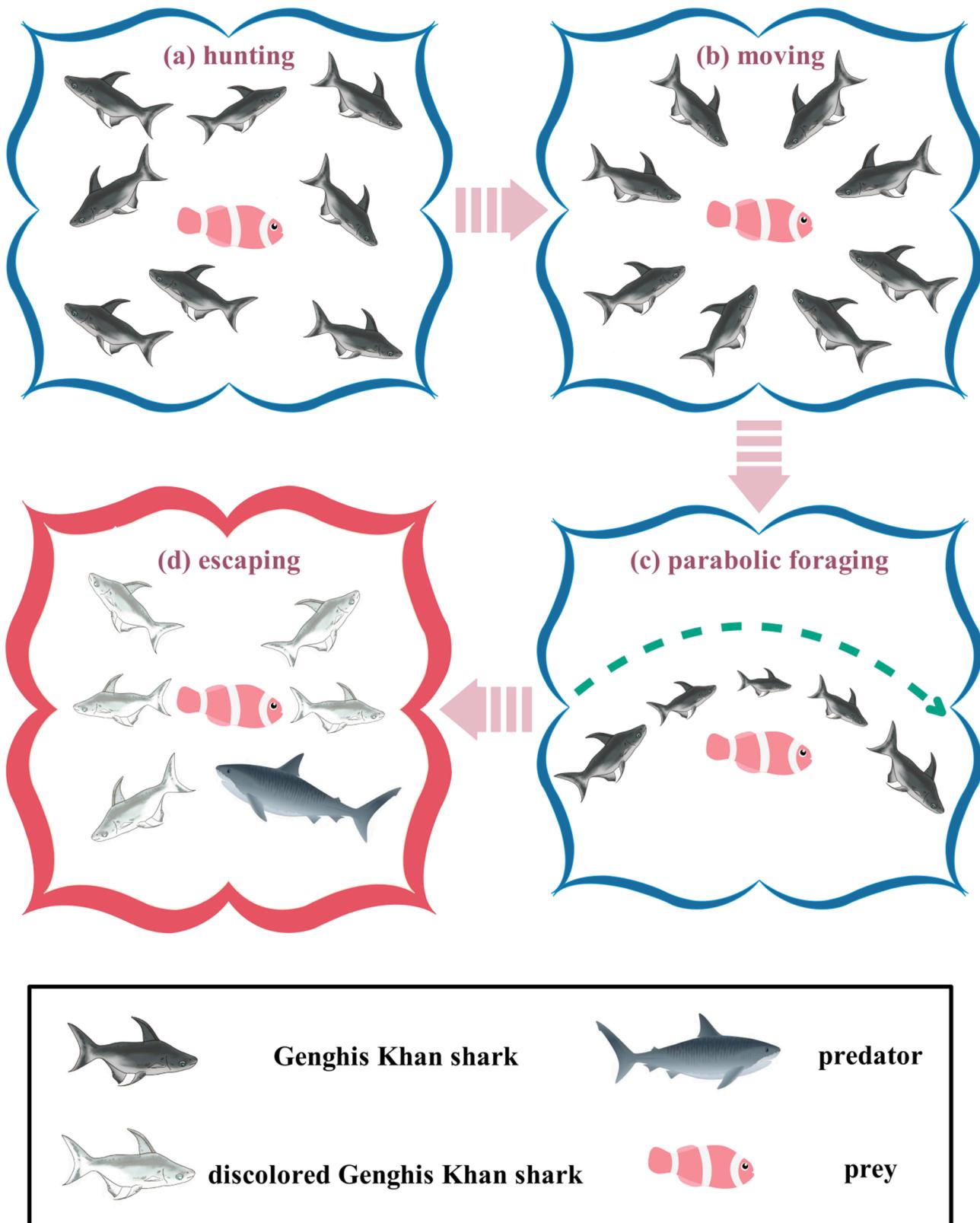


Fig. 6. Proposed GKS optimization process model.

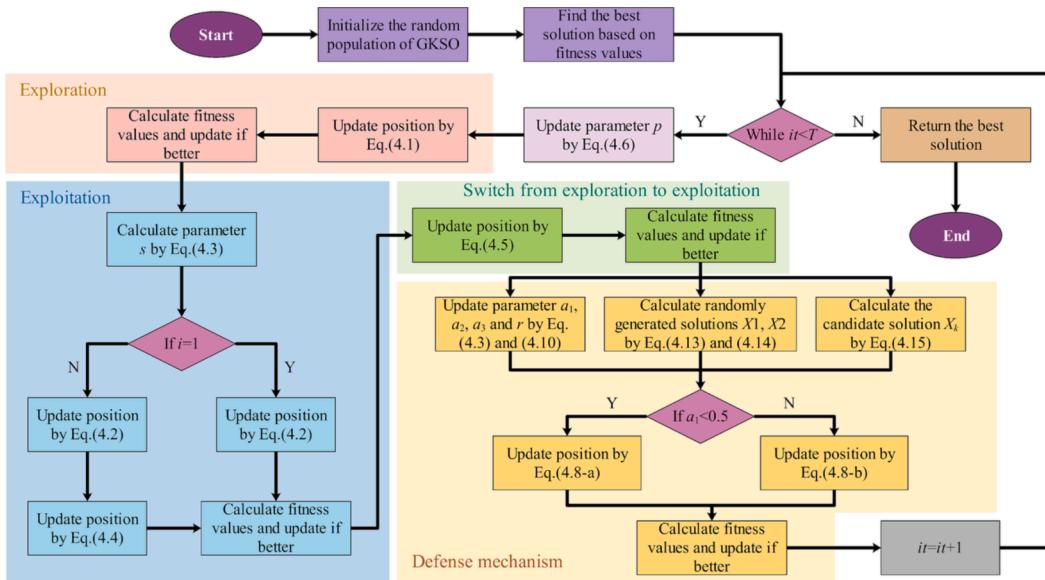


Fig. 7. Flowchart of the proposed GKSO algorithm.

results in the ENE balance of the algorithm.

Algorithm 1: Pseudo-code of GKSO

```

Start GKSO
Input: Algorithmic parameters: agents number(N), maximum iterations (T)
Output: Optimal solution
1: Initialize the population and compute fitness values, store the current optimal
solution
2: For  $it = 1$  to  $T$  Do
3:   Update parameters  $p$  using Eq. (4.6)
4:   For  $i = 1$  to  $N$  Do
5:      $X_i^j(t+1) = X_i^j(t) + (lb_j + r_1 * (ub_j - lb_j)) / it$ 
6:   End For
7:   For  $i = 1$  to  $N$  Do
8:     Compute parameter  $s$  using Eq. (4.3)
9:     If  $i == 1$  Then
10:       $X_i^j(t+1) = s^*(X_{best}^j(t) - X_i^j(t))$ 
11:    Else
12:       $\hat{X}_i^j(t+1) = s^*(X_{best}^j(t) - X_i^j(t))$ 
13:       $X_i^j(t+1) = \frac{1}{2}(\hat{X}_i^j(t+1) + X_{i-1}^j(t))$ 
14:    End If
15:   End For
16:   For  $i = 1$  to  $N$  Do
17:      $X_i^j(t+1) = X_{best}^j(t) + r_2 * (X_{best}^j(t) - X_i^j(t)) + \lambda * p^2 * (X_{best}^j(t) - X_i^j(t))$ 
18:   End For
19:   For  $i = 1$  to  $N$  Do
20:     Update parameters  $a_1, a_2, a_3$  and  $\rho$  using Eq. (4.9) and Eq. (4.10)
21:     Compute randomly generated solutions  $X1, X2$  using Eq. (4.13) and Eq. (4.14)
22:     Compute the candidate solution  $X_k$  using Eq. (4.15)
23:     If  $a_1 < 0.5$  Then
24:        $X_i^j(t+1) = X_i^j(t) + k_1(a_1 X_{best}^j(t) - a_2 X_k^j(t)) + k_2 \rho (a_3 (X2_i^j(t) - X1_i^j(t)))$ 
25:        $+ a_2 (X_{i1}(t) - X_{i2}(t)) / 2$ 
26:     Else
27:        $X_i^j(t+1) = X_{best}^j(t) + k_1(a_1 X_{best}^j(t) - a_2 X_k^j(t)) + k_2 \rho (a_3 (X2_i^j(t) - X1_i^j(t)))$ 
28:        $+ a_2 (X_{i1}(t) - X_{i2}(t)) / 2$ 
29:     End For
30:     Adjust GKSS' positions that move beyond the boundary
31:     Compute fitness values and store the current optimal solution
32:      $T = T + 1$ 
33:   End For
34: Output the optimal solution
End GKSO

```

So as to more apparently emanate GKSO's flow and structure, algorithm 1 gives GKSO's pseudo-code, and plots GKSO's flowsheet as shown in

Fig. 7.

4.3. Time complexity analysis of GKSO algorithm

The time complexity (TC) of GKSO depends on three main factors: initialization, updating solutions, and number of fitness evaluations. They depend on the N, D, T and fit , where fit is the fitness evaluation performed on each agent. However, the complexity of the fitness function is dependent on the problem, so we do not discuss fit here. Firstly, the required TC during the initialization phase is $O(N \times D)$. Subsequently, GKSO enters the iterative search phase of updating solutions. During this procedure, the TC of hunting and moving behavior corresponding to ENE is $O(N \times D) + O(N \times D)$. In addition, the TC of foraging behavior switched from exploration to exploitation is $O(N \times D)$. Finally, the TC of the defense mechanism is $O(N \times D)$. Therefore, The overall TC of GKSO is calculated as follows:

$$\begin{aligned}
O(GKSO) &= O(\text{initialization}) + (O(\text{hunting}) + O(\text{moving}) \\
&\quad + O(\text{foraging}) + O(\text{defense mechanism})) \\
&= O(N \times D) + O(4 \times T \times N \times D) \\
&= O(N \times D \times (1 + 4 \times T))
\end{aligned} \tag{4.16}$$

5. Experimental results and discussion

This section conducts simulation research and evaluation on the optimization performance of GKSO. Firstly, the test suites used in the experiment and the parameters related to the experiment are given.

Table 2
Parameter setting of FCAs.

FCAs	Proposed year	Specifications
MBF	2018	$SP = 0.8, SPdamp = 0.95, Dis = 1.5, Pdis = 0.4$.
ROA	2021	$C = 0.1$.
SFO	2019	$A = 4, \varepsilon = 0.001$.
WSO	2022	Undulating motion frequency $f_{min} = 0.07, f_{max} = 0.75$, acceleration coefficient $\tau = 4.11, a_0 = 6.25, a_1 = 100, a_2 = 0.0005$.
TSO	2021	$z = 0.05, a = 0.7$.
AFSA	2002	Perceptual distance $Visual = 25$, crowding factor $\delta = 27$, Step = 3.
MRFO	2020	Somersault factor $S = 2$.
YSGA	2018	Clusters' number $k = 4$.
GKSO	2023	$m = 1.5$.

Table 3
Parameter settings of OCAs.

OCAs	Proposed year	Specifications
SCA	2016	Constant $a = 2$.
FFA	2018	Parameters $K = 2$, $\alpha = 0.6$, $\beta = 0.4$, $W = 1$, $Q = 0.7$.
GTO	2021	Controlling parameters $p = 0.03$, $\beta = 3$, $w = 0.8$.
AVOA	2021	Probability parameters $L_1 = 0.8$, $L_2 = 0.2$, $w = 2.5$, $P_1 = 0.6$, $P_2 = 0.4$, $P_3 = 0.6$.
MGO	2022	–
ARO	2022	–
BWO	2022	Whale fall probability W_f decreased between from 0.1 to 0.05.
BDO	2022	Acceleration factor $a_f = 4.0$, strategy randomizer factor $S_r = 0.8$, $\theta_{\min} = 0.2$.
COA	2023	–

Meanwhile, different types of algorithms in two sets of experiments are introduced. In addition, the sensitivity analysis of a group of control parameters in GKS0 is carried out. Finally, the performance of GKS0

thoroughly analyzed from both qualitative and quantitative perspectives, and the experimental results are statistically analyzed. The specific experimental content and result analysis are as follows.

5.1. Experimental design

In order to get a better grasp of the proposed GKS0, this section selects CEC2019 [87] and CEC2022 [88] to assess the performance of GKS0. CEC2019 contains 10 functions for single-objective optimization problems. Among them, F1 to F3 have different dimensions and domains, while F4 to F10 are 10-dimensional minimization problems bounded by [-100, 100]. CEC2022 is an effective method for evaluating algorithm performance and verifying its ability to solve complex OPs. It contains 12 test functions, including unimodal function (F1), basic functions (F2-F5), hybrid functions (F6-F8) and composition functions (F9-F12). In order to avoid unexpected factors affecting the experiment, each test function is run 20 times independently for various algorithms. Meanwhile, when performing the test in CEC2019, N and T are set to 50 and 500, respectively. When testing in CEC2022, as the dimensions of

Table 4
Experimental results of different m on CEC2019 ($N = 50$, $T = 500$).

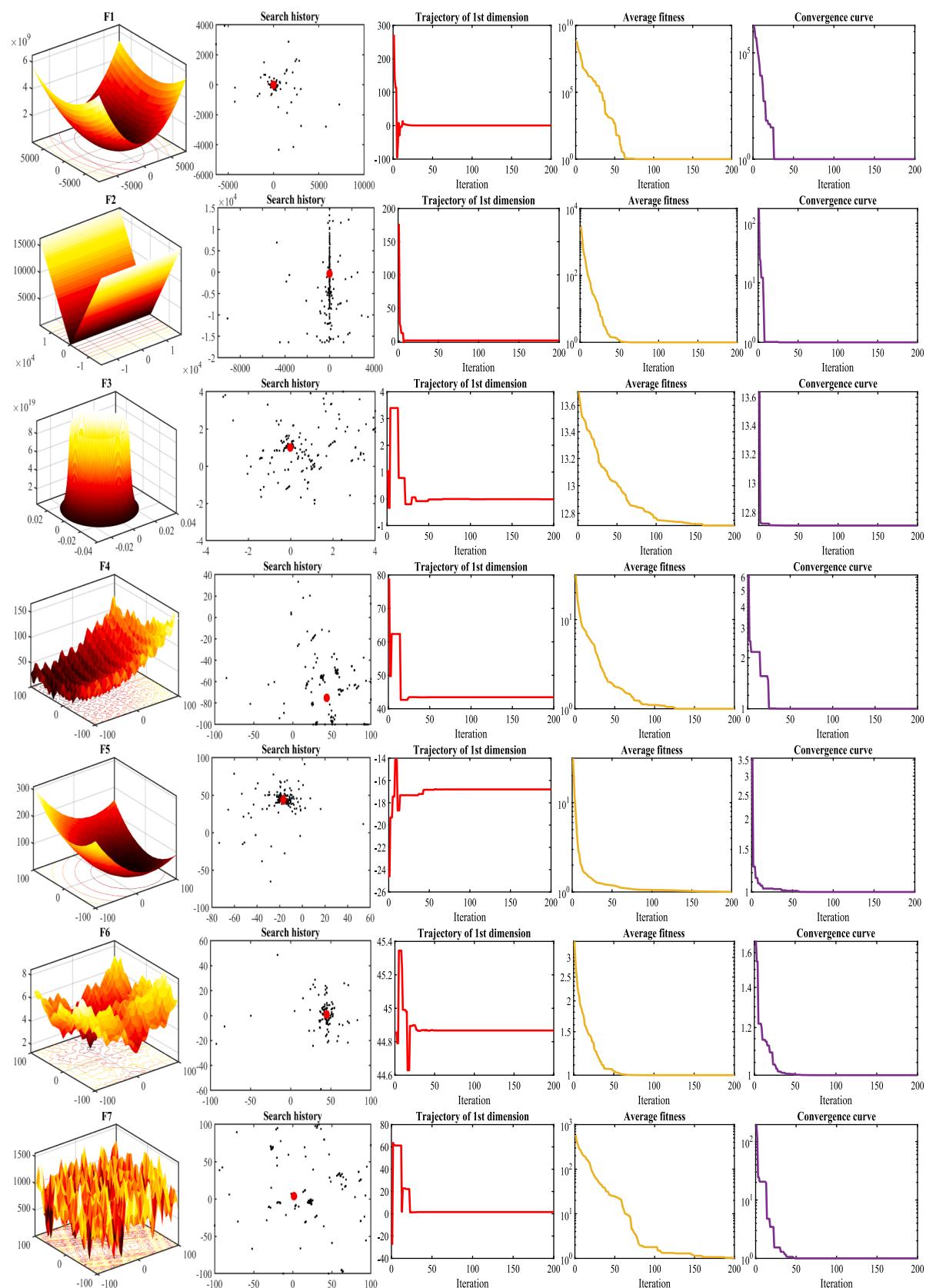
F	$m = 0.5$	$m = 1$	$m = 1.5$	$m = 2$	$m = 2.5$	$m = 3$	$m = 4$	$m = 5$
F1	1.0000E + 00							
F2	4.3998E + 00	4.3536E + 00	4.3142E + 00	4.5134E + 00	4.4728E + 00	4.4112E + 00	4.6302E + 00	4.4265E + 00
F3	1.5699E + 00	1.4092E + 00	1.3887E + 00	1.8826E + 00	1.7085E + 00	1.7725E + 00	1.7133E + 00	1.8824E + 00
F4	2.9325E + 01	2.4780E + 01	2.0850E + 01	2.4431E + 01	2.1098E + 01	2.5277E + 01	2.6969E + 01	2.2989E + 01
F5	1.2059E + 00	1.1918E + 00	1.1678E + 00	1.1860E + 00	1.1799E + 00	1.1744E + 00	1.1962E + 00	1.2200E + 00
F6	3.4475E + 00	3.6774E + 00	3.1787E + 00	3.5218E + 00	4.1404E + 00	4.4334E + 00	3.8223E + 00	3.5862E + 00
F7	7.9075E + 02	8.5730E + 02	7.4074E + 02	8.1979E + 02	8.1379E + 02	7.8550E + 02	9.1706E + 02	8.9531E + 02
F8	3.6515E + 00	3.7582E + 00	3.5887E + 00	3.7049E + 00	3.7056E + 00	3.7741E + 00	3.7042E + 00	3.6348E + 00
F9	1.1669E + 00	1.1536E + 00	1.1431E + 00	1.1543E + 00	1.1476E + 00	1.1539E + 00	1.1501E + 00	1.1736E + 00
F10	1.7308E + 01	1.8575E + 01	1.7243E + 01	2.0173E + 01	1.9355E + 01	2.0250E + 01	2.0096E + 01	1.9408E + 01
Rank	2	4	1	5	2	5	8	5

Table 5
Experimental results of different m on 10-dimensional CEC2022 ($N = 50$, $T = 500$).

F	$m = 0.5$	$m = 1$	$m = 1.5$	$m = 2$	$m = 2.5$	$m = 3$	$m = 4$	$m = 5$
F1	3.0000E + 02							
F2	4.0788E + 02	4.0573E + 02	4.0370E + 02	4.0589E + 02	4.0500E + 02	4.1178E + 02	4.0720E + 02	4.0804E + 02
F3	6.0297E + 02	6.0225E + 02	6.0158E + 02	6.0335E + 02	6.0172E + 02	6.0085E + 02	6.0178E + 02	6.0184E + 02
F4	8.1960E + 02	8.1811E + 02	8.1617E + 02	8.2074E + 02	8.1701E + 02	8.1816E + 02	8.2134E + 02	8.1856E + 02
F5	9.0326E + 02	9.0311E + 02	9.0228E + 02	9.0404E + 02	9.0252E + 02	9.0221E + 02	9.0178E + 02	9.0189E + 02
F6	2.8500E + 03	2.5952E + 03	2.5539E + 03	3.7115E + 03	3.0798E + 03	2.7138E + 03	3.1400E + 03	2.7317E + 03
F7	2.0195E + 03	2.0171E + 03	2.0201E + 03	2.0213E + 03	2.0200E + 03	2.0186E + 03	2.0201E + 03	2.0198E + 03
F8	2.2189E + 03	2.2190E + 03	2.2151E + 03	2.2154E + 03	2.2185E + 03	2.2173E + 03	2.2183E + 03	2.2161E + 03
F9	2.5293E + 03							
F10	2.5118E + 03	2.5123E + 03	2.5003E + 03	2.5116E + 03	2.5003E + 03	2.5063E + 03	2.5003E + 03	2.5003E + 03
F11	2.6550E + 03	2.6275E + 03	2.6567E + 03	2.6553E + 03	2.6400E + 03	2.7100E + 03	2.6476E + 03	2.6200E + 03
F12	2.8651E + 03	2.8644E + 03	2.8643E + 03	2.8652E + 03	2.8646E + 03	2.8640E + 03	2.8668E + 03	2.8644E + 03
Rank	7	5	1	8	3	4	6	2

Table 6
Experimental results of different m on 20-dimensional CEC2022 ($N = 100$, $T = 1000$).

F	$m = 0.5$	$m = 1$	$m = 1.5$	$m = 2$	$m = 2.5$	$m = 3$	$m = 4$	$m = 5$
F1	3.0000E + 02							
F2	4.4374E + 02	4.3514E + 02	4.3233E + 02	4.4542E + 02	4.3457E + 02	4.4682E + 02	4.5006E + 02	4.3504E + 02
F3	6.1674E + 02	6.1676E + 02	6.1596E + 02	6.1753E + 02	6.1738E + 02	6.1977E + 02	6.1958E + 02	6.1829E + 02
F4	8.6452E + 02	8.7094E + 02	8.6527E + 02	8.6726E + 02	8.6308E + 02	8.6567E + 02	8.6660E + 02	8.6646E + 02
F5	1.3536E + 03	1.3483E + 03	1.2391E + 03	1.2051E + 03	1.2917E + 03	1.3485E + 03	1.2499E + 03	1.1871E + 03
F6	6.5326E + 03	6.0466E + 03	5.4416E + 03	8.5392E + 03	7.1639E + 03	7.7116E + 03	6.1374E + 03	7.4394E + 03
F7	2.0666E + 03	2.0587E + 03	2.0559E + 03	2.0645E + 03	2.0644E + 03	2.0625E + 03	2.0653E + 03	2.0740E + 03
F8	2.2240E + 03	2.2305E + 03	2.2240E + 03	2.2227E + 03	2.2255E + 03	2.2361E + 03	2.2301E + 03	2.2301E + 03
F9	2.4808E + 03							
F10	2.5159E + 03	2.5373E + 03	2.5090E + 03	2.5270E + 03	2.5273E + 03	2.5275E + 03	2.5004E + 03	2.5147E + 03
F11	2.8700E + 03	2.9327E + 03	2.9280E + 03	2.9100E + 03	2.9657E + 03	2.9050E + 03	2.9100E + 03	2.9350E + 03
F12	2.9583E + 03	2.9645E + 03	2.9591E + 03	2.9789E + 03	2.9636E + 03	2.9702E + 03	2.9613E + 03	2.9678E + 03
Rank	2	7	1	5	3	8	4	5

**Fig. 8.** Qualitative results of GKS0 on CEC2019.

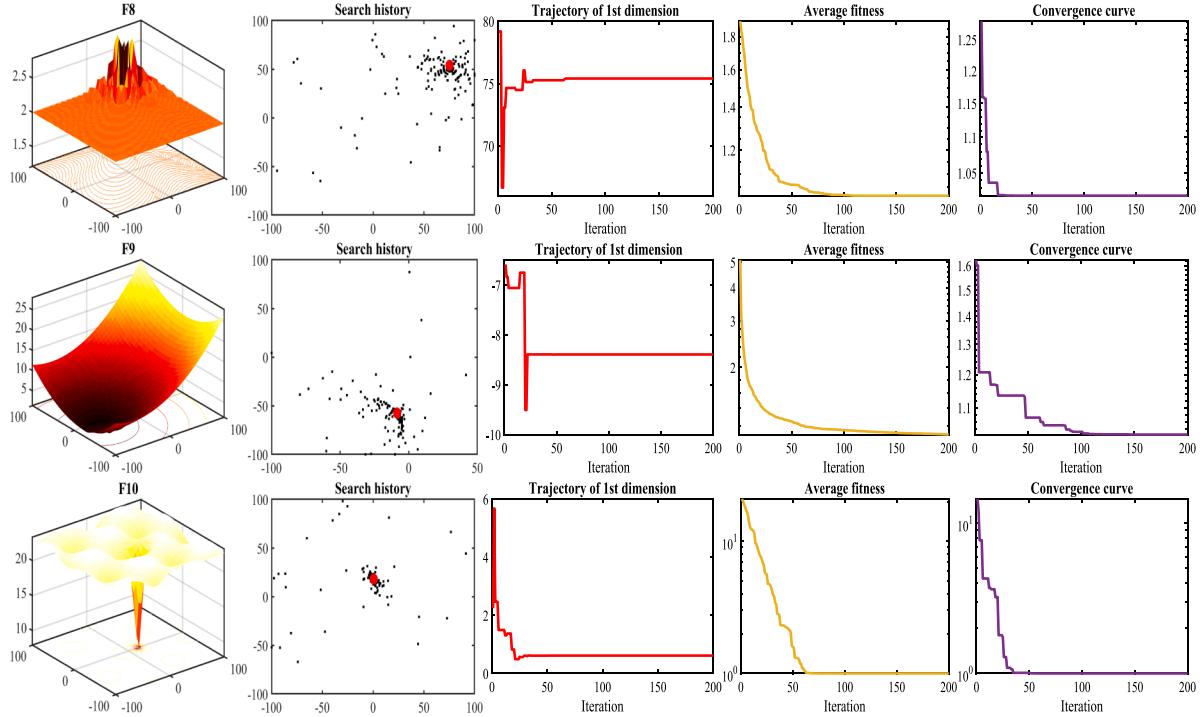


Fig. 8. (continued).

the problem change, we set the sizes of N and T differently to verify the robustness of GKS. The specific parameter settings will be provided in subsequent corresponding experiments. All experiments are carried out in the Matlab R2022b environment and a computer equipped with an Intel (R) Core (TM) i7-1165G7 processor and 16 GB of RAM is used.

5.1.1. Introduction to fish comparison algorithms (FCAs)

In the first set of experiments, GKS is contrasted with eight existing MAs for fish. (Note: Whales and dolphins belong to the whale class and are not fish, while GKS belongs to fish, so we select fish algorithms for comparison.) Various FCAs' parameter settings are detailed in Table 2. FCAs' specific content is as follows:

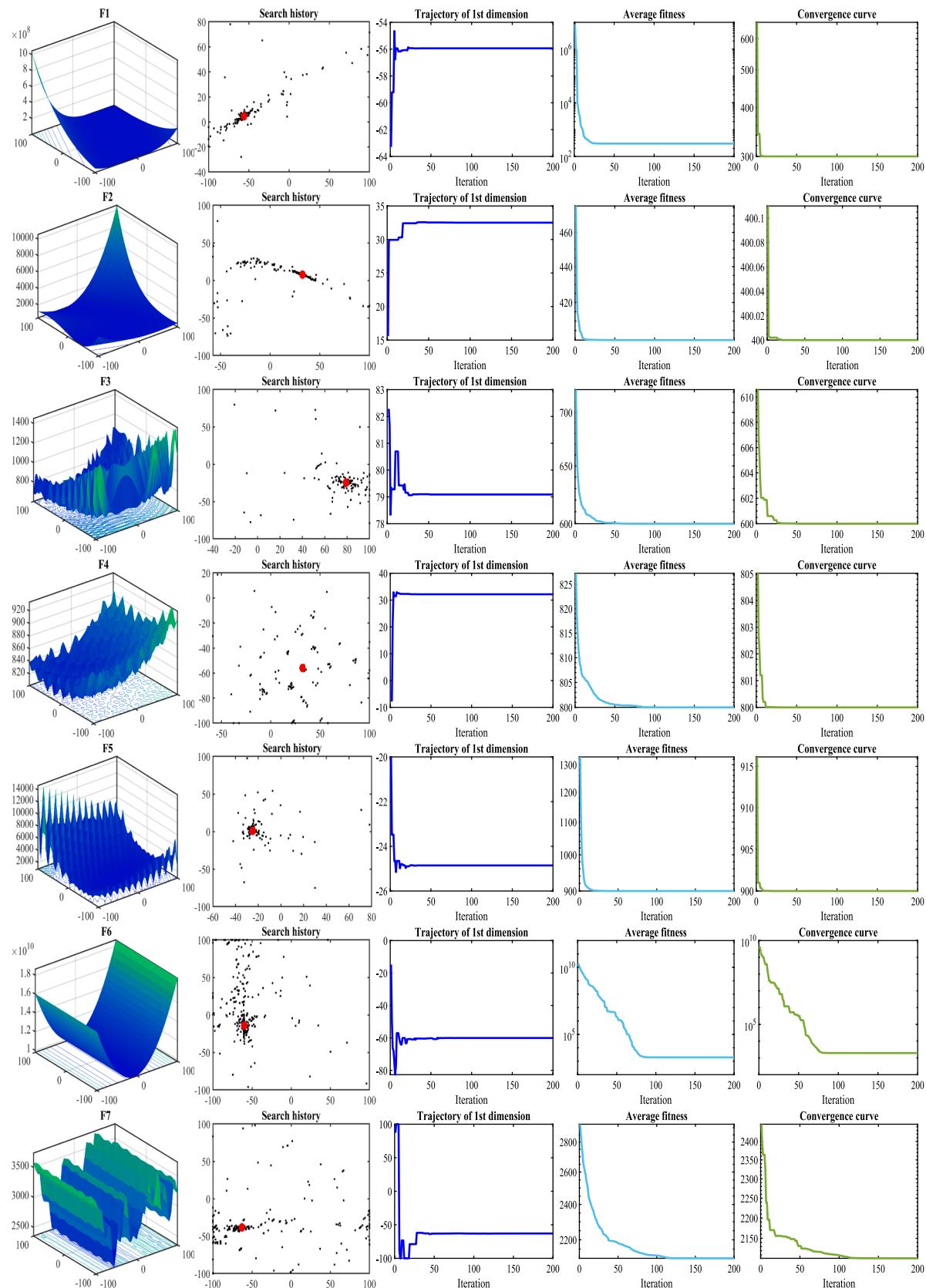
- Mouth Brooding Fish (MBF) algorithm [89]: MBF mimics symbiotic interaction strategies adopted by organisms for survival and reproduction in ecosystems. MBF ferrets about the optimum based on the motion, dispersion, and protective behavior patterns of Mouth Brooding Fish.
- Remora Optimization Algorithm (ROA) [90]: ROA simulates the foraging process of remora fish adsorbed on different body types of hosts in two stages of ENE.
- Sailfish Optimizer (SFO) [77]: SFO includes two populations, sailfish is used to strengthen
- the current optimal search, and sardine is used to disperse the search space.
- White Shark Optimizer (WSO) [91]: WSO simulates the rapid movement of white sharks towards their prey, surrounding the best prey, approaching the best white shark, and the behavior of white shark swarms.
- Tuna Swarm Optimization (TSO) [84]: TSO simulates tuna schools' cooperative foraging behaviors, including two strategies: spiral and parabolic foraging.
- Artificial Fish Swarm Algorithm (AFSA) [9]: AFSA mimics artificial fish swarms' three basic behaviors: foraging, clustering, and tail chasing behavior.

- Manta Ray Foraging Optimization (MRFO) [92]: MRFO simulates manta rays' feeding processes in the ocean, including chain feeding, spiral feeding, and flipping feeding.
- Yellow Saddle Goatfish Algorithm (YSGA) [93]: YSGA performs different search paths by playing two search agent roles: chasing fish and intercepting fish. The algorithm framework is mainly composed of four different behavioral patterns: chasing fish, intercepting fish, character swapping, and changing regions.

5.1.2. Introduction to other comparative algorithms (OCAs)

In the second set of experiments, GKS is contrasted with the other nine MAs. Various OCAs' parameter settings are detailed in Table 3. OCAs' specific content is as follows:

- Sine Cosine Algorithm (SCA) [60]: SCA adopts a mathematical model of sine and cosine functions, and the resulting candidate solutions are guided to fluctuate in the direction of the optimal solution or in the reverse.
- Farmland Fertility Algorithm (FFA) [94]: FFA divides farmland into different areas based on soil quality, and optimizes the soil quality of each area through a specific material to increase yield and produce high-quality products.
- Artificial Gorilla Troops Optimizer (GTO) [95]: GTO mimics the collective life and social intelligence behavior of gorilla troops.
- African Vultures Optimization Algorithm (AVOA) [96]: AVOA mimics the foraging and navigation behavior of African vultures.
- Mountain Gazelle Optimizer (MGO) [97]: MGO mimics the group life and hierarchical structure of wild mountain gazelles.
- Artificial Rabbits Optimization (ARO) [57]: ARO simulates rabbits' survival strategies for foraging and hiding, and switched between the two strategies through energy contraction.
- Beluga Whale Optimization (BWO) [98]: BWO establishes three stages of exploration, exploitation and whale fall, corresponding to paired swimming, predation and whale fall behavior.
- Bottlenose Dolphin Optimizer (BDO) [99]: BDO simulates bottlenose dolphins' mud ring feeding behaviors, with two main stages driving dolphin selection and movement towards prey.

**Fig. 9.** Qualitative results of GKS0 on CEC2022.

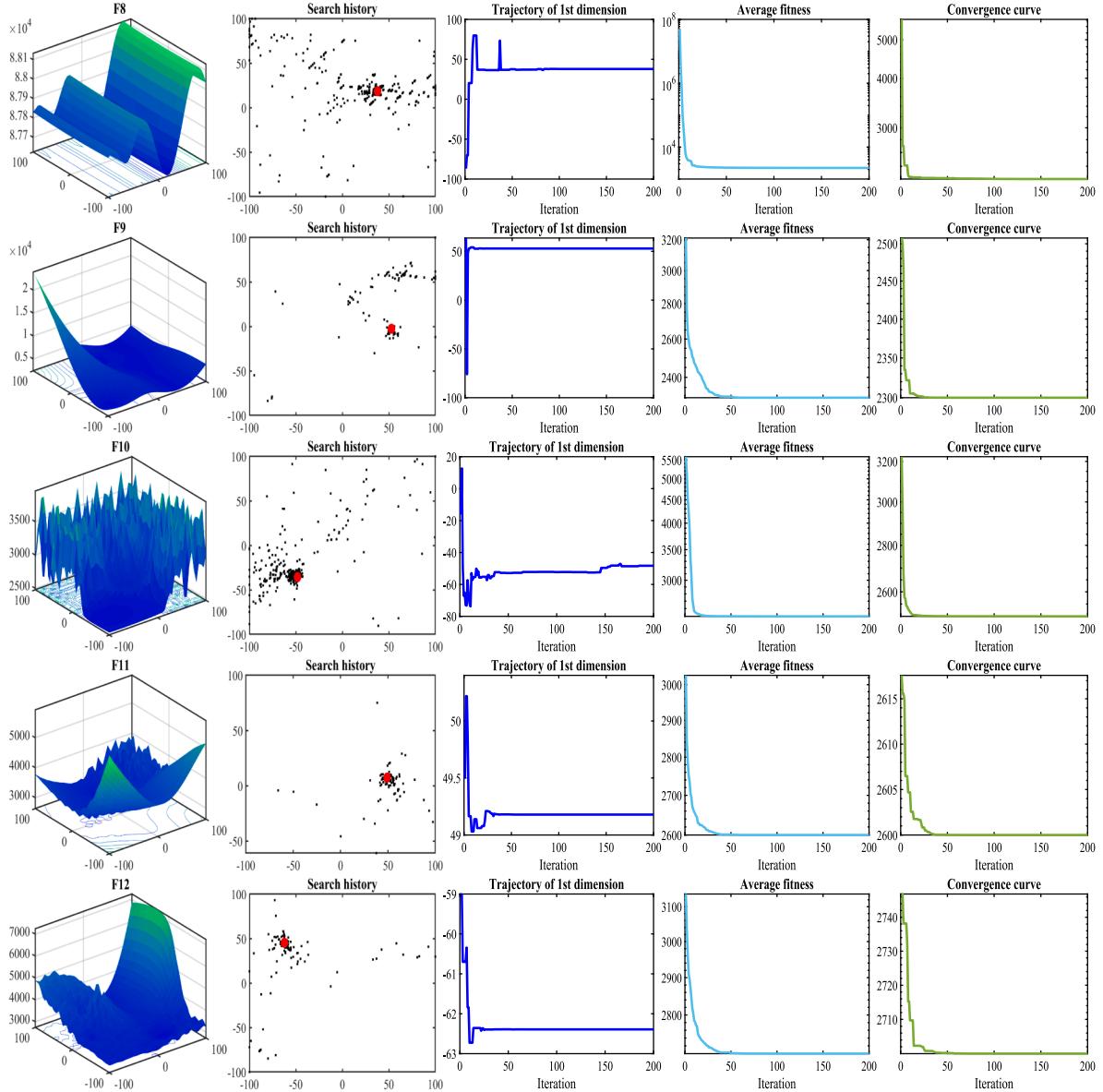


Fig. 9. (continued).

- Coati Optimization Algorithm (COA) [100]: COA simulates two natural behaviors of coatis: hunting iguanas and fleeing predators.

5.2. Sensitivity analysis of parameter (m)

A control parameter that fits the algorithm can affect its optimization performance. In GKSO, there is only one control parameter m that affects the optimal search ability of the algorithm. As mentioned in subsection 4.2.2, the scale parameter m , as a non-negative constant, affects the olfactory intensity s of GKS when moving towards the optimal prey. Considering the setting requirements of the s value, the m value also needs to be set reasonably. This section tests m on CEC2019 and CEC2022, where N and T in CEC2019 are set to 50 and 500, respectively. On the 10-dimensional CEC2022, the sizes of N and T are the same as the settings in CEC2019, and on the 20 dimension, N and T are set to 100 and 1000. Tables 4 to 6 show the mean values and rankings obtained for different m in CEC2019 and CEC2022. All three tables indicate that GKSO achieves the best results on both test sets when $m = 1.5$. At this point, the ENE ability of the olfactory intensity s in Eq. (4.3) achieves the optimal balance, and the algorithm achieves the best performance.

Therefore, the subsequent experiments in this chapter also set $m = 1.5$.

5.3. Qualitative analysis

The so-called qualitative approach refers to exploring the essence of things through non quantitative means. In this section, we use four well-known qualitative indicators: search history, 1st individual's trajectory in the 1st dimension, average fitness and convergence curve. Through these indicators, we can clearly observe the optimization behavior of GKSO during the iteration process on CEC2019 and CEC2022, thereby helping us to intuitively analyze the performance of GKSO.

Fig. 8 and **Fig. 9** reflect GKSO's qualitative outcomes addressing CEC2019 and CEC2022, respectively. The first column draws shapes of test functions in 2D space to concentrate on the domain's topology. The second column depicts historical location search records of the interaction between GKSS and prey in space. We select search agents' first two dimensions and iteratively generate corresponding two-dimensional scatter plots. The third column displays the trajectory curve of 1st agent in the 1st dimension, reflecting the dynamic changes in individual position. The fourth column reflects the search agent's

Table 7

Comparison of results between GKSO and other FCAs on CEC2019.

F	Index	Algorithms								
		MBF	ROA	SFO	WSO	TSO	AFSA	MRFO	YSGA	GKSO
F1	Mean	4.2111E+05	1.0000E+00	1.0000E+00	8.6171E+02	1.0000E+00	1.1195E+09	1.0000E+00	2.8560E+06	1.0000E+00
	Std	5.0159E+05	8.7295E-07	5.3824E-11	2.5646E+03	0.0000E+00	5.9644E+08	0.0000E+00	2.7614E+06	0.0000E+00
	RMSE	6.4524E+05	9.0056E-07	6.3049E-11	2.6437E+03	0.0000E+00	1.2614E+09	0.0000E+00	3.9244E+06	0.0000E+00
	δ	8.1230E+03	2.2204E-16	2.2204E-16	1.6032E-03	0.0000E+00	2.8996E+08	0.0000E+00	2.6852E-10	0.0000E+00
	p-value	8.0065E-09(-)	8.0065E-09	8.0065E-09	8.0065E-09(-)	NaN(=)	8.0065E-09(-)	NaN(=)	8.0065E-09	-
	Rank	7	5	4	6	1	9	1	8	1
	Mean	2.9572E+02	5.0001E+00	5.0000E+00	7.1500E+01	4.6002E+00	2.9460E+04	4.5030E+00	3.0026E+03	4.3112E+00
	Std	1.5293E+02	2.7957E-04	5.8882E-06	5.5075E+01	3.1639E-01	5.1646E+03	3.3683E-01	6.8554E+02	1.6598E-01
	RMSE	3.3027E+02	4.0001E+00	4.0000E+00	8.8610E+01	3.6134E+00	2.9886E+04	3.5183E+00	3.0751E+03	3.3151E+00
F2	δ	8.3109E+01	4.0000E+00	4.0000E+00	1.2085E+01	3.2350E+00	1.9796E+04	3.2419E+00	1.6598E+03	3.2172E+00
	p-value	6.7956E-08(-)	6.7956E-08	6.7956E-08	6.7956E-08(-)	1.0161E-03(-)	6.7956E-08(-)	1.0991E-01	6.7956E-08	-
	Rank	7	5	4	6	3	9	2	(=)	(-)
	Mean	2.0993E+00	6.2593E+00	6.4461E+00	2.3023E+00	1.5991E+00	3.9219E+00	1.4091E+00	2.6009E+00	1.5689E+00
	Std	1.2331E+00	1.5000E+00	9.0346E-01	1.0976E+00	4.5211E-01	4.5163E-01	1.1344E-09	1.4921E+00	7.1405E-01
	RMSE	1.6288E+00	5.4587E+00	5.5169E+00	1.6853E+00	7.4371E-01	2.9548E+00	4.0913E-01	2.1628E+00	8.9887E-01
	δ	4.0913E-01	2.8766E+00	3.5787E+00	9.5762E-02	4.0913E-01	2.3710E+00	4.0913E-01	4.0915E-01	4.0914E-01
	p-value	9.6763E-01	9.1728E-08	7.8980E-08	1.2941E-04(-)	4.5695E-01	9.1266E-07(-)	6.4399E-08(-)	2.4706E-04	-
	Rank	4	8	9	5	3	7	1	6	2
F3	Mean	2.1865E+01	5.8645E+01	6.9023E+01	1.3778E+01	2.2541E+01	2.9862E+01	2.8312E+01	4.5580E+01	2.5426E+01
	Std	7.3213E+00	1.6246E+01	1.1079E+01	7.7293E+00	7.5379E+00	5.3830E+00	1.3390E+01	1.3255E+01	8.8314E+00
	RMSE	2.2052E+01	5.9781E+01	6.8875E+01	1.4833E+01	2.2759E+01	2.9335E+01	3.0269E+01	4.6415E+01	2.5899E+01
	δ	6.9647E+00	3.2208E+01	4.5007E+01	2.9866E+00	9.9496E+00	1.5436E+01	9.9506E+00	1.9899E+01	8.9546E+00
	p-value	2.1841E-01	2.9598E-07	6.7956E-08	1.2941E-04	2.3394E-01	7.2045E-02	9.3532E-01	2.5960E-05	-
	Rank	2	8	9	1	3	6	5	7	4
	Mean	1.1937E+00	5.5479E+01	7.5386E+01	1.1599E+00	1.2633E+00	1.1209E+00	1.0987E+00	2.6207E+00	1.1743E+00
	Std	1.0761E-01	3.8943E+01	3.2776E+01	1.4943E-01	2.1355E-01	3.3451E-02	6.4325E-02	1.1855E+00	8.2381E-02
	RMSE	2.2029E-01	6.6398E+01	8.0956E+01	2.1626E-01	3.3561E-01	1.2522E-01	1.1689E-01	1.9904E+00	1.9193E-01
F4	δ	3.9368E-02	1.3122E+01	2.8480E+01	1.2317E-02	7.1330E-02	7.0049E-02	7.3960E-03	5.9518E-01	4.1876E-02
	p-value	7.1498E-01	6.7956E-08	6.7956E-08	1.3328E-01	9.0907E-02	2.0735E-02	1.7824E-03	6.7956E-08	-
	Rank	5	8	9	3	6	2	1	7	4
	Mean	3.2359E+00	1.0018E+01	1.0736E+01	2.3356E+00	5.3369E+00	4.9233E+00	3.0901E+00	8.4804E+00	4.4037E+00
	Std	1.0268E+00	1.5996E+00	1.0501E+00	8.6188E-01	1.5788E+00	7.1365E-01	1.2125E+00	9.9719E-01	1.3675E+00
	RMSE	2.4497E+00	9.1518E+00	9.7897E+00	1.5778E+00	4.6018E+00	3.9845E+00	2.4011E+00	7.5432E+00	3.6554E+00
	δ	3.9017E-01	5.9899E+00	8.1719E+00	1.1411E-01	2.2573E+00	2.0963E+00	3.7191E-02	6.1535E+00	1.2649E+00
	p-value	1.6669E-02	6.7956E-08	6.7956E-08	9.7480E-06	4.9864E-02(-)	6.7868E-02	6.0403E-03	6.7956E-08	-
	Rank	3	8	9	1	6	5	2	7	4
F5	Mean	7.4541E+02	1.6778E+03	1.6253E+03	8.8374E+02	8.4487E+02	8.3071E+02	6.5632E+02	9.5276E+02	7.0078E+02
	Std	2.2693E+02	2.9164E+02	2.7049E+02	4.2732E+02	3.3356E+02	1.2174E+02	2.4726E+02	2.4390E+02	2.2819E+02
	RMSE	7.7657E+02	1.7007E+03	1.6455E+03	9.7606E+02	9.0433E+02	8.3815E+02	6.9823E+02	9.8100E+02	7.3428E+02
	δ	3.3570E+02	1.0347E+03	1.1577E+03	1.3530E+01	1.4415E+02	5.7938E+02	1.3401E+02	3.6208E+02	2.7838E+02
	p-value	6.3594E-01	9.1728E-08	6.7956E-08	6.3892E-02	6.3892E-02	4.6792E-02(-)	4.4075E-01	2.1393E-03	-
	Rank	3	9	8	6	5	4	(=)	(=)	(=)
	Mean	3.7564E+00	4.6200E+00	4.5252E+00	3.5066E+00	3.8096E+00	4.2280E+00	3.1851E+00	4.2268E+00	3.4481E+00
	Std	4.0597E-01	2.8458E-01	1.5400E-01	3.7929E-01	4.1170E-01	1.7666E-01	5.8822E-01	3.2486E-01	3.5771E-01
	RMSE	2.7847E+00	3.6306E+00	3.5284E+00	2.5337E+00	2.8381E+00	3.2325E+00	2.2591E+00	3.2422E+00	2.4728E+00
F6	δ	2.1739E+00	2.8953E+00	3.2523E+00	1.7779E+00	2.1326E+00	2.9306E+00	1.4067E+00	2.6261E+00	1.6146E+00
	p-value	2.2270E-02(-)	9.1728E-08	6.7956E-08	4.9033E-01	1.2345E-02(-)	3.4156E-07(-)	6.3892E-02	1.3761E-06	-
	Rank	3	9	8	6	5	4	(=)	(=)	(=)
	Mean	1.2793E+00	2.1239E+00	1.9236E+00	1.1901E+00	1.3686E+00	1.0906E+00	1.2837E+00	1.4555E+00	1.1494E+00
	Std	1.1867E-01	6.0660E-01	2.2326E-01	5.2680E-02	1.4846E-01	2.4319E-02	1.2550E-01	1.0668E-01	5.4451E-02
	RMSE	3.0230E-01	1.2700E+00	9.4886E-01	1.9688E-01	3.9602E-01	9.3663E-02	3.0893E-01	4.6725E-01	1.5851E-01
	δ	1.4145E-01	3.9893E-01	5.6960E-01	1.1749E-01	1.5715E-01	4.8080E-02	8.5075E-02	2.4993E-01	6.9320E-02
	p-value	1.6098E-04(-)	6.7956E-08	6.7956E-08	6.3892E-02	5.1658E-06(-)	7.5788E-04	1.1590E-04(-)	6.7956E-08	-
	Rank	4	9	8	3	6	1	5	7	2
F9	Mean	2.1028E+01	2.1448E+01	2.1449E+01	2.0549E+01	2.1375E+01	2.0842E+01	2.1373E+01	2.1004E+01	2.0248E+01
	Std	5.9834E-02	1.1028E-01	8.9325E-02	4.1447E+00	1.3942E-01	2.6330E+00	1.0345E-01	7.4133E-03	3.9251E+00
	RMSE	2.0028E+01	2.0449E+01	2.0449E+01	1.9962E+01	2.0375E+01	2.0007E+01	2.0373E+01	2.0004E+01	1.9625E+01
	δ	2.0000E+01	2.0186E+01	2.0295E+01	1.9434E+00	2.0018E+01	8.6589E+00	2.0183E+01	2.0000E+01	2.5799E+00
	p-value	2.9441E-02(-)	3.9388E-07	2.9598E-07	1.8030E-06(-)	7.5774E-06(-)	1.5757E-06(-)	5.1658E-06(-)	3.7051E-05	-
	Rank	5	8	9	2	7	3	6	4	1
	Mean Rank	4.3000	7.7000	7.7000	3.9000	4.5000	5.0000	2.5000	6.8000	2.3000
	Final Ranking	4	8	8	3	5	6	2	7	1
	+/-=	1/5/4	0/10/0	0/10/0	2/4/4	0/5/5	2/6/2	2/3/5	0/10/0	-

Table 8

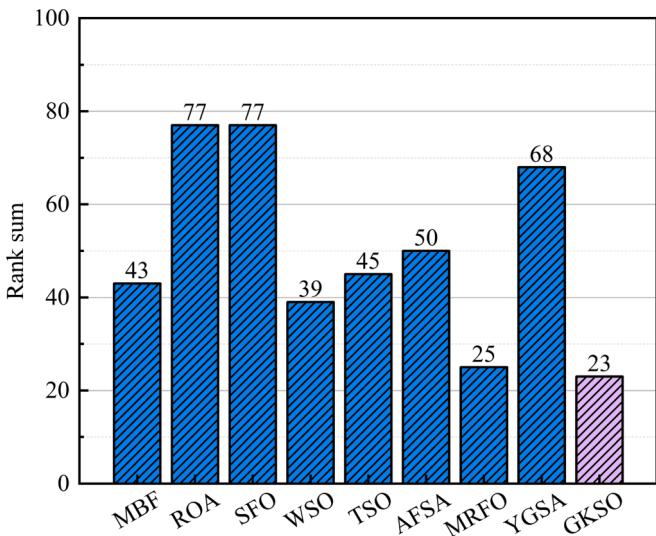
Sr and Me results of GKS0 and other FCAs on CEC2019.

F	Index	Algorithms								
		MBF	ROA	SFO	WSO	TSO	AFSA	MRFO	YSGA	
F1	<i>Sr</i>	0.0000	100.0000	100.0000	55.0000	100.0000	0.0000	100.0000	5.0000	100.0000
	<i>Me</i>	–	7.2750E + 03	2.4125E + 03	1.8382E + 04	7.1850E + 03	–	7.7250E + 02	2.2370E + 04	1.7263E + 04
F2	<i>Sr</i>	0.0000	100.0000	100.0000	0.0000	100.0000	0.0000	100.0000	0.0000	100.0000
	<i>Me</i>	–	6.2475E + 03	2.3225E + 03	–	5.7875E + 03	–	8.0750E + 02	–	8.8325E + 03
F3	<i>Sr</i>	95.0000	25.0000	10.0000	100.0000	100.0000	100.0000	100.0000	95.0000	100.0000
	<i>Me</i>	4.3909E + 03	3.2910E + 04	4.3850E + 04	1.4443E + 04	7.9850E + 03	1.6641E + 05	5.4825E + 03	9.9026E + 03	1.3183E + 04
F4	<i>Sr</i>	80.0000	0.0000	0.0000	85.0000	60.0000	15.0000	45.0000	10.0000	55.0000
	<i>Me</i>	7.0238E + 03	–	–	1.6579E + 04	4.0167E + 03	3.3902E + 05	6.5389E + 03	1.6000E + 04	1.1423E + 04
F5	<i>Sr</i>	100.0000	0.0000	0.0000	100.0000	100.0000	100.0000	100.0000	80.0000	100.0000
	<i>Me</i>	1.0438E + 03	–	–	4.6550E + 03	2.4625E + 03	1.6610E + 05	1.9825E + 03	1.5809E + 04	8.1525E + 03
F6	<i>Sr</i>	100.0000	0.0000	0.0000	100.0000	50.0000	40.0000	95.0000	0.0000	75.0000
	<i>Me</i>	2.8022E + 03	–	–	6.2950E + 03	2.1950E + 03	3.1664E + 04	4.7026E + 03	–	1.3241E + 04
F7	<i>Sr</i>	85.0000	0.0000	0.0000	55.0000	60.0000	90.0000	90.0000	50.0000	90.0000
	<i>Me</i>	4.0280E + 03	–	–	1.8805E + 04	7.5875E + 03	2.1710E + 05	8.8083E + 03	7.4570E + 03	1.1103E + 04
F8	<i>Sr</i>	65.0000	5.0000	0.0000	90.0000	60.0000	10.0000	85.0000	30.0000	95.0000
	<i>Me</i>	8.4938E + 03	1.2000E + 04	–	1.1811E + 04	8.7792E + 03	4.1628E + 05	1.0003E + 04	1.5623E + 04	1.8066E + 04
F9	<i>Sr</i>	65.0000	0.0000	0.0000	95.0000	40.0000	100.0000	70.0000	5.0000	100.0000
	<i>Me</i>	4.3628E + 03	–	–	7.8079E + 03	8.8188E + 03	2.2995E + 05	2.1304E + 04	3.6500E + 03	2.0033E + 04
F10	<i>Sr</i>	10.0000	0.0000	0.0000	5.0000	0.0000	5.0000	0.0000	75.0000	10.0000
	<i>Me</i>	2.6174E + 04	–	–	1.3150E + 04	–	2.9345E + 05	–	4.0527E + 04	6.9425E + 04

Table 9

MT (in seconds) of GKS0 and other FCAs on CEC2019.

F	Algorithms								
	MBF	ROA	SFO	WSO	TSO	AFSA	MRFO	YSGA	GKS0
F1	0.5758	0.2911	0.1756	0.1009	0.0990	14.3207	0.2227	0.3988	0.4485
F2	0.4893	0.1737	0.1585	0.0805	0.0737	1.5531	0.1799	0.3510	0.3223
F3	0.6407	0.1895	0.1814	0.0645	0.0690	6.2287	0.1822	0.3732	0.3361
F4	0.5128	0.2140	0.1449	0.0705	0.0742	6.1521	0.1866	0.3482	0.3497
F5	0.5138	0.2260	0.1501	0.0757	0.0805	4.0656	0.1962	0.3749	0.3666
F6	1.7032	2.8024	1.6192	0.8947	0.9651	145.7794	1.9770	2.0891	3.9780
F7	0.9397	0.3748	0.2460	0.1121	0.1579	16.2812	0.3838	0.5943	0.6291
F8	0.5289	0.2178	0.1404	0.0749	0.0855	3.3057	0.2010	0.3954	0.3513
F9	1.6749	0.5806	0.4140	0.2017	0.2716	12.3628	0.5734	1.1314	1.1078
F10	2.0045	0.7169	0.4965	0.1846	0.2496	9.3849	0.6311	1.4430	1.1516
MT	0.9584	0.5787	0.3727	0.1860	0.2126	21.9434	0.4734	0.7499	0.9041

**Fig. 10.** Column chart of rank sum obtained by GKS0 and other FCAs on CEC2019.

average fitness in iteration processes. The last column presents the optimal convergence curve achieved by the search agent so far. Below we will conduct a specific analysis of four qualitative indicators on two

test sets:

- Search history: The initial particles uniformly fill the entire problem space within a certain range. When the optimal concentration area is determined, the particles will converge to the optimum from various directions. When GKS0 addresses CEC2019, as shown in the particle distribution of F5 and F7 in Fig. 8, the density of particle distribution is dramatically distinct for different types functions in different phases of optimization processes. For test functions of CEC2022, similar scenarios can also be reflected on F4 and F10 in Fig. 9. These cases manifest that GKS0's capability to balance ENE varies when dealing with different OPs. From the general trend of particle changes, we detect that GKS0 has strong convergence in the search space, indicating that GKS0 has excellent development capabilities.

- 1st GKS's trajectory: In order to further study GKS0's characteristics, we take the movement trajectory of the first GKS in the 1st dimension as an example. From the trajectory curve, it can be seen that the first solution in the 1st dimension often has a high search frequency and major trajectory fluctuations in the early phase. When GKS0 addresses CEC2019, as shown in the trajectories of F1, F4, F6 and F8 in Fig. 8, there are significant mutations in the early stages of GKS0, with the magnitude of the changes almost covering the entire search space. For test functions of CEC2022, similar scenarios can also be reflected on F7 in Fig. 9. These cases indicate that GKS0 has good exploration ability. In addition, from the trajectory of F10 in Fig. 9, it can be seen that there is still dynamic activity of the search agent in the later phase of GKS0, which proves that the population update is still ongoing. For other complex functions, as the iterative process proceeds, the search

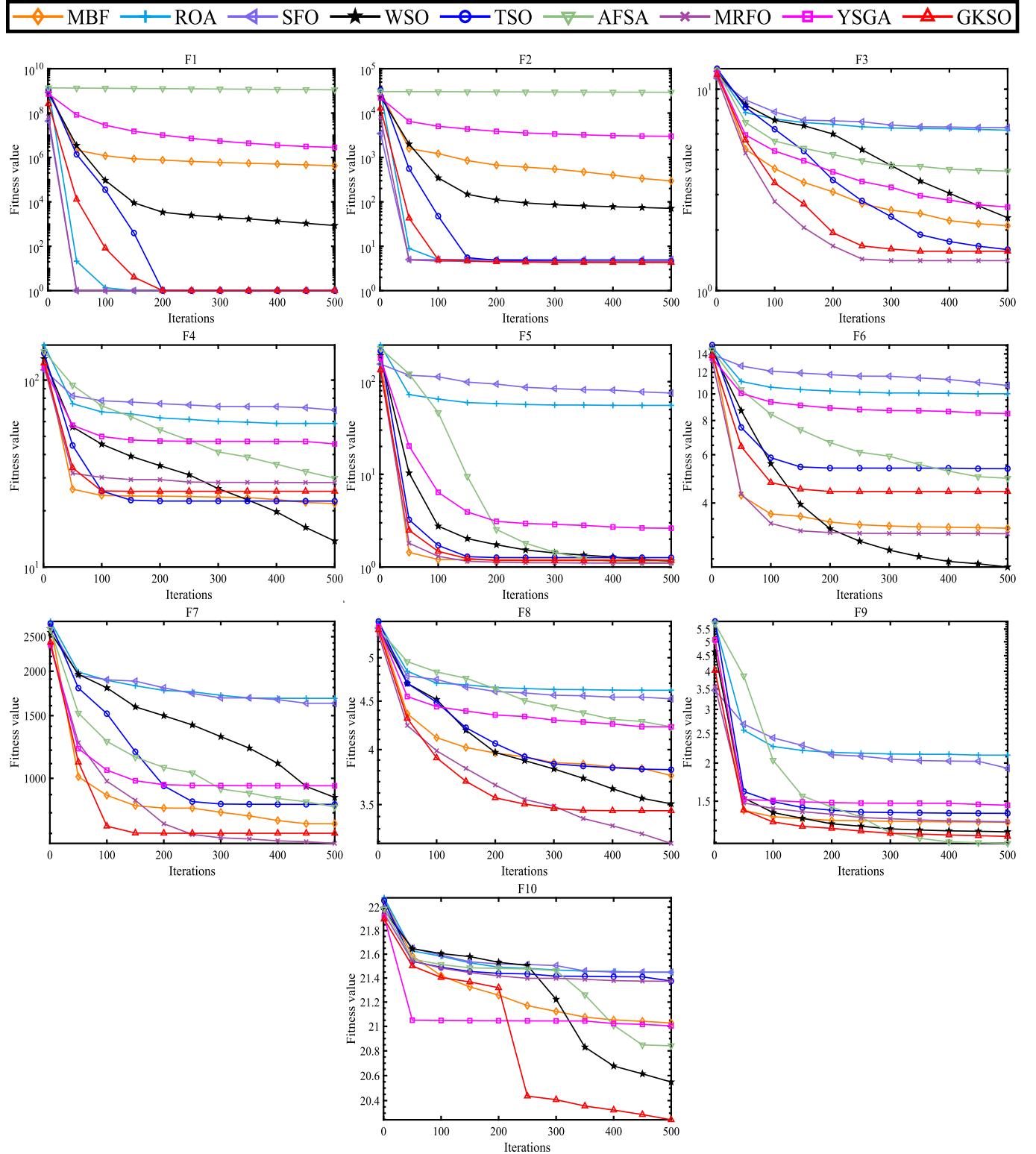


Fig. 11. Convergence curves of GKS0 and other FCAs for solving CEC2019.

amplitude of the agent will decline by degrees until the activity stops, which is consistent with the process of GKS0 switching from exploration to exploitation. As shown in trajectories of F3 in Fig. 8 and F8 in Fig. 9, during the pursuit of the optimum in the iterative later phase, the motion of GKS gradually stabilizes and converges to the optimum.

- Average fitness measurement: The average fitness curve reflects the average fitness value's variation of GKS0 during iteration processes.

From the curves in the fourth column of Fig. 8 and Fig. 9, search agents have different initial values during iterations, indicating that GKS0 maintains population diversity in the initial stage. In addition, during the iterative process of GKS0, the average fitness emerges a decreasing trend and converges to the optimum by degrees. Specifically, as shown in Fig. 8, the average fitness of GKS0 for solving F7 on CEC2019 continues to decline, which shows that the overall population of each

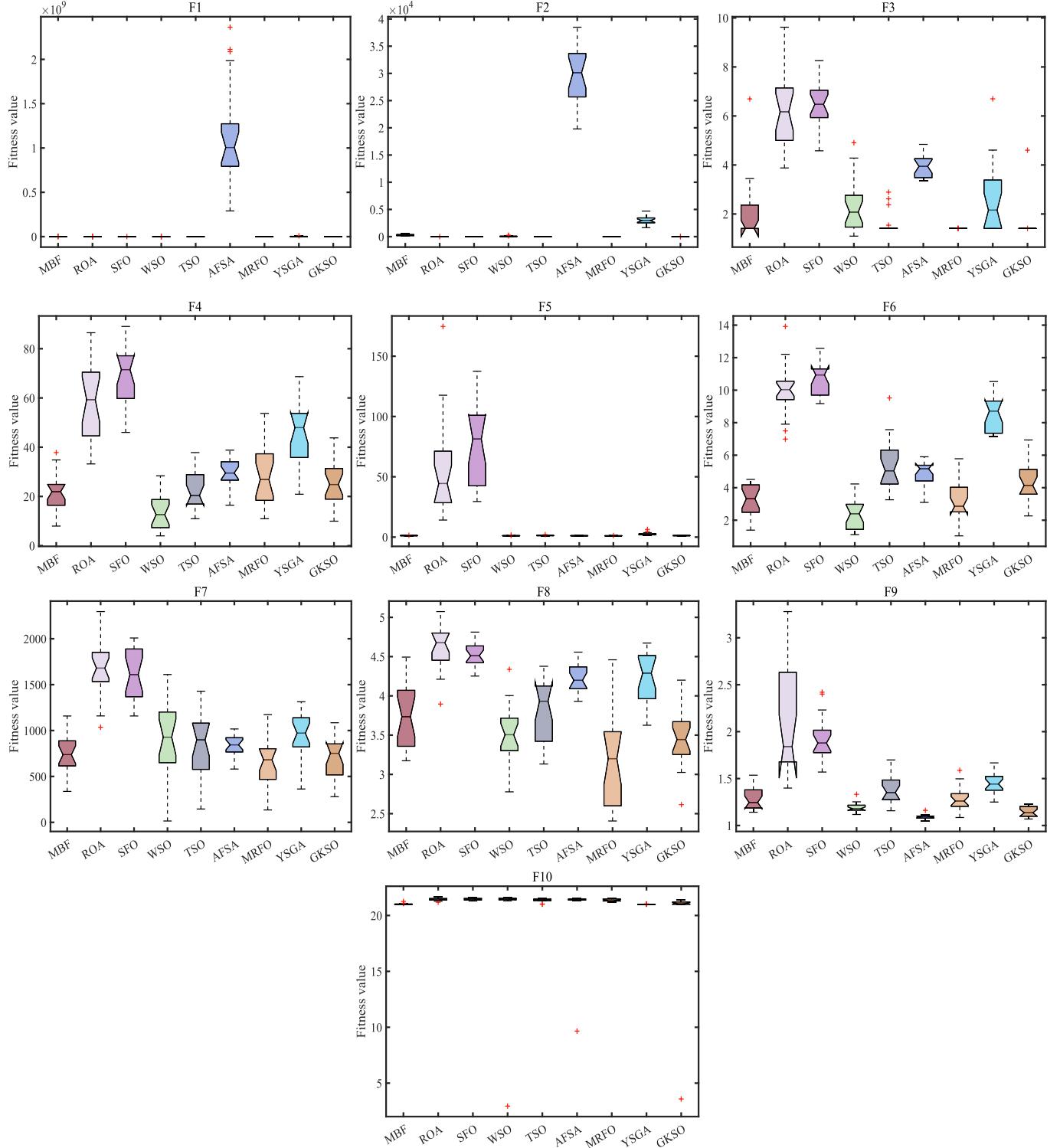


Fig. 12. Boxplots of GKS0 and other FCAs for solving CEC2019.

generation is constantly approaching the optimum.

- Convergence metric: Convergence curves reflect how GKS0 improves the quality of candidate solutions during iterations and converges towards the optimum by degrees. The convergence curve reflects the trend of fitness values, as shown in the last column of Fig. 8, there is a significant convergence stagnation phenomenon during the iteration process in addressing F4 and F9 on CEC2019, which may be due to a decrease in diversity among population, and leading to the gradual

updating of the optimal individuals in the population. In addition, GKS0 almost exhibits a cliff like convergence trend in solving F2, F3, and F5 on CEC2019, indicating that GKS0 has excellent search efficiency in the optimization process of problem space. This situation is also reflected in most functions of CEC2022.

Table 10Comparison of results between GKSO and OCAs on 10-dimensional CEC2022 ($N = 50$, $T = 500$).

Index	Algorithms										
	SCA	FFA	GTO	AVOA	MGO	ARO	BWO	BDO	COA	GKSO	
F1	Mean	1.9024E + 03	4.0222E + 02	3.0000E + 02	3.0819E + 02	3.0000E + 04	1.0498E + 03	5.4869E + 02	3.1552E + 03	6.2494E + 02	
	Std	1.1613E + 03	1.4604E + 02	5.8152E - 09	1.5272E + 01	2.2634E - 07	7.1476E + 03	2.0922E + 03	3.7651E + 01	1.5967E + 03	1.6901E - 09
	RMSE	1.9619E + 03	1.7525E + 02	5.8325E - 09	1.6989E + 01	2.3903E - 07	1.2351E + 04	5.5734E + 03	3.9845E + 01	6.1496E + 03	1.8206E - 09
	δ	5.1131E + 02	3.3827E + 00	5.6843E - 14	2.9028E - 05	5.6559E - 11	1.0074E + 03	1.6752E + 03	5.7380E - 03	2.5311E + 03	5.6843E - 13
	p-value	6.7956E - 08	(-)	6.7956E - 08	2.6898E - 06	6.7956E - 08	1.6098E - 04	6.7956E - 08	6.7956E - 08	6.7956E - 08	-
	Rank	7	6	2	4	3	10	8	5	9	1
F2	Mean	4.7433E + 02	4.2647E + 02	4.0845E + 02	4.0943E + 02	4.0914E + 02	4.2404E + 02	4.1884E + 02	4.2215E + 02	1.1717E + 03	4.0411E + 02
	Std	2.4966E + 01	2.4965E + 01	1.5392E + 01	1.5122E + 01	1.4856E + 01	2.7639E + 01	4.6297E + 00	2.8085E + 01	3.3852E + 02	3.6746E + 00
	RMSE	7.8211E + 01	3.5955E + 01	1.7220E + 01	1.7497E + 01	1.7124E + 01	3.6104E + 01	1.9373E + 01	3.5210E + 01	8.3931E + 01	5.4549E + 00
	δ	1.8582E + 01	9.1583E - 02	1.2796E - 05	7.1646E - 04	1.5995E - 01	4.7223E - 02	1.0081E + 01	4.8503E + 00	1.5716E + 02	1.2797E - 03
	p-value	6.7956E - 08	(-)	4.1550E - 04	4.7344E - 01	4.7348E - 01	1.1933E - 01	3.7051E - 05	6.7956E - 08	3.0553E - 03	6.7956E - 08
	Rank	9	8	2	4	3	7	5	6	10	1
F3	Mean	6.2115E + 02	6.0001E + 02	6.0511E + 02	6.1582E + 02	6.0008E + 02	6.0683E + 02	6.0769E + 02	6.1001E + 02	6.4056E + 02	6.0222E + 02
	Std	6.8669E + 00	2.6851E - 02	4.0634E + 00	1.2826E + 01	1.2909E - 01	9.6599E + 00	2.2649E + 00	6.7566E + 00	1.0467E + 01	2.0254E + 00
	RMSE	2.2182E + 01	2.7430E - 02	6.4680E + 00	2.0160E + 01	1.5146E - 01	1.1632E + 01	7.9977E + 00	1.1984E + 01	4.1825E + 01	2.9672E + 00
	δ	9.9290E + 00	1.7167E - 11	4.3196E - 01	7.6767E - 01	2.2737E - 13	1.4438E - 01	4.7738E + 00	9.9151E - 01	2.0863E + 01	5.7208E - 03
	p-value	6.7956E - 08	(-)	2.5629E - 07	9.7865E - 03	1.3761E - 06	5.2550E - 05	9.0907E - 02	3.4156E - 07	4.1658E - 05	6.7956E - 08
	Rank	9	1	4	8	2	5	6	7	10	3
F4	Mean	8.4441E + 02	8.2320E + 02	8.2278E + 02	8.2888E + 02	8.1328E + 02	8.3503E + 02	8.2607E + 02	8.2548E + 02	8.4740E + 02	8.2209E + 02
	Std	8.1175E + 00	8.1074E + 00	9.8913E + 00	1.2610E + 01	7.6749E + 01	1.0938E + 01	4.6906E + 00	1.1179E + 01	6.8730E + 00	8.1319E + 00
	RMSE	4.5107E + 01	2.4512E + 01	2.4740E + 01	3.1386E + 01	1.5244E + 01	3.6612E + 01	2.6463E + 01	2.7715E + 01	4.7869E + 01	2.3467E + 01
	δ	2.8278E + 01	8.9467E + 00	4.9748E + 00	7.9597E + 01	4.9748E + 01	1.6938E + 01	1.6667E + 01	1.1940E + 01	3.7009E + 01	8.9546E + 00
	p-value	3.9388E - 07	(-)	3.7933E - 01	6.8491E - 01	8.1032E - 02	8.3383E - 04	1.9971E - 04	2.2270E - 02	4.7348E - 01	1.9177E - 07
	Rank	9	4	3	7	1	8	6	5	10	2
F5	Mean	9.9920E + 02	9.0325E + 02	9.5187E + 02	1.2643E + 03	9.0605E + 02	1.7730E + 03	9.3958E + 02	9.8659E + 02	1.3046E + 03	9.0208E + 02
	Std	4.7107E + 01	7.7613E + 00	8.6896E + 01	2.6160E + 02	1.6486E + 01	5.0270E + 02	2.3934E + 01	1.0141E + 02	1.1967E + 02	3.5484E + 00
	RMSE	1.0931E + 02	8.2349E + 00	9.9319E + 01	4.4464E + 02	1.7171E + 01	1.0011E + 03	4.5947E + 01	1.3140E + 02	4.2112E + 02	4.0382E + 00
	δ	3.9813E + 01	4.4930E - 07	1.1528E + 00	2.2414E + 01	5.6843E - 13	1.8429E + 02	1.1142E + 01	9.0878E - 01	1.7814E + 02	9.3446E - 07
	p-value	6.7956E - 08	(-)	9.6763E - 01	3.9874E - 06	6.7956E - 08	3.5068E - 01	6.7956E - 08	9.1728E - 08	1.0473E - 06	6.7956E - 08
	Rank	7	2	5	8	3	10	4	6	9	1
F6	Mean	4.0419E + 06	1.9898E + 04	1.8792E + 03	3.6255E + 03	2.0479E + 03	4.8106E + 05	1.4919E + 05	5.1003E + 05	6.7207E + 06	2.6058E + 03
	Std	3.7211E + 06	7.0016E + 04	4.0408E + 01	1.1048E + 03	2.0870E + 02	2.5144E + 03	1.1513E + 05	2.5978E + 03	7.3191E + 06	1.2184E + 03
	RMSE	5.4292E + 06	7.0603E + 04	8.8412E + 01	2.1195E + 03	3.2064E + 02	3.8820E + 05	1.8525E + 03	4.1597E + 05	9.7997E + 06	1.4351E + 03
	δ	4.9133E + 04	1.7082E + 02	2.0177E + 01	1.1005E + 02	2.7314E + 01	1.0546E + 02	5.9848E + 03	1.9268E + 02	5.1669E + 04	7.1977E + 01
	p-value	6.7956E - 08	(-)	6.0403E - 03	1.8030E - 06	4.7025E - 03	2.2270E - 02	4.7025E - 03	6.7956E - 08	3.3819E - 04	6.7956E - 08
	Rank	9	7	1	4	2	5	8	6	10	3
F7	Mean	2.0630E + 03	2.0169E + 03	2.0295E + 03	2.0404E + 03	2.0194E + 03	2.0457E + 03	2.0320E + 03	2.0315E + 03	2.0888E + 03	2.0167E + 03
	Std	8.8412E + 00	1.1013E + 01	1.1474E + 01	2.1480E + 01	8.3594E + 00	3.1221E + 01	4.2007E + 01	1.3906E + 01	1.5704E + 01	1.0375E + 01
	RMSE	6.3625E + 01	2.0060E + 01	3.1532E + 01	4.5541E + 01	2.1033E + 01	5.4942E + 01	3.2258E + 01	3.4257E + 01	9.0092E + 01	1.9485E + 01

(continued on next page)

Table 10 (continued)

Index	Algorithms									
	SCA	FFA	GTO	AVOA	MGO	ARO	BWO	BDO	COA	GKSO
F8	δ 01	5.3282E + 6.2243E-04	1.3839E + 01	1.8310E + 01	3.0692E-03	1.3098E + 01	2.3488E + 01	1.9673E + 01	6.2286E + 01	1.1754E + 00
	p -value (-)	6.7956E-08	6.9489E-01	5.1153E-03	3.7051E-05	5.9786E-01	1.0141E-03	3.4995E-06	1.3486E-03	6.7956E-08
	Rank 9	2	4	7	3	8	6	5	10	1
	Mean	2.2334E + 03	2.2247E + 03	2.2199E + 03	2.2245E + 03	2.2207E + 03	2.2449E + 03	2.2253E + 03	2.2251E + 03	2.2337E + 03
	Std	3.3389E + 00	7.1397E + 00	5.6367E + 00	7.0525E + 00	6.3663E-01	4.8943E + 01	3.0494E + 00	3.2409E + 00	5.2832E + 00
	RMSE	3.3550E + 01	2.5703E + 01	2.0598E + 01	2.5423E + 01	2.0753E + 01	6.5507E + 01	2.5476E + 01	2.5281E + 01	3.4113E + 01
	δ 01	2.7769E + 00	8.0948E + 00	1.2014E + 00	3.9654E + 00	1.9906E + 01	1.8897E + 01	1.5645E + 01	2.0667E + 01	2.0917E + 01
	p -value (-)	6.7956E-08	2.2220E-04	9.6196E-02	7.5774E-06	8.1815E-01	5.5605E-03	5.1658E-06	2.0616E-06	1.6571E-07
	Rank 8	5	2	4	3	10	7	6	9	1
	Mean	2.5764E + 03	2.5320E + 03	2.5295E + 03	2.5297E + 03	2.5293E + 03	2.5392E + 03	2.5430E + 03	2.5409E + 03	2.7246E + 03
F9	Std	2.1109E + 01	3.0236E + 00	7.0828E-01	1.1221E + 00	6.5968E-04	1.7081E + 01	9.4890E + 00	3.2591E + 01	3.7938E + 01
	RMSE	2.7718E + 02	2.3205E + 02	2.2947E + 02	2.2968E + 02	2.2928E + 02	2.3974E + 02	2.4320E + 02	2.4299E + 02	4.2619E + 02
	δ 02	2.5332E + 02	2.2928E + 02	2.2928E + 02	2.2928E + 02	2.2930E + 02	2.3140E + 02	2.2928E + 02	3.5292E + 02	2.2928E + 02
	p -value (-)	6.7956E-08	9.1728E-08	7.1307E-06	1.1267E-08	4.4075E-01	6.7956E-08	6.7956E-08	3.1254E-02	6.7956E-08
	Rank 9	5	3	4	2	6	8	7	10	1
	Mean	2.5175E + 03	2.5005E + 03	2.5326E + 03	2.5471E + 03	2.5336E + 03	2.5512E + 03	2.5452E + 03	2.5715E + 03	2.6586E + 03
	Std	4.4885E + 01	2.1607E-01	5.7359E + 01	6.5338E + 01	5.2052E + 01	7.0853E + 01	6.2008E + 01	6.6061E + 01	1.1383E + 02
	RMSE	1.2533E + 02	1.0055E + 02	1.4395E + 02	1.6029E + 02	1.4294E + 02	1.6621E + 02	1.5729E + 02	1.8317E + 02	2.8136E + 02
	δ 02	1.0114E + 02	1.0034E + 02	1.0016E + 02	1.0041E + 02	1.0023E + 02	1.0049E + 02	1.0059E + 02	1.0042E + 02	1.2930E + 02
	p -value (-)	6.7956E-08	6.8682E-04	3.0566E-03	1.6571E-07	1.2941E-04	6.7956E-08	6.7956E-08	1.2346E-07	6.7956E-08
F11	Rank 3	2	4	7	5	8	6	9	10	1
	Mean	2.7838E + 03	2.7278E + 03	2.6350E + 03	2.6895E + 03	2.7052E + 03	2.8313E + 03	2.7176E + 03	2.7037E + 03	3.4201E + 03
	Std	1.7133E + 01	1.1071E + 02	9.7524E + 01	1.3813E + 02	1.1001E + 01	1.8938E + 02	1.9760E + 01	2.0606E + 02	3.2834E + 01
	RMSE	1.8459E + 02	1.6728E + 02	1.0131E + 02	1.6165E + 02	1.5024E + 02	2.9589E + 02	1.1914E + 02	2.2605E + 02	8.8031E + 01
	δ 02	1.6715E + 01	1.9698E + 01	9.0949E-13	6.1284E-06	4.5475E-13	3.8517E + 00	7.9377E + 01	2.6750E-05	3.2783E + 02
	p -value (-)	1.2009E-06	4.1658E-05	1.2912E-04	1.8074E-05	8.1808E-01	1.5757E-06	1.2941E-04	3.2931E-05	6.7956E-08
	Rank 8	7	2	3	5	9	6	4	10	1
	Mean	2.8710E + 03	2.8666E + 03	2.8645E + 03	2.8658E + 03	2.8645E + 03	2.8758E + 03	2.8655E + 03	2.8735E + 03	2.9292E + 03
	Std	1.7066E + 00	1.6103E + 00	3.0176E + 00	3.2423E + 00	2.0135E + 00	1.9954E + 01	1.2207E + 00	1.9802E + 01	3.8863E + 00
	RMSE	1.7104E + 02	1.6660E + 02	1.6448E + 02	1.6579E + 02	1.6448E + 02	1.7689E + 02	1.6547E + 02	1.7454E + 02	2.3231E + 02
F12	δ 02	1.6710E + 02	1.6459E + 02	1.6257E + 02	1.6214E + 02	1.5866E + 02	1.6521E + 02	1.6326E + 02	1.6214E + 02	1.7805E + 02
	p -value (-)	6.7956E-08	7.5774E-06	6.3585E-01	8.5855E-02	4.2486E-01	2.2178E-07	7.7118E-03	2.0729E-02	6.7956E-08
	Rank 7	6	2	5	2	9	4	8	10	1
	Mean Rank	7.8333	4.5833	2.8333	5.4167	2.8333	7.9167	6.1667	6.1667	9.7500
	Final Ranking	8	4	2	5	2	9	6	10	1
	+/-=	0/12/0	1/8/3	1/7/4	0/9/3	3/2/7	0/11/1	0/12/0	0/11/1	0/12/0

5.4. Quantitative analysis

Quantitative refers to attributes that exist in the form of quantity, so they can be measured. We have demonstrated through qualitative analysis that GKSO has good ENE capabilities, but GKSO's performance has not been sufficiently studied. In order to further verify the proposed GKSO's strengths and potential, In this section, we compare GKSO with other FCAs and OCAs on CEC2019 and CEC2022 separately, and

introduce statistical metrics to quantitatively analyze GKSO's performance. A summary of statistical results is shown in Table 7-Table 15.

5.4.1. Comparative evaluation of GKSO and other FCAs on CEC2019

Tables 7 to 9 present the comparison results between GKSO and FCAs on CEC2019, with the optimal results are displayed in bold (the same below). As shown in Table 7, we spot that GKSO has a mean rank of 2.3, ranking 1st among all FCAs, followed closely by MRFO and WSO, with

Table 11Sr and Me results of GKS0 and OCAs on 10-dimensional CEC2022 ($N = 50$, $T = 500$).

Index		Algorithms									
		SCA	FFA	GTO	AVOA	MGO	ARO	BWO	BDO	COA	GKS0
F1	<i>Sr</i>	0.0000	45.0000	100.0000	85.0000	100.0000	0.0000	0.0000	90.0000	0.0000	100.0000
	<i>Me</i>	–	6.5039E + 04	1.8763E + 04	1.7041E + 04	6.0250E + 03	–	–	2.2036E + 04	–	2.6913E + 04
F2	<i>Sr</i>	5.0000	50.0000	95.0000	90.0000	95.0000	65.0000	55.0000	80.0000	0.0000	100.0000
	<i>Me</i>	7.1500E + 03	1.0025E + 04	9.3921E + 03	6.6528E + 03	2.3974E + 03	1.1208E + 04	3.1205E + 04	7.8875E + 03	–	1.3923E + 04
F3	<i>Sr</i>	0.0000	100.0000	60.0000	10.0000	100.0000	55.0000	10.0000	30.0000	0.0000	90.0000
	<i>Me</i>	–	6.9475E + 03	1.3446E + 04	1.0450E + 04	1.1950E + 03	1.0755E + 04	3.7975E + 04	1.5442E + 04	–	2.0136E + 04
F4	<i>Sr</i>	0.0000	30.0000	40.0000	30.0000	85.0000	5.0000	10.0000	40.0000	0.0000	40.0000
	<i>Me</i>	–	3.7858E + 04	7.1688E + 03	2.1750E + 03	2.1971E + 03	1.5000E + 04	3.2825E + 04	1.8731E + 04	–	1.1881E + 04
F5	<i>Sr</i>	0.0000	80.0000	25.0000	0.0000	90.0000	0.0000	0.0000	15.0000	0.0000	90.0000
	<i>Me</i>	–	1.9241E + 04	1.2570E + 04	–	2.3944E + 03	–	–	1.5617E + 04	–	2.1792E + 04
F6	<i>Sr</i>	0.0000	40.0000	100.0000	25.0000	100.0000	35.0000	0.0000	25.0000	0.0000	70.0000
	<i>Me</i>	–	1.7950E + 04	1.8433E + 04	3.4200E + 03	2.0625E + 03	7.2000E + 03	–	9.2000E + 03	–	3.7846E + 04
F7	<i>Sr</i>	0.0000	90.0000	65.0000	45.0000	100.0000	50.0000	35.0000	65.0000	0.0000	95.0000
	<i>Me</i>	–	1.5828E + 04	1.1127E + 04	9.4944E + 03	3.4750E + 03	1.3260E + 04	2.8664E + 04	1.1131E + 04	–	1.3887E + 04
F8	<i>Sr</i>	0.0000	15.0000	10.0000	10.0000	10.0000	0.0000	10.0000	0.0000	0.0000	20.0000
	<i>Me</i>	–	3.2217E + 04	5.6750E + 03	9.6750E + 03	1.2250E + 04	–	2.6275E + 04	–	–	1.6813E + 04
F9	<i>Sr</i>	0.0000	45.0000	95.0000	85.0000	100.0000	55.0000	0.0000	70.0000	0.0000	100.0000
	<i>Me</i>	–	1.4322E + 04	1.0613E + 04	1.0250E + 04	1.6525E + 03	1.4394E + 04	–	9.5500E + 03	–	1.7343E + 04
F10	<i>Sr</i>	90.0000	100.0000	75.0000	65.0000	70.0000	65.0000	65.0000	45.0000	5.0000	100.0000
	<i>Me</i>	3.4500E + 03	2.6575E + 03	1.4906E + 03	1.2955E + 03	2.0357E + 02	1.4769E + 03	1.3038E + 03	4.7938E + 02	1.8000E + 03	4.5025E + 03
F11	<i>Sr</i>	0.0000	15.0000	85.0000	55.0000	45.0000	20.0000	0.0000	75.0000	0.0000	85.0000
	<i>Me</i>	–	2.7267E + 04	1.3650E + 04	5.6750E + 03	1.7500E + 03	1.5350E + 04	–	1.3573E + 04	–	1.3803E + 04
F12	<i>Sr</i>	0.0000	10.0000	75.0000	45.0000	60.0000	0.0000	50.0000	40.0000	0.0000	65.0000
	<i>Me</i>	–	2.2225E + 04	1.3073E + 04	4.0667E + 03	2.2208E + 03	–	3.1005E + 04	8.0000E + 03	–	1.3850E + 04

Table 12MT (in seconds) of GKS0 and OCAs on 10-dimensional CEC2022 ($N = 50$, $T = 500$).

	Algorithms									
	SCA	FFA	GTO	AVOA	MGO	ARO	BWO	BDO	COA	GKS0
F1	0.0938	0.3358	0.2162	0.1250	0.9998	0.0988	0.1295	0.0615	0.1382	0.4974
F2	0.0781	0.3447	0.2039	0.1132	0.9488	0.1025	0.1309	0.0584	0.1345	0.4426
F3	0.1094	0.4033	0.2732	0.1500	1.1583	0.1404	0.1677	0.0890	0.2170	0.6049
F4	0.0938	0.3553	0.2293	0.1303	1.1262	0.1306	0.1573	0.0754	0.1733	0.5589
F5	0.0938	0.3819	0.2385	0.1328	1.0403	0.1160	0.1510	0.0743	0.1815	0.5575
F6	0.0781	0.3679	0.2477	0.1204	1.0864	0.1140	0.1345	0.0625	0.1571	0.4898
F7	0.1406	0.4155	0.3273	0.1873	1.2947	0.1655	0.1970	0.1115	0.2743	0.6934
F8	0.1719	0.4966	0.3890	0.2011	1.4118	0.1878	0.2634	0.1454	0.3494	0.9388
F9	0.1719	0.4162	0.2978	0.1594	1.2182	0.1431	0.1723	0.1029	0.2436	0.6427
F10	0.1094	0.3579	0.2933	0.1485	1.1554	0.1392	0.1700	0.1020	0.2345	0.5951
F11	0.1250	0.4295	0.3324	0.1721	1.1863	0.1548	0.1941	0.1339	0.2981	0.6929
F12	0.1563	0.4194	0.3469	0.1827	1.2704	0.1603	0.2039	0.1269	0.3055	0.7269
MT	0.1185	0.3937	0.2830	0.1519	1.1581	0.1378	0.1726	0.0953	0.2256	0.6201

mean ranks of 2.5 and 3.9, respectively. Among the 10 benchmark functions, GKS0 outperforms other FCAs on three functions of F1, F2, and F10, all achieving a smaller mean value. Furthermore, GKS0 also achieves a smaller standard deviation (Std) while obtaining optimal results in F1 and F2, which may be the aggregation effect of GKS0 moving to the best hunting position. This enables agents to approach the optimal position from all directions, highlighting the exploitation ability of GKS0. In particular, GKS0, TSO and MRFO obtain completely consistent data on F1, all reaching the TO of the problem. Therefore, these algorithms demonstrate outstanding optimization ability when facing F1. While MRFO obtains a small mean value on five functions of F1, F3, F5, F7 and F8. Being able to stand out among half of the functions

is due to the three special foraging operations of MRFO, which improves the search ability and convergence rate of MRFO. But it performs poorly on F10, only finishing 6th. Meanwhile, WSO achieves 1st in F4 and F6, thanks to its combination of promising exploratory and exploitative searches in its update mechanism to stochastically update and generate new candidate solutions. It is worth mentioning that AFSA, which ranks 6th, achieves 1st on F9. As one of the classic algorithms, AFSA still has a certain competitiveness in competitions with other advanced FCAs. Although the overall performance of the proposed GKS0 in FCAs is relatively good, its performance on F4 to F6 is relatively weak. Other methods to improve GKS0 performance can be considered in the future.

In addition to investigating the convergence accuracy of GKS0 and

Table 13Comparison of results between GKSO and OCAs on 20-dimensional CEC2022 ($N = 100$, $T = 1000$).

Index	Algorithms									
	SCA	FFA	GTO	AVOA	MGO	ARO	BWO	BDO	COA	GKSO
F1	Mean	1.0844E + 04	5.4497E + 03	3.0000E + 02	3.2332E + 02	3.0000E + 02	3.1638E + 04	2.6841E + 04	2.7053E + 03	3.1929E + 04
	Std	3.4380E + 03	2.1366E + 03	2.2488E - 06	4.9022E + 01	4.1290E - 05	9.6961E + 03	6.1553E + 03	1.8306E + 03	4.6905E + 03
	RMSE	1.1063E + 04	5.5548E + 03	2.2625E - 06	5.3168E + 01	4.4211E - 05	3.2732E + 04	2.7211E + 04	2.9948E + 03	3.1957E + 04
	δ	5.7630E + 03	1.3816E + 03	2.7296E - 10	4.6401E - 05	1.7611E - 07	1.3639E + 04	1.7156E + 04	3.6219E + 02	2.4128E + 04
	p-value	6.7956E - 08	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(-)
	Rank	7	6	2	4	3	9	8	5	10
F2	Mean	6.7052E + 02	4.5862E + 02	4.4437E + 02	4.5695E + 02	4.4560E + 02	4.7157E + 02	4.9438E + 02	4.6901E + 03	1.9169E + 02
	Std	5.7100E + 01	1.1850E + 01	2.4020E + 01	1.6929E + 01	2.0873E + 01	2.2159E + 01	1.9246E + 01	3.7088E + 01	3.9151E + 02
	RMSE	2.7618E + 02	5.9743E + 01	5.0164E + 01	5.9296E + 01	4.9930E + 01	7.4758E + 01	9.6222E + 01	7.7908E + 01	1.5642E + 01
	δ	1.6327E + 02	4.9104E + 01	9.7488E - 01	6.7462E + 00	1.2861E - 01	3.8690E + 01	7.5315E + 01	7.0314E + 00	8.0571E + 02
	p-value	6.7956E - 08	(-)	(-)	(=)	(=)	(-)	(-)	(-)	(-)
	Rank	9	5	2	4	3	7	8	6	10
F3	Mean	6.3784E + 02	6.0000E + 02	6.2471E + 02	6.3380E + 02	6.0256E + 02	6.0422E + 02	6.1098E + 02	6.2576E + 02	6.7008E + 02
	Std	6.5764E + 00	1.0143E - 02	8.3803E + 00	1.3346E + 01	3.4062E + 00	2.8490E + 00	2.2564E + 00	1.7623E + 01	1.1923E + 01
	RMSE	3.8380E + 01	1.0257E - 02	2.6024E + 01	3.6215E + 01	4.1945E + 00	5.0545E + 00	1.1196E + 01	3.0963E + 01	7.1040E + 01
	δ	2.6698E + 01	3.0695E - 12	7.3647E + 00	4.3809E + 00	6.9251E - 02	8.0325E - 01	7.1823E + 00	2.5519E + 00	4.7015E + 01
	p-value	6.7956E - 08	(-)	(+)	(-)	(-)	(+)	(+)	(=)	(-)
	Rank	9	1	6	8	2	3	4	7	10
F4	Mean	9.3363E + 02	8.9651E + 02	8.8079E + 02	8.8022E + 02	8.3540E + 02	9.0595E + 02	8.8376E + 02	8.9289E + 02	9.5480E + 02
	Std	1.3934E + 01	1.2977E + 01	1.4165E + 01	1.8021E + 01	7.7944E + 01	3.5101E + 01	7.8370E + 02	3.1322E + 01	2.0185E + 01
	RMSE	1.3432E + 02	9.7337E + 01	8.1962E + 01	8.2122E + 01	3.6204E + 01	1.1134E + 02	8.4103E + 01	9.7775E + 01	1.5605E + 02
	δ	1.0952E + 02	6.2916E + 01	5.4723E + 01	5.6712E + 01	2.0894E + 01	4.5818E + 01	6.8226E + 01	2.5872E + 01	1.0663E + 02
	p-value	6.7956E - 08	(-)	(-)	(-)	(=)	(+)	(-)	(-)	(-)
	Rank	9	7	4	3	1	8	5	6	10
F5	Mean	1.9324E + 03	9.0710E + 02	1.6538E + 03	2.3565E + 03	1.0490E + 03	3.4894E + 03	1.9168E + 03	1.5561E + 03	3.0937E + 03
	Std	2.6305E + 02	9.4779E + 00	3.8501E + 02	3.9923E + 02	1.8515E + 02	8.0180E + 02	5.0806E + 02	4.0628E + 02	3.7157E + 02
	RMSE	1.0638E + 03	1.1653E + 01	8.4209E + 02	1.5076E + 03	2.3405E + 02	2.7047E + 03	1.1310E + 03	7.6632E + 02	2.2234E + 03
	δ	6.1039E + 02	5.7205E - 01	1.7179E + 02	7.0037E + 00	3.3618E + 03	1.3748E + 01	2.0916E + 02	1.0611E + 03	1.5631E + 03
	p-value	1.0473E - 06	(-)	(+)	(-)	(=)	(+)	(-)	(-)	(-)
	Rank	7	1	5	8	2	10	6	4	9
F6	Mean	1.1871E + 08	4.0016E + 05	2.0356E + 03	7.5178E + 03	5.2853E + 03	1.4465E + 04	1.5499E + 06	1.6803E + 05	1.2224E + 09
	Std	4.7776E + 07	8.7199E + 05	1.3597E + 02	5.9663E + 02	3.2061E + 03	1.2894E + 04	5.2132E + 05	4.6466E + 05	7.0527E + 08
	RMSE	1.2707E + 08	9.1816E + 05	2.6862E + 02	8.0455E + 02	4.6258E + 03	1.7608E + 04	1.6252E + 05	4.7112E + 05	1.3935E + 09
	δ	4.7663E + 07	3.9778E + 02	5.4347E + 01	1.7536E + 02	8.2075E + 01	1.5691E + 03	5.8305E + 05	3.1333E + 03	2.6237E + 08
	p-value	6.7956E - 08	(-)	(-)	(+)	(-)	(-)	(-)	(-)	(-)
	Rank	9	7	1	5	4	3	5	8	6
F7	Mean	2.1279E + 03	2.0536E + 03	2.0965E + 03	2.1313E + 03	2.0572E + 03	2.1102E + 03	2.0672E + 03	2.0908E + 03	2.1681E + 03
	Std	1.8166E + 01	1.3278E + 01	3.8389E + 01	5.1370E + 01	1.5309E + 01	5.1165E + 01	1.0121E + 01	4.2236E + 01	3.3674E + 01
	RMSE	1.2913E + 02	5.5154E + 01	1.0354E + 02	1.4050E + 02	5.9069E + 01	1.2093E + 02	6.7961E + 01	9.9733E + 01	1.7129E + 02

(continued on next page)

Table 13 (continued)

Index	Algorithms									
	SCA	FFA	GTO	AVOA	MGO	ARO	BWO	BDO	COA	GKSO
F8	δ 01	9.5581E + 01	3.9256E + 01	4.5489E + 01	5.3916E + 01	2.7851E + 01	4.3439E + 01	5.0139E + 01	3.9157E + 01	1.2239E + 01
	p-value	8.5974E-06 (-)	4.5695E-01 (=)	4.7025E-03 (-)	5.8959E-05 (-)	8.8173E-01 (=)	2.3413E-03 (=)	1.0173E-01 (=)	3.8515E-02 (-)	6.0148E-07 (-)
	Rank	8	1	6	9	2	7	4	5	10
	Mean	2.2561E + 03	2.2363E + 03	2.2500E + 03	2.2369E + 03	2.2225E + 03	2.2961E + 03	2.2285E + 03	2.2494E + 03	2.3199E + 03
	Std	7.3476E + 00	4.6107E + 00	4.9425E + 01	1.6589E + 01	3.7631E + 00	7.1147E + 01	1.6401E + 00	3.0295E + 01	8.2016E + 01
	RMSE	5.6509E + 01	3.6556E + 01	6.9445E + 01	4.0281E + 01	2.2830E + 01	1.1851E + 02	2.8572E + 01	5.7522E + 01	1.4413E + 02
	δ 01	3.6968E + 01	2.7754E + 01	2.1867E + 01	2.1545E + 01	2.0638E + 01	2.1587E + 01	2.5563E + 01	2.3574E + 01	3.8729E + 01
	p-value	1.2346E-07 (-)	2.9249E-05 (-)	3.6388E-03 (-)	3.3819E-04 (-)	1.8058E-01 (=)	1.2941E-04 (=)	1.2272E-03 (-)	7.5774E-06 (-)	7.8980E-08 (-)
	Rank	8	4	7	5	1	9	3	6	10
F9	Mean	2.5598E + 03	2.4815E + 03	2.4808E + 03	2.4808E + 03	2.4809E + 03	2.4876E + 03	2.4883E + 03	2.4948E + 03	3.0510E + 03
	Std	1.6948E + 01	1.1232E + 00	9.2068E-03 02	1.8002E-03 02	1.2000E-01 02	5.6729E + 00	3.5027E + 00	1.9962E + 01	2.1876E + 02
	RMSE	2.6037E + 02	1.8152E + 02	1.8079E + 02	1.8078E + 02	1.8088E + 02	1.8766E + 02	1.8832E + 02	1.9580E + 02	7.8069E + 02
	δ 02	2.3967E + 02	1.8078E + 02	1.8078E + 02	1.8078E + 02	1.8078E + 02	1.8083E + 02	1.8338E + 02	1.8080E + 02	4.4011E + 02
	p-value	6.7956E-08 (-)	1.2346E-07 (-)	3.9388E-07 (-)	4.4073E-01 (=)	1.3761E-06 (-)	6.7956E-08 (-)	6.7956E-08 (-)	6.7956E-08 (-)	6.7956E-08 (-)
	Rank	9	5	3	2	4	6	7	8	10
	Mean	2.5746E + 03	2.5694E + 03	2.6066E + 03	2.8008E + 03	2.5895E + 03	2.6498E + 03	2.5294E + 03	2.6857E + 03	4.6329E + 03
	Std	1.7163E + 02	1.4685E + 02	3.9867E + 02	3.2994E + 02	2.5745E + 02	1.9425E + 02	6.9096E + 01	5.6131E + 02	1.6932E + 01
	RMSE	2.4178E + 02	2.2181E + 02	4.4006E + 02	5.1388E + 02	3.1447E + 02	3.1348E + 02	1.4589E + 02	6.1720E + 02	2.7766E + 02
F10	δ 02	1.1511E + 02	1.0045E + 02	1.0038E + 02	3.4352E + 01	1.0037E + 02	1.9819E + 01	1.0095E + 02	1.0057E + 02	1.9156E + 02
	p-value	8.2924E-05 (-)	5.0907E-04 (-)	5.0907E-04 (-)	1.7936E-04 (-)	7.1135E-03 (-)	6.6104E-05 (-)	7.4064E-05 (-)	1.0373E-04 (-)	2.5629E-07 (-)
	Rank	4	3	6	9	5	7	2	8	10
	Mean	4.7439E + 03	3.0106E + 03	2.8950E + 03	3.0108E + 03	2.9050E + 03	3.0058E + 03	3.2987E + 03	3.0649E + 03	7.4210E + 03
	Std	3.5155E + 02	1.9182E + 02	1.0990E + 02	5.4741E + 02	2.2361E + 02	2.3733E + 01	8.4226E + 01	3.9613E + 02	8.2535E + 01
	RMSE	2.1711E + 03	4.5119E + 02	3.1385E + 02	6.7339E + 02	3.0578E + 02	4.6709E + 02	7.0353E + 02	6.0434E + 02	4.8876E + 02
	δ 03	1.2697E + 00	3.6843E + 00	4.2292E-11 (+)	1.3874E-05 (=)	3.0000E + (+)	1.7635E + 02	5.0464E + 01	2.1736E-01 02	3.3387E + 03
	p-value	6.7956E-08 (-)	8.3572E-04 (-)	1.6253E-03 (+)	4.9033E-01 (=)	6.1783E-06 (+)	1.9292E-02 (-)	6.9166E-07 (-)	7.7118E-03 (-)	6.7956E-08 (-)
	Rank	9	5	1	6	2	4	8	7	10
F12	Mean	3.0432E + 03	2.9494E + 03	2.9911E + 03	2.9889E + 03	2.9621E + 03	3.0014E + 03	2.9566E + 03	3.0075E + 03	3.4809E + 03
	Std	2.3826E + 01	1.3062E + 01	6.2517E + 01	3.9398E + 01	1.6297E + 01	4.0530E + 01	8.2981E + 00	4.4977E + 01	1.7369E + 02
	RMSE	3.4403E + 02	2.4971E + 02	2.9737E + 02	2.9142E + 02	2.6261E + 02	3.0394E + 02	2.5676E + 02	3.1066E + 02	7.9900E + 02
	δ 02	2.9915E + 02	2.3419E + 02	2.4485E + 02	2.4909E + 02	2.4144E + 02	2.5823E + 02	2.4158E + 02	2.3961E + 02	5.1459E + 02
	p-value	6.7956E-08 (-)	5.1153E-03 (+)	6.7868E-02 (=)	3.6388E-03 (-)	9.4608E-01 (=)	3.3819E-04 (-)	5.9786E-01 (=)	3.7499E-04 (-)	6.7956E-08 (-)
	Rank	9	1	6	5	4	7	2	8	10
	Mean Rank	8.0833	3.8333	4.0833	5.5833	2.6667	6.8333	5.4167	6.3333	9.9167
	Final Ranking	9	3	4	6	2	8	5	7	10
	+/-=	0/12/0	3/8/1	2/8/2	0/8/4	4/4/4	1/11/0	1/9/2	0/11/1	0/12/0

other FCAs, we also provide a comparison of root mean square error (RMSE) and relative error (δ) of GKSO and other FCAs on different test functions. Both RMSE and δ reflect the degree of approximation between the data and the TO, and the smaller the value of both, the smaller the error in the algorithm results and the better the stability of the algorithm. From the error data presented in Table 7, it can be seen that GKSO achieves smaller RMSE and δ values on most functions, indicating that GKSO has a high exactitude to ensure that all observations are closer to

TO.

Combining the ranking of FCAs and data, Table 7 also displays the Wilcoxon rank sum test (WRST) outcomes of GKSO compared with other FCAs under the condition of significance level $\alpha = 0.05$. The symbol “–” indicates the number of FCAs inferior to GKSO; “+” is the quantity with the opposite effect of “–”; “=” represents numbers of GKSO and other FCAs with similar performance. According to the p-value data in Table 7, the number of functions superior/inferior/similar to GKSO for each FCA

Table 14Sr and Me results of GKS0 and OCAs on 20-dimensional CEC2022 ($N = 100$, $T = 1000$).

Index	Algorithms									
	SCA	FFA	GTO	AVOA	MGO	ARO	BWO	BDO	COA	GKS0
F1	Sr	0.0000	0.0000	100.0000	75.0000	100.0000	0.0000	0.0000	0.0000	100.0000
	Me	-	-	1.1578E + 05	7.7880E + 04	3.3790E + 04	-	-	-	1.4639E + 05
F2	Sr	0.0000	50.0000	70.0000	55.0000	80.0000	10.0000	0.0000	45.0000	0.0000
	Me	-	1.4077E + 05	1.0169E + 05	3.3618E + 04	1.0564E + 04	7.4350E + 04	-	7.6878E + 04	1.2817E + 05
F3	Sr	0.0000	100.0000	5.0000	10.0000	95.0000	95.0000	30.0000	25.0000	0.0000
	Me	-	2.3300E + 04	3.1500E + 04	6.6000E + 04	5.5895E + 03	3.6563E + 04	1.3698E + 05	5.1740E + 04	7.4100E + 04
F4	Sr	0.0000	5.0000	25.0000	40.0000	100.0000	15.0000	0.0000	25.0000	0.0000
	Me	-	6.9700E + 04	2.2540E + 04	8.3125E + 03	4.0100E + 03	2.6333E + 04	-	8.6940E + 04	3.5130E + 04
F5	Sr	0.0000	100.0000	0.0000	0.0000	65.0000	0.0000	0.0000	0.0000	15.0000
	Me	-	5.2820E + 04	-	-	6.7462E + 03	-	-	-	5.0300E + 04
F6	Sr	0.0000	10.0000	100.0000	40.0000	60.0000	30.0000	0.0000	10.0000	0.0000
	Me	-	7.6250E + 04	9.6585E + 04	1.0600E + 04	6.6417E + 03	8.6850E + 04	-	1.5950E + 04	4.7656E + 04
F7	Sr	0.0000	85.0000	15.0000	5.0000	70.0000	30.0000	60.0000	40.0000	0.0000
	Me	-	1.1757E + 05	2.3300E + 04	4.5700E + 04	1.1414E + 04	4.1450E + 04	1.0779E + 05	6.4175E + 04	4.0150E + 04
F8	Sr	0.0000	10.0000	65.0000	50.0000	95.0000	35.0000	80.0000	20.0000	0.0000
	Me	-	1.6655E + 05	3.6362E + 04	5.3740E + 04	1.0405E + 04	1.7529E + 04	1.1147E + 05	5.7275E + 04	6.3331E + 04
F9	Sr	0.0000	45.0000	100.0000	85.0000	100.0000	5.0000	0.0000	50.0000	0.0000
	Me	-	7.2944E + 04	5.1815E + 04	5.7543E + 04	1.5829E + 04	7.0000E + 04	-	4.8825E + 04	9.7690E + 04
F10	Sr	90.0000	75.0000	85.0000	40.0000	80.0000	55.0000	85.0000	85.0000	0.0000
	Me	5.1965E + 04	3.0027E + 04	2.4865E + 04	3.2300E + 04	1.4063E + 03	1.5800E + 04	1.1735E + 04	6.1935E + 04	5.8176E + 03
F11	Sr	0.0000	15.0000	75.0000	90.0000	95.0000	10.0000	0.0000	35.0000	0.0000
	Me	-	8.1433E + 04	8.1636E + 04	4.3022E + 04	2.0453E + 04	4.5550E + 04	-	8.6514E + 04	9.9064E + 04
F12	Sr	0.0000	90.0000	45.0000	25.0000	65.0000	20.0000	85.0000	20.0000	0.0000
	Me	-	6.1722E + 04	2.5300E + 04	2.3680E + 04	2.2538E + 03	2.4550E + 04	8.1947E + 04	3.2200E + 04	2.9275E + 04

Table 15MT (in seconds) of GKS0 and OCAs on 20-dimensional CEC2022 ($N = 100$, $T = 1000$).

Algorithms										
	SCA	FFA	GTO	AVOA	MGO	ARO	BWO	BDO	COA	GKS0
F1	0.4219	0.6333	0.9913	0.5309	4.4616	0.4744	0.6771	0.2903	0.7013	4.1244
F2	0.3906	0.6967	0.9603	0.5072	4.4183	0.4704	0.6291	0.2603	0.5919	3.7189
F3	0.7188	0.8194	1.4534	0.7800	5.5112	0.8235	0.9323	0.5318	1.2372	4.6509
F4	0.4688	0.6928	1.0664	0.5672	4.5606	0.5367	0.6762	0.3269	0.7560	3.5868
F5	0.4844	0.6787	1.0781	0.5634	4.4701	0.5305	0.6758	0.3331	0.7705	3.4395
F6	0.5625	0.6571	1.0035	0.5207	4.4760	0.4798	0.6225	0.2700	0.6331	3.2877
F7	1.0313	1.1046	1.9858	1.0221	6.8227	0.9869	1.2035	0.7241	1.7917	6.3433
F8	0.8906	0.9655	2.0863	1.0568	6.6490	0.9987	1.2007	0.7776	1.8695	5.9017
F9	0.8750	0.9959	1.7846	0.9176	6.2240	0.8420	1.0298	0.6347	1.5218	5.3147
F10	0.7188	0.8709	1.6601	0.8551	5.9001	0.7564	0.9781	0.5557	1.3521	4.5867
F11	1.0469	1.1332	2.1329	1.1591	7.3008	1.0841	1.3123	0.8937	2.0734	6.2583
F12	1.0313	1.0356	2.0989	1.0928	6.7031	1.0672	1.2582	0.8483	2.0419	5.3623
MT	0.7201	0.8570	1.5251	0.7977	5.6248	0.7542	0.9330	0.5372	1.2784	4.7146

is 1/5/4, 0/10/0, 0/10/0, 2/4/4, 0/5/5, 2/6/2, 2/3/5 and 0/10/0, sequentially. By observing this set of data, we spot that the performance of the 2nd ranked MRFO and the 5th ranked TSO is comparable to GKS0 in half of the functions. However, TSO is not superior to GKS0 in any one function, while MRFO is superior to GKS0 in two functions. In addition, the 3rd ranked WSO and the 6th ranked AFSA also have two functions superior to GKS0, but AFSA is inferior to GKS0 in six functions. This indicates a polarization in AFSA's performance, resulting in a lower final ranking. Coincidentally, the 8th in parallel ranked ROA and SFO and the 7th ranked YSGA also have the same WRST results, but their performance is far inferior to the proposed GKS0. Therefore, according to statistical results in Table 7, we spot that the proposed GKS0's

performance is superior to other FCAs.

In order to compare the convergence performance of GKS0 and other FCAs from multiple perspectives, success rate (Sr) and mean evaluation (Me) are two commonly used indicators. Sr is the ratio of the number of successful runs of the algorithm to the total number of runs, and the condition for an algorithm to run successfully is that its obtains objective function value is better than or equal to a threshold we have given in advance. Me is the average number of evaluations corresponding to successful runs. When a run is marked as successful, the minimum number of evaluations required to find a solution with an objective function value that reaches the threshold will be recorded [101]. After determining the number of successful runs and the minimum number of

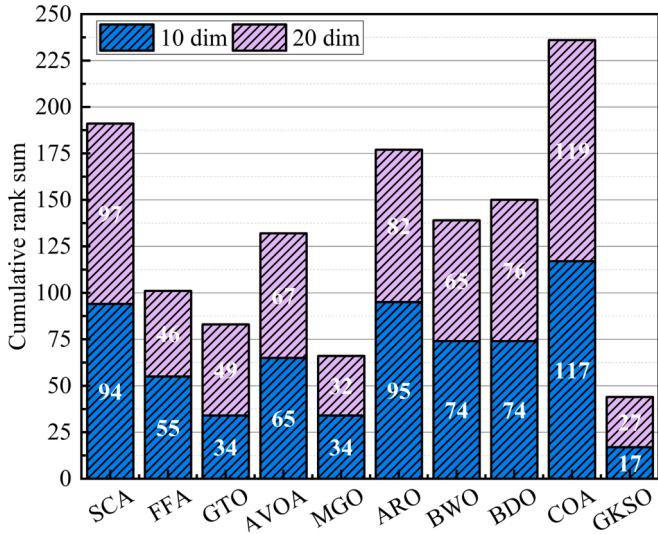


Fig. 13. Cumulative rank sum of GKS0 and OCAs on different dimensions.

evaluations required to reach the threshold in successful runs, S_r and Me indicators can be calculated. In the experiment, the threshold of F1 and F5 is set to 3, a threshold of F2 is set to 10, the threshold is set to 5 for F3 and F6, 25 is assigned as the threshold to F4, $1.000E + 03$ is assigned to F7 as the threshold, while 4 and 1.3 are assigned as the threshold to F8 and F9, respectively. Finally, a threshold of 21 is given for F10. Table 8 summarizes S_r and Me determined by the independent operation of FCAs for 20 times. It is easy to see that there is a significant difference in the success rate and average evaluation between GKS0 and other FCAs. Firstly, GKS0 can reach the determined threshold on all 10 benchmark problems, while in other FCAs, there is more or less a situation where $S_r = 0$ in one or more benchmark problems. This indicates that GKS0 is always able to break through the set threshold and continuously approach the optimal solution when facing various challenges, indicating that GKS0 has comprehensive capabilities. Therefore, S_r is another way to demonstrate the outstanding convergence performance of GKS0. In addition, when the S_r of GKS0 is the same as that of other FCAs, the minimum number of evaluations required for GKS0 to reach the threshold may be more, which may be due to the need for agents to continuously adjust their positions in four stages to obtain more valuable updates. Therefore, GKS0 needs more Me to ensure its strong competitiveness in FCAs.

An Algorithm's mean time (MT) is immediately interrelated to its complexity, which is a direct reflection of its complexity. The smaller MT value, the faster the algorithm's execution rate and the lower TC. The experimental outcomes are shown in Table 9, with minimum values highlighted in bold. Among them, WSO's MT is the shortest, only 0.1860 s, which also corresponds to its efficient update mechanism. The same high-speed MT is TSO, which takes 0.2126 s, because TSO has two efficient foraging behaviors: spiral and parabolic. While the longest MT belongs to AFSA, which takes 21.9434 s, and the running time on F6 in particular reaches 145.7794 s. In the end, the MT of GKS0 is 0.9041 s. Although GKS0 is not the fastest algorithm, it sacrifices a certain amount of MT to achieve more comprehensive advantages.

Fig. 10 shows the rank sum statistical results of GKS0 and other FCAs. We spot that GKS0 has the lowest rank sum value, indicating that GKS0 has the best overall performance. Fig. 11 shows convergence curves of GKS0 and other FCAs with 500 iterations. We find that GKS0 converges faster in the early phase of most functions compared to other FCAs, implying that GKS0 has a high search efficiency in the early phase, and each generation can effectively update the previous generation's solution. In addition, in the later iteration stage of F10, the convergence curve of GKS0 still shows a downward trend, which may be due to GKS0's self-protection mechanism, which helps the algorithm

jump out of local optimum. Fig. 12 shows the boxplot of GKS0 and other FCAs, and it can be seen that the box shape of GKS0 is relatively narrow and positioned lower in most functions. This indicates that the proposed GKS0 has good stability and high solving accuracy. In summary, GKS0 is superior to other FCAs on CEC2019.

5.4.2. Comparative evaluation of GKS0 and OCAs on CEC2022

Tables 10 to 12 show the comparison results between GKS0 and OCAs on 10-dimensional CEC2022. The experimental N is set to 50, and T is assigned as 500. Furthermore, the maximum fitness evaluation quantity depends on N and T . In order to investigate the impact of changes in the maximum fitness evaluation quantity on the performance of GKS0 solving, we also compare GKS0 and OCAs on 20-dimensional CEC2022, with experimental N set to 100 and T set to 1000. The detailed statistical results are shown in Tables 13 to 15. Below we will compare and analyze the performance of GKS0 around these two scales on CEC2022.

As shown in Tables 10 and 13, we spot that the mean ranking of GKS0 is 1.4167 and 2.2500, respectively, ranking 1st in comparison with OCAs at different scales. Followed closely by MGO, with mean ranks of 2.8333 and 2.6667 on two scales, respectively. It is worth mentioning that GTO, MGO, and GKS0 almost finds the TO on F1, with only a slight difference in Std, while GKS0 has the smallest Std, indicating that GKS0 is more stable. As shown in Table 10, GKS0 achieves 1st place in 9 of 12 benchmark problems, indicating that GKS0 is able to beat OCAs in the 10-dimensional problem by a wide margin. Correspondingly, as shown in Table 13, GKS0 achieves 1st place on four functions in the 20-dimensional CEC2022, which is also a good performance. It should be noted that although GKS0 performs well on most functions, it performs slightly worse on two scales of F3. Further consideration should be given to improving the accuracy of F3 solution in the future. Combining the cumulative rank sum of GKS0 and OCAs in different dimensions shown in Fig. 13, it can be seen that GKS0 has the lowest cumulative rank sum height compared to OCAs, and the resulting stacked bar chart has the smallest area, indicating that it can provide good output in both dimensions. When facing challenges from different dimensions, GKS0 always adheres to the principle of consistent ENE advantages, and outperforms OCAs in most functions. This result proves that with the increase of maximum fitness evaluation quantity and dimension, the quality of candidate solutions for GKS0 is not significantly affected, and it also proves its good robustness. Similarly, observing the changes in the rank sum of OCAs in different dimensionality in Fig. 13, FFA, MGO, ARO, and BWO can achieve better results as function optimization becomes more difficult.

Tables 10 and 13 provide error data for GKS0 and OCAs regarding RMSE and δ . GKS0 has smaller RMSE and δ on most benchmark problems, indicating that the optimal solution obtained by GKS0 is more stable. Meanwhile, Tables 10 and 13 also provide WRST results for GKS0 and OCAs at different scales. By comparing two sets of data, it can be seen that the only one that can compete with GKS0 is MGO. MGO ranks 2nd on both scales, behind GKS0. The number of functions that are superior/inferior/similar to GKS0 on different scales is 3/2/7 and 4/4/4 in that order. From this, we can find that MGO has as many as seven similar performances to GKS0 in 10 dimensions and outperforms GKS0 in 20 dimensions for four functions. This is due to the fact that MGO uses many finite vectors, which ameliorating the ability to explore all optimization spaces. In addition, the WRST test results of SCA, BDO and COA at both scales are the same, which are 0/12/0, 0/11/1, and 0/12/0 in order. Apparently, there is a significant gap in performance between these three methods and GKS0. Therefore, according to statistical results in Tables 10 and 13, we find that the proposed GKS0 performs better than OCAs.

Tables 11 and 14 provide the statistical results of S_r and Me for GKS0 and OCAs at two scales. In the 10-dimensional experiment, the threshold of F1 is set to $3.300E + 02$, a threshold of F2 is set to $4.200E + 02$, the threshold is set to $6.050E + 02$ for F3, $8.200E + 02$ is assigned as the

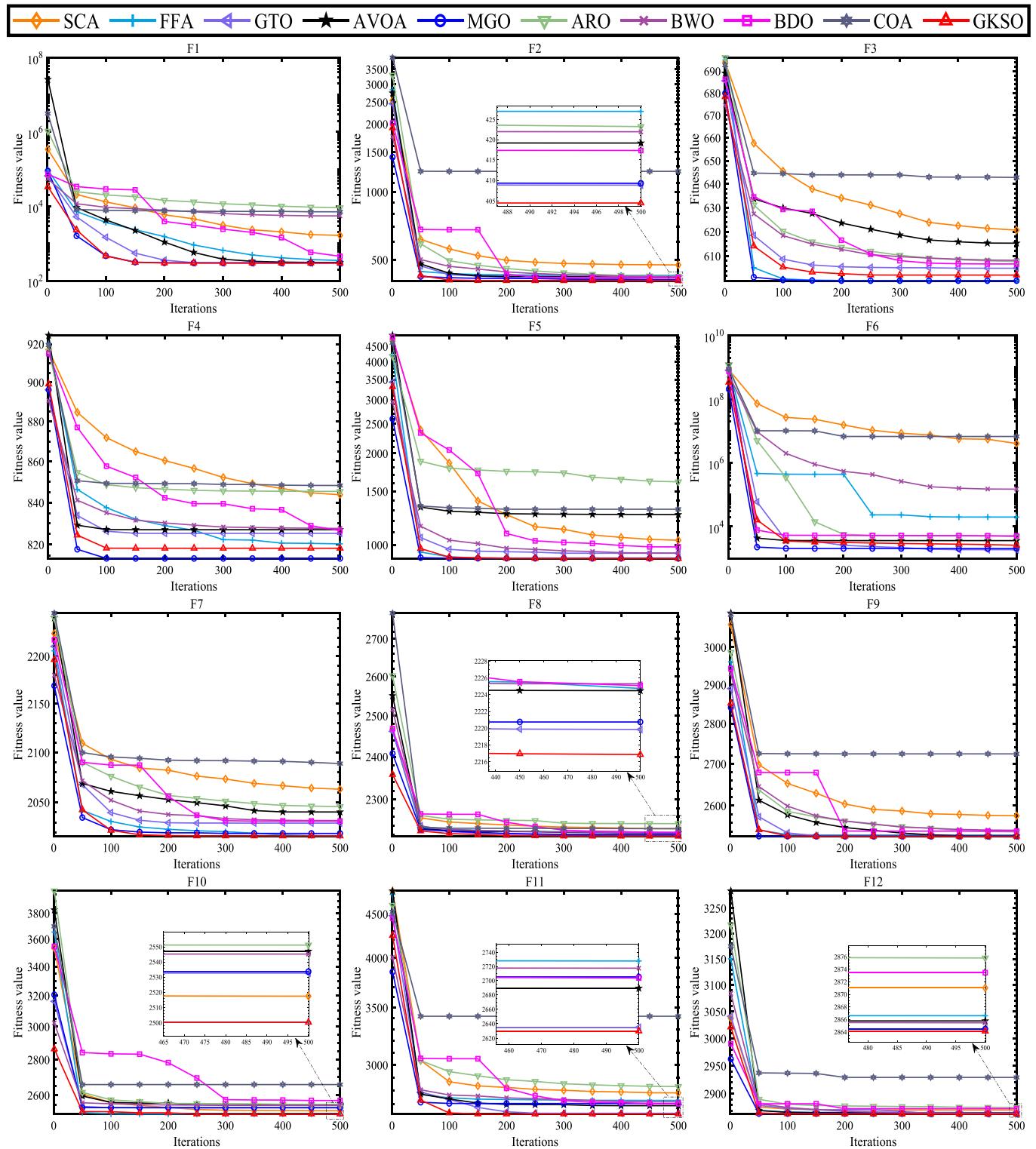


Fig. 14. Convergence graphs of GKS0 and OCAs for solving 10-dimensional CEC2022 ($N = 50$, $T = 500$).

threshold to F4, $9.050E + 02$ is assigned to F5 as the threshold, $2.500E + 03$ is assigned as the threshold to F6, the threshold of F7 is set to $2.030E + 03$, $2.220E + 03$ is assigned as the threshold to F8, the threshold of F9 and F10 is set to $2.530E + 03$, the threshold is set to $2.650E + 03$ for F11, Finally, a threshold of $2.865E + 03$ is given for F12. In the 20-dimensional experiment, the threshold of F1 is set to $3.200E + 02$, a threshold of F2 is set to $4.500E + 02$, the threshold is set to $6.100E + 02$ for F3, $8.700E + 02$ is assigned as the threshold to F4, $1.000E + 03$

is assigned to F5 as the threshold, $5.000E + 03$ is assigned as the threshold to F6, the threshold of F7 is set to $2.070E + 03$, $2.230E + 03$ is assigned as the threshold to F8, $2.481E + 03$ is assigned to F9 as the threshold, the threshold of F10 is set to $2.540E + 03$, the threshold is set to $2.901E + 03$ for F11, Finally, a threshold of $2.965E + 03$ is given for F12. It is not difficult to see that GKS0 achieves solutions that are equal to or better than the set threshold for all benchmark problems at both scales. However, SCA achieves good S_r only on the two scales of F10,

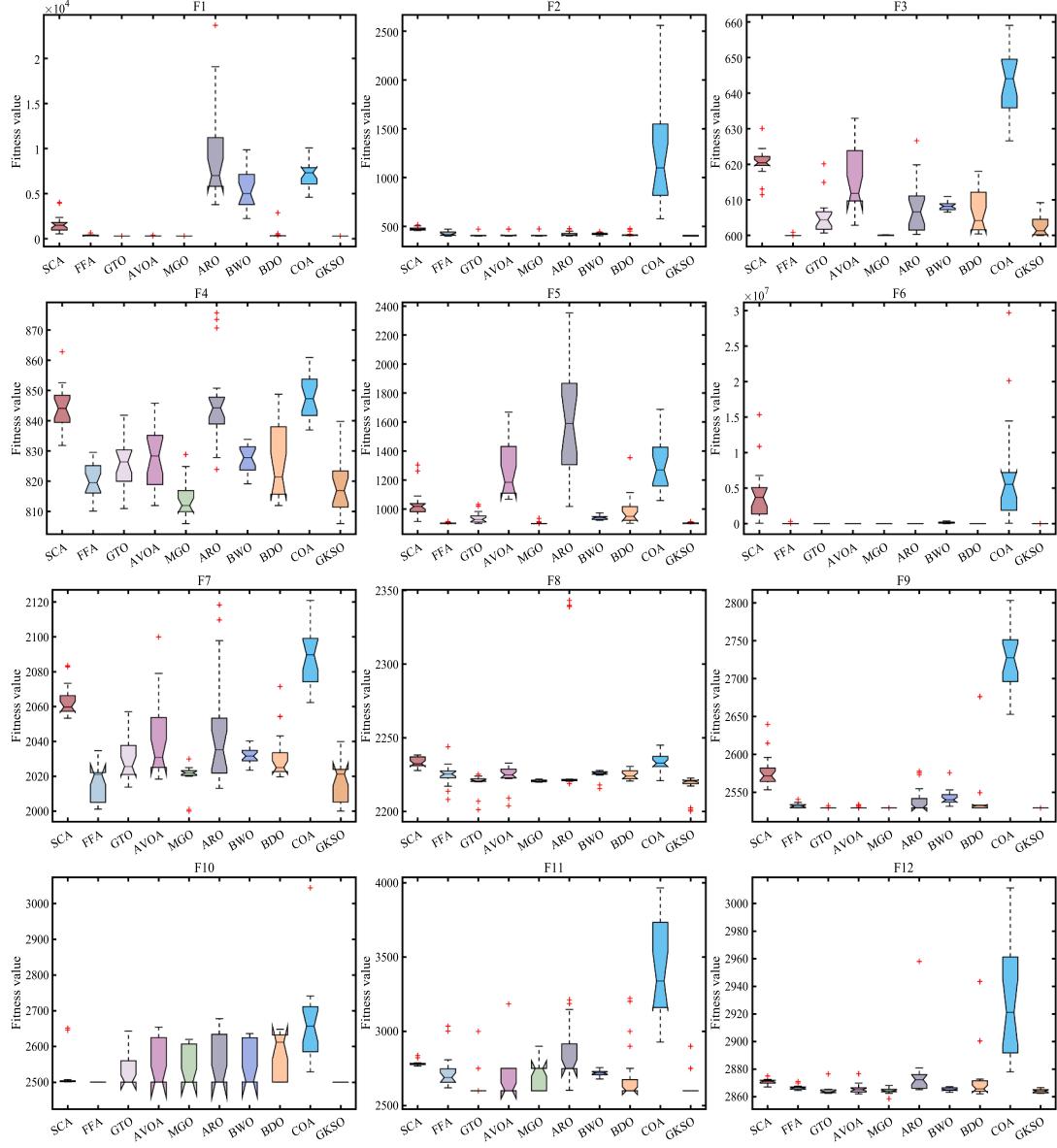


Fig. 15. Boxplots of GKS0 and OCAs for solving 10-dimensional CEC2022 ($N = 50$, $T = 500$).

while COA achieves 5 % Sr on the 10 dimensions of F10, which means that out of 20 independent runs, only one is successful, and the threshold set for any benchmark problem is not reached on the 20 dimensions. Among OCAs, MGO is more competitive, as it can achieve higher Sr while using fewer Me quantities on most functions. In addition, when GKS0 wins the same Sr as other OCAs, the required number of Me is still higher. Therefore, GKS0 needs to self replace by evaluating more candidate solutions to ensure its leading status in MAs.

Tables 12 and 15 reflect the MT's experimental results, where BDO has the shortest MT, which is only 0.0953 and 0.5372 s at two scales, respectively. This is because BDO has an efficient two-stage behavior. While the longest MT belongs to MGO, which takes 1.1581 and 5.6248 s at both scales, respectively. Although MGO has considerable advantages among many MAs, it consumes too much MT. In the end, The MT of GKS0 at two scales is 0.6201 and 4.7146 s, respectively. GKS0 is still not the fastest algorithm, but it sacrifices a certain amount of MT to achieve more significant advantages.

Figs. 14 and 16 show the comparison of convergence curves between GKS0 and OCAs at different scales. By observing these curves, we find that GKS0 has a very fast rate of convergence on most functions, and can

find the global optimum in the early and middle stages of the iteration. But its convergence accuracy on F3 is slightly inferior. Figs. 15 and 17 reflect the box graph of GKS0 and OCAs at two scales, it can be seen that the box shape of GKS0 is relatively narrow except for F3, F4, and F7, and positioned lower in most functions. This indicates that the proposed GKS0 has good stability and high solving accuracy. Overall, GKS0 is superior to OCAs on CEC2022.

6. GKS0 for solving CEC2020 constrained optimization test set

In the previous section, GKS0 shows certain advantages when compared to different types of algorithms. This section selects 50 famous engineering OPs in CEC2020 benchmark constrained optimization function [102] to evaluate the optimization ability of GKS0. These include industrial chemical processes problems (Prob. 1: #1-#7), process synthesis and design problems (Prob. 2: #8-#14), mechanical engineering problems (Prob. 3: #15-#33), power system problems (Prob. 4: #34-#44), and power electronics problems (Prob. 5: #45-#50). These real-world constrained OPs involve abundant decision variables, sophisticated non-linear constraints and objective functions, which are

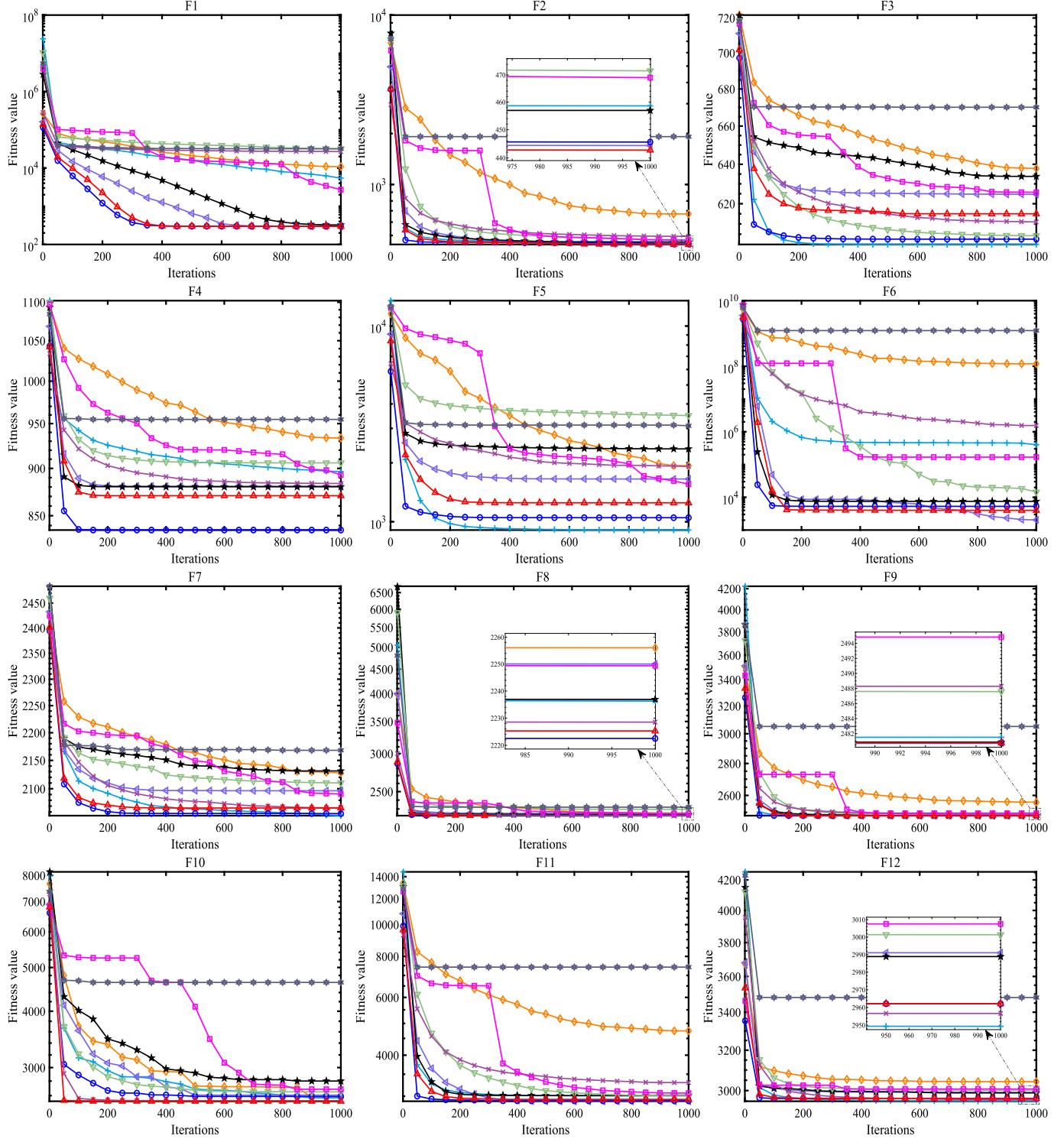


Fig. 16. Convergence graphs of GKS0 and OCAs for solving 20-dimensional CEC2022 ($N = 100$, $T = 1000$).

very challenging. Their specific mathematical models are detailed in [103]. Fig. 18 provides a diagrammatic sketch of some engineering problems. According to NFL theory, since different algorithms may behave widely divergent in different problems, we select several algorithms from two different types of comparison algorithms in the antecedent section to compare with the proposed GKS0, including ARO [57], BWO [98], SCA [60], BDO [99], ROA [90], TSO [84], and MRFO [92]. For fairness, the parameters of algorithms are set exactly the same as those in Section 5. Finally, the experimental results are evaluated and

analyzed.

Additionally, all engineering OPs achieve constraint processing (penalty term) by linearly weighting objective functions and nonlinear constraints [104]. In general, constrained minimization engineering OPs are defined as follows.

$$\text{Minimize : } f(\sigma), \sigma = [\sigma_1, \sigma_2, \dots, \sigma_m]. \quad (6.1)$$

Subject to:

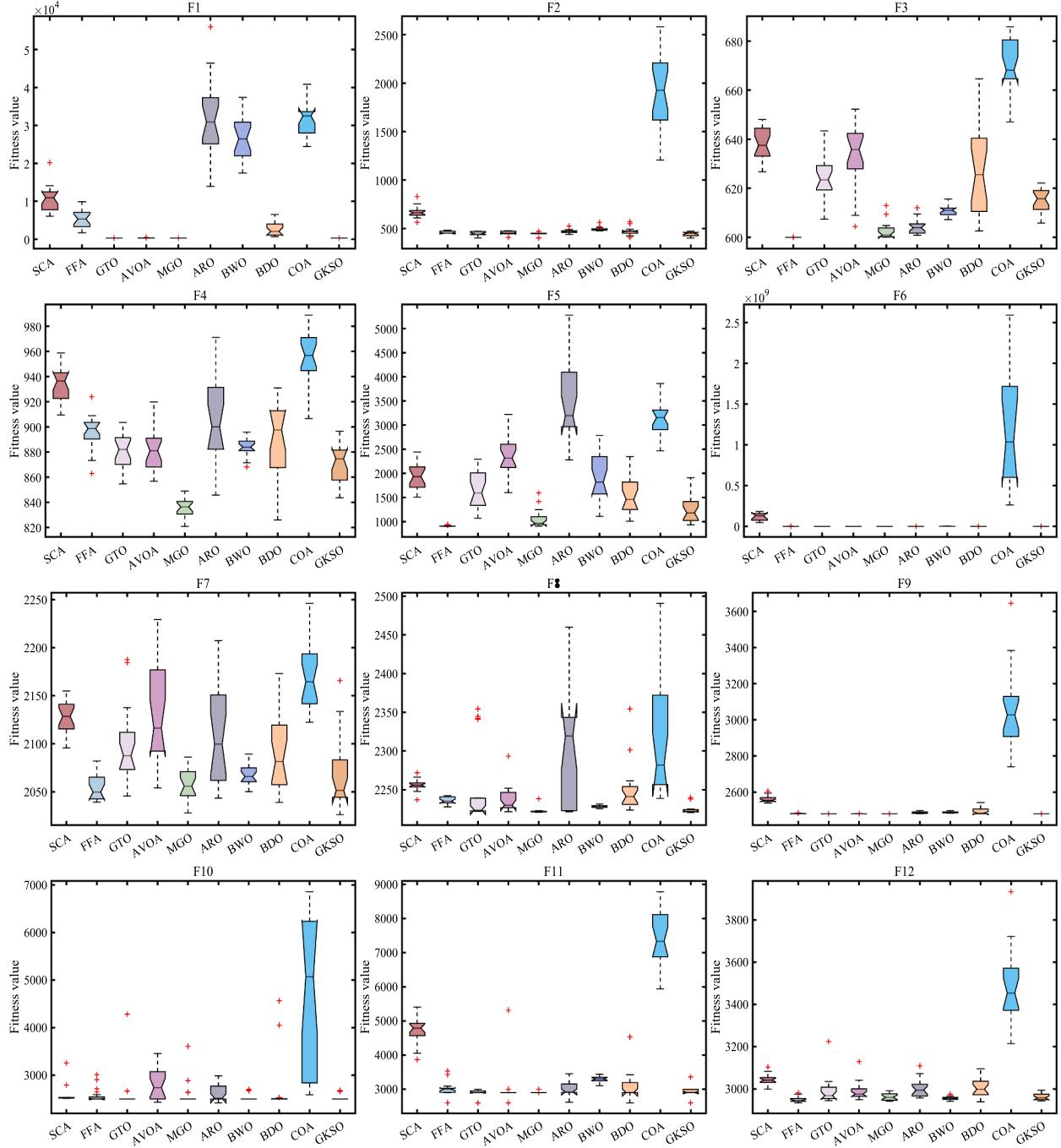


Fig. 17. Boxplots of GKS0 and OCAs for solving 20-dimensional CEC2022 ($N = 100$, $T = 1000$).

$$\begin{cases} \eta_i(\sigma) \leq 0, & i = 1, \dots, u, \\ \delta_j(\sigma) = 0, & j = 1, \dots, v. \end{cases} \quad (6.2)$$

where \setminus^* MERGEFORMAT represents the feasible solution of the problem. m indicates the number of variables. u and v are the number of multiple and equilibrium constraints, respectively. For the case with boundary constraints, each variable \setminus^* MERGEFORMAT has a ULBs requirement:

$$lb_c \leq \sigma_c \leq ub_c, \quad c = 1, 2, \dots, m. \quad (6.3)$$

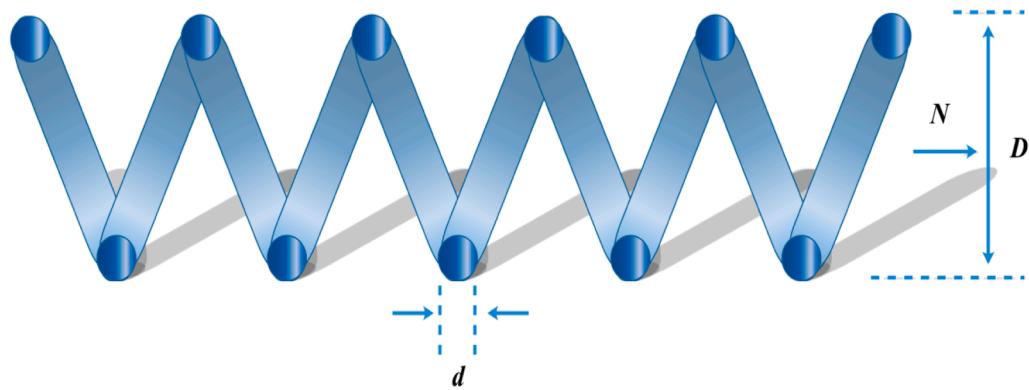
where lb_c and ub_c are the ULBs of the variable \setminus^* MERGEFORMAT.

Constrained minimization engineering OPs can be described by the linear weighting of objective functions and nonlinear constraints as follows.

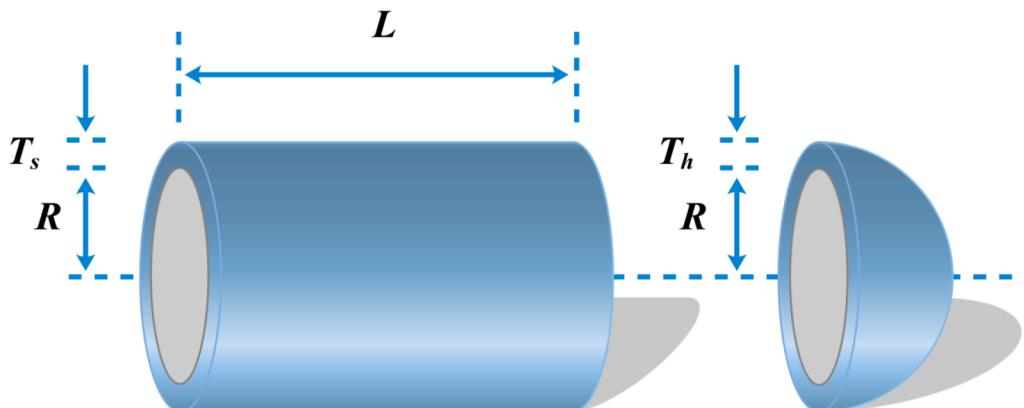
$$F(\sigma) = f(\sigma) + \mu_1 \sum_{i=1}^u \max\{\eta_i(\sigma), 0\} + \mu_2 \sum_{j=1}^v \max\{\delta_j(\sigma), 0\}. \quad (6.4)$$

where \setminus^* MERGEFORMAT and \setminus^* MERGEFORMAT are the weight of multiple and equilibrium constraints, respectively. This paper will provide very large weights to ensure that the solution is severely penalized if it exceeds any constraints. This measure will ensure that algorithms intentionally avoid illegal solutions during the optimization process.

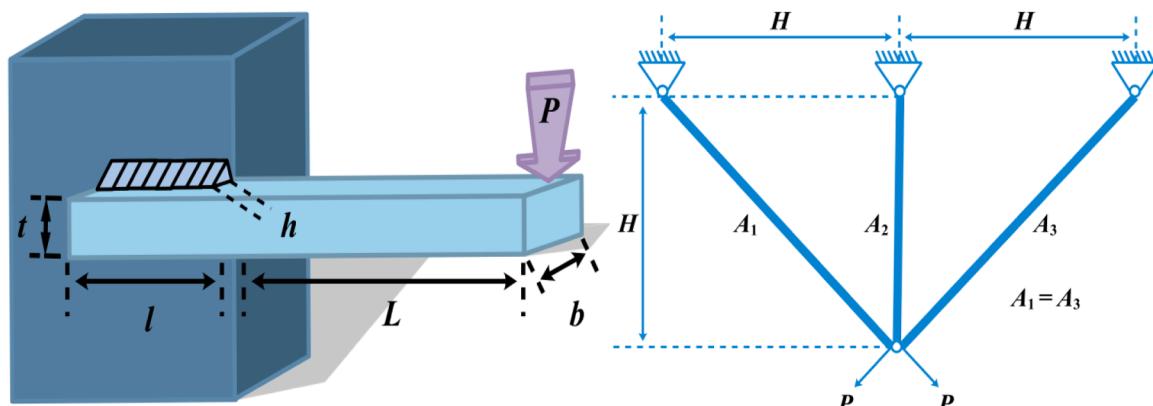
For the convenience of contrast and observation, **Table 16 to 20** list corresponding outcomes of each algorithm solving various types of OPs. **Table 21** summarizes the average ranking of **Table 16 to 20** to obtain the final ranking of each algorithm addressing 50 engineering problems. Among them, MR represents each algorithm's average ranking on



(a) #17: Tension/compression spring design (case 1).



(b) #18: Pressure vessel design problem.



(c) #19: Welded beam design problem.

(d) #20: Three-bar truss design problem.

Fig. 18. Partial diagrammatic sketch of 50 engineering benchmark suits.

different types OPs, and these data can be obtained from the penultimate row of [Tables 16 to 20](#), respectively.

By observing results in [Table 16 to 20](#), we spot that GKS0 obtains the optimal average fitness value among 19/50 engineering OPs. Firstly, in Prob. 1, GKS0 obtains the optimal value of 1/7 OPs (#7). Secondly, in Prob. 2, GKS0 successfully achieves the optimal values for 4/7 OPs (#8, #9, #12, and #13). In addition, GKS0 successfully wins the optimal values of 12/19 OPs (#17- #19, #21- #22, #24, and #28- #33) in Prob. 3, and 2/11 OPs (#42 and #43) in Prob. 4. Finally, among the 6 OPs in Prob. 5, GKS0 fails to obtain the optimal value. It can be seen that GKS0 wins first place on more than half of the OPs in Prob. 2 and Prob. 3,

indicating that the algorithm is competitive in solving these OPs, while there is still some room for improvement in Prob. 1, Prob. 4 and Prob. 5.

As shown in [Table 21](#), GKS0 ranks 1st with an average MR of 2.7754. Observing the bold data in the [Table 21](#), we spot that GKS0 achieves 1st place on Prob. 1-Prob. 4, indicating that GKS0 has universal effectiveness in OPs in different fields and its performance is superior to OCAs. The 2nd ranked algorithm is BDO, with an average MR of 3.4638. The algorithm has a MR of 1.5 in Prob. 5, while the subsequent GKS0 has a MR of only 3. Therefore, BDO wins the 1st place in the field of OPs with a significant advantage. MRFO and TSO rank 3rd and 4th respectively, they achieve 1st place in one optimization field. The results indicate that

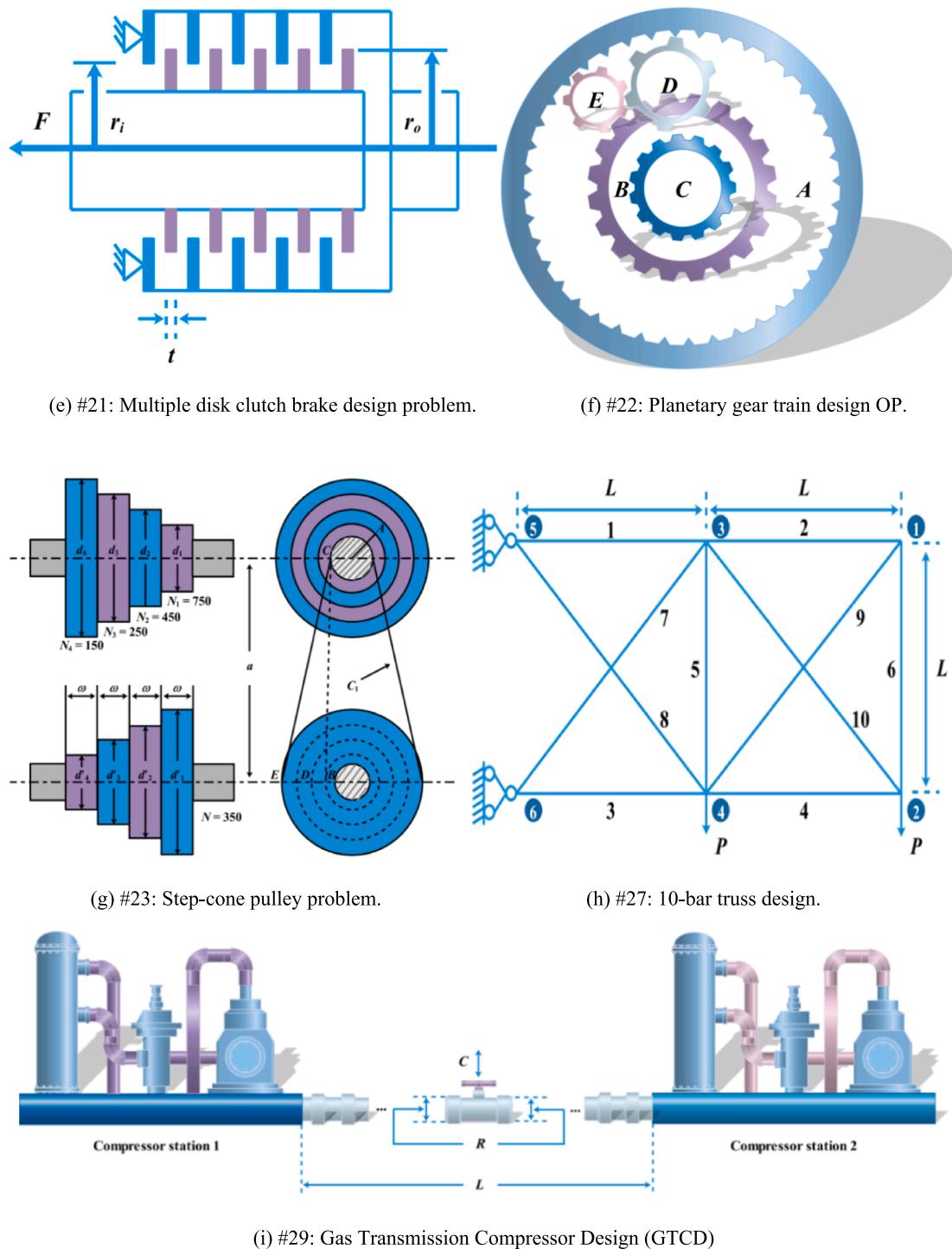


Fig. 18. (continued).

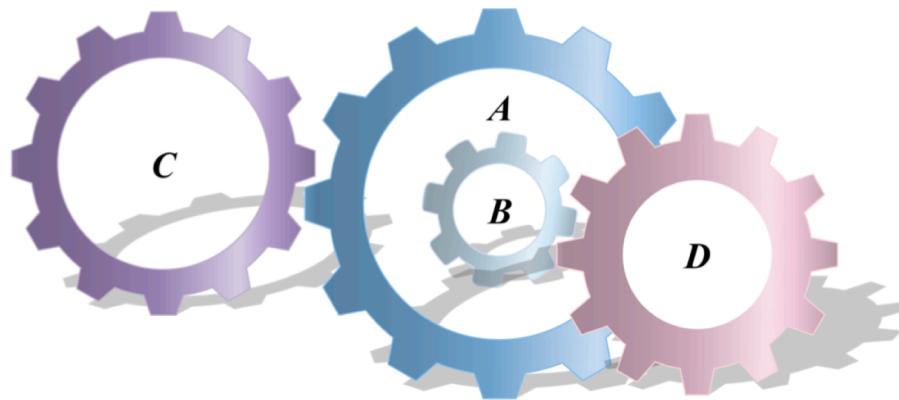
no MA has the top-notch performance for all OPs with various structures, and some MAs can address a few problems but do not hold good performance to address another problem.

Fig. 19 visually displays the stacked MR histograms of various algorithms in 5 different engineering OP categories. We find that the proposed GKSO has the lowest cumulative height and the smallest stacked MR histogram area in the five types of problems, which has a strong advantage in solving real-world OPs. However, overall optimization results of GKSO in solving Prob. 5 are not good enough, indicating that it still has some room for improvement. In addition, we find that the 5th ranked ARO obtained almost the same cross-sectional area for each

part when dealing with these five types of problems, indicating that ARO can provide stable output in the face of problems with different structures, and has good robustness.

7. Conclusion and future work

In this study, we propose a new population-based MA called the GKS Optimizer (GKSO). The basic concept of this algorithm is the predation and survival behavior of GKS. Inspired by the hunting, movement, foraging, and color changing escape behaviors of GKS, we establish a mathematical model for the entire optimization process of GKSO. The



(j) #31: Gear train design Problem.

Fig. 18. (continued).

Table 16

Comparison of results of solving Prob. 1 with different algorithms (#1-#7).

#	Index	Algorithms							
		ARO	BWO	SCA	BDO	ROA	TSO	MRFO	GKSO
1	MeanRank	4.1377E + 02	4.6300E + 02	1.4573E + 02	2.9945E + 02	4.5070E + 02	5.0828E + 02	4.2148E + 02	4.0249E + 02
	4	7	1	2	6	8	5	3	
2	MeanRank	6.8323E + 03	6.9954E + 03	1.3201E + 04	7.0270E + 03	7.3199E + 03	7.0490E + 03	7.0490E + 03	7.1719E + 03
	1	2	8	3	7	4	4	6	
3	MeanRank	-1.8800E + 04	-1.8486E + 04	-1.8522E + 04	-1.8495E + 04	-1.2877E + 04	-1.8600E + 04	-1.9550E + 04	-1.8435E + 04
	7	3	5	4	1	6	8	2	
4	MeanRank	-2.7470E-01	-3.1950E-01	0.0000E + 00	-7.0900E-02	-3.5290E-01	-1.0840E-01	-3.5250E-01	-2.7780E-01
	5	3	8	7	1	6	2	4	
5	MeanRank	-2.5342E + 03	-2.6809E + 03	-2.7711E + 03	-2.5750E + 03	-5.5909E + 02	-2.7661E + 03	-2.4432E + 03	-2.4365E + 03
	4	6	8	5	1	7	3	2	
6	MeanRank	1.0778E + 00	1.0062E + 00	9.9790E-01	1.8969E + 00	1.2588E + 00	1.2152E + 00	2.0313E + 00	1.4384E + 00
	6	7	8	1	4	5	2	3	
7	MeanRank	1.0692E + 00	1.0090E + 00	9.9780E-01	1.1806E + 00	1.2188E + 00	1.1544E + 00	1.2878E + 00	1.4615E + 00
	6	7	8	4	3	5	2	1	
Mean RankFinal		4.7143	5.0000	6.5714	3.7143	3.2857	5.8571	3.7143	3.0000
Ranking		5	6	8	3	2	7	3	1

Table 17

Comparison of results of solving Prob. 2 with different algorithms (#8-#14).

#	Index	Algorithms							
		ARO	BWO	SCA	BDO	ROA	TSO	MRFO	GKSO
8	Mean	2.0000E + 00							
	Rank	1	1	1	1	1	1	1	1
9	Mean	2.5581E + 00	2.5618E + 00	2.5625E + 00	2.5762E + 00	2.5582E + 00	2.5578E + 00	2.5578E + 00	2.5578E + 00
	Rank	4	6	7	8	5	1	1	1
10	Mean	1.1546E + 00	1.2412E + 00	1.1693E + 00	1.1459E + 00	1.1026E + 00	1.1459E + 00	1.0765E + 00	1.0972E + 00
	Rank	6	8	7	4	3	4	1	2
11	Mean	1.0989E + 02	1.0047E + 02	9.9229E + 01	1.0420E + 02	1.3012E + 02	1.0372E + 02	1.0169E + 02	1.0363E + 02
	Rank	7	2	1	6	8	5	3	4
12	Mean	3.7962E + 00	2.9332E + 00	8.4005E + 00	7.1653E + 00	5.9236E + 00	3.1422E + 00	2.9673E + 00	2.9248E + 00
	Rank	5	2	8	7	6	4	3	1
13	Mean	2.2303E + 04							
	Rank	1	1	1	1	1	1	1	1
14	Mean	2.0598E + 04	2.0598E + 04	2.0874E + 04	2.0598E + 04				
	Rank	2	2	1	2	2	2	2	2
Mean Rank		3.7143	3.1429	3.7143	4.1429	3.7143	2.5714	1.7143	1.7143
Final Ranking		5	4	5	8	5	3	1	1

hunting and movement behaviors are applied to ENE. The foraging behavior balances ENE, and the color changing escape behavior prevents GKSO from sinking into local optimum. GKSO, as an attempted expansion of the existing system, not only overcomes the weaknesses of most MAs in latency and low efficiency, but also achieves a balance between population diversity and convergence.

This study provides in-depth analysis of the proposed GKSO from both qualitative and quantitative perspectives. Qualitative analysis verifies that GKSO has good ENE capability. Simultaneously, GKSO is quantitatively analyzed with eight existing fish optimization algorithms and nine other well-known MAs on CEC2019 and CEC2022, respectively. The applicability and robustness of GKSO are verified by

Table 18

Comparison of results of solving Prob. 3 with different algorithms (#15-#33).

#	Index	Algorithms							
		ARO	BWO	SCA	BDO	ROA	TSO	MRFO	GKSO
15	MeanRank	2.7152E + 03	2.7335E + 03	2.7771E + 03	2.7275E + 03	2.7582E + 03	2.7130E + 03	2.7130E + 03	2.7130E + 03
	5	3	1	4	2	6	6	6	6
16	MeanRank	6.2210E-01	4.9500E-02	1.1560E-01	6.7270E-01	1.2845E + 02	6.2200E-02	4.7500E-02	2.0060E-01
	6	2	4	7	8	3	1	5	
17	MeanRank	1.3000E-02	1.3100E-02	1.2900E-02	1.4600E-02	1.3400E-02	1.2700E-02	1.2700E-02	1.2700E-02
	5	6	4	8	7	1	1	1	1
18	MeanRank	4.7948E + 02	4.7464E + 02	4.7441E + 02	4.8097E + 02	4.7780E + 02	4.7396E + 02	4.7396E + 02	4.7396E + 02
	7	5	4	8	6	1	1	1	1
19	MeanRank	1.8996E + 00	1.8644E + 00	1.7664E + 00	1.9114E + 00	2.0111E + 00	1.6705E + 00	1.6702E + 00	1.6702E + 00
	6	5	4	7	8	3	1	1	
20	MeanRank	9.8968E + 01	9.8989E + 01	9.8900E + 01	9.8970E + 01	9.9485E + 01	9.8968E + 01	9.8968E + 01	9.8968E + 01
	4	2	8	3	1	4	4	4	4
21	MeanRank	2.3520E-01	2.3530E-01	2.3580E-01	2.3520E-01	2.6620E-01	2.3520E-01	2.3520E-01	2.3520E-01
	1	6	7	1	8	1	1	1	1
22	MeanRank	5.7340E-01	5.7160E-01	6.2400E-01	5.4290E-01	7.3970E-01	5.3340E-01	5.3010E-01	5.2960E-01
	6	5	7	4	8	3	2	1	
23	MeanRank	8.7220E + 00	8.6990E + 00	9.3155E + 00	9.0258E + 00	1.1791E + 01	8.5682E + 00	8.5633E + 00	8.5633E + 00
	4	5	2	3	1	6	7	7	
24	MeanRank	5.6527E + 00	4.3007E + 00	4.2893E + 00	3.6232E + 00	2.1332E + 01	2.9630E + 00	2.9301E + 00	2.7038E + 00
	7	6	5	4	8	3	2	1	
25	MeanRank	8.3559E + 02	1.2090E + 03	6.7095E + 02	2.2578E + 02	5.1565E + 02	1.9514E + 02	2.3366E + 02	2.7778E + 02
	2	1	3	7	4	8	6	5	
26	MeanRank	6.7552E + 01	1.7008E + 02	5.4918E + 01	2.8547E + 01	3.5572E + 01	4.8040E + 01	4.5314E + 01	5.0964E + 01
	7	8	6	1	2	4	3	5	
27	MeanRank	4.5758E + 01	4.5569E + 01	4.7614E + 01	4.5722E + 01	4.6232E + 01	4.5759E + 01	4.5759E + 01	4.5759E + 01
	3	1	8	2	7	4	4	4	
28	MeanRank	5.5994E + 03	5.5994E + 03	5.5994E + 03	5.6082E + 03	5.6680E + 03	5.5994E + 03	5.5994E + 03	5.5994E + 03
	1	1	1	7	8	1	1	1	
29	MeanRank	9.7607E + 05	9.7607E + 05	9.7607E + 05	9.7607E + 05	1.0708E + 06	9.7607E + 05	9.7607E + 05	9.7607E + 05
	1	1	1	1	8	1	1	1	
30	MeanRank	2.7992E + 00	3.0051E + 00	2.7613E + 00	2.9340E + 00	2.9124E + 00	2.7441E + 00	2.6647E + 00	2.6606E + 00
	5	8	4	7	6	3	2	1	
31	MeanRank	2.2662E-16	3.1986E-12	1.4565E-11	0.0000E + 00	9.0883E-13	3.1679E-17	0.0000E + 00	0.0000E + 00
	5	7	8	1	6	4	1	1	
32	MeanRank	-3.2217E + 04							
	1	1	1	1	1	1	1	1	
33	MeanRank	2.6393E + 00	2.6393E + 00	3.4476E + 00	2.6393E + 00	2.6393E + 00	2.6393E + 00	2.6400E + 00	2.6393E + 00
	1	1	8	1	1	1	7	1	
Mean RankFinal		4.0526	3.8947	4.5263	4.0526	5.2632	3.0526	2.7368	2.5263
Ranking		5	4	7	5	8	3	2	1

Table 19

Comparison of results of solving Prob. 4 with different algorithms (#34-#44).

#	Index	Algorithms							
		ARO	BWO	SCA	BDO	ROA	TSO	MRFO	GKSO
34	MeanRank	2.8141E + 01	1.6822E + 01	2.5415E + 01	1.9473E + 01	1.7523E + 01	2.1069E + 01	2.0869E + 01	1.7436E + 01
	8	1	7	4	3	6	5	2	
35	MeanRank	1.4207E + 01	1.0161E + 02	4.1755E + 02	5.2039E + 02	7.4466E + 00	2.1876E + 01	1.2965E + 03	4.0918E + 02
	2	4	6	7	1	3	8	5	
36	MeanRank	3.4948E + 02	5.5288E + 02	3.0673E + 02	4.1748E + 02	4.7914E + 02	2.1275E + 02	1.0438E + 03	7.0055E + 02
	3	6	2	4	5	1	8	7	
37	MeanRank	3.9360E + 00	6.2953E + 00	4.5177E + 00	6.0447E + 00	6.2341E + 00	6.0311E + 00	6.2953E + 00	6.1485E + 00
	1	7	2	4	6	3	7	5	
38	MeanRank	7.0992E + 00	1.3383E + 01	1.0390E + 01	1.2516E + 01	1.3371E + 01	1.2196E + 01	1.3608E + 01	1.3326E + 01
	1	7	2	4	6	3	8	5	
39	MeanRank	8.7455E + 00	1.6388E + 01	1.1565E + 01	1.6618E + 01	1.5120E + 01	1.5323E + 01	1.6868E + 01	1.6247E + 01
	1	6	2	7	3	4	8	5	
40	MeanRank	1.1706E + 04	1.6838E + 04	3.1293E + 04	8.4653E + 02	8.9534E + 03	4.6357E + 02	2.0929E + 02	6.5028E + 02
	6	7	8	4	5	2	1	3	
41	MeanRank	4.8045E + 03	3.7413E + 02	4.2239E + 03	5.7416E + 01	6.0073E + 02	8.4482E + 01	7.3209E + 01	8.1144E + 01
	8	5	7	1	6	4	2	3	
42	MeanRank	-7.4972E + 01	-2.0030E-01	-5.6660E-01	-2.1130E-01	-2.2170E-01	-2.4690E-01	-8.3640E-01	0.0000E + 00
	8	2	6	3	4	5	7	1	
43	MeanRank	-1.9115E + 01	-8.5400E-02	-6.8760E-01	-1.0450E-01	-1.1060E-01	-1.2420E-01	-1.8050E-01	0.0000E + 00
	8	2	7	3	4	5	6	1	
44	MeanRank	-6.0312E + 03	-5.2220E + 03	-5.2575E + 03	-5.9680E + 03	-5.3054E + 03	-5.7108E + 03	-5.6604E + 03	-5.7556E + 03
	1	8	7	2	6	4	5	3	
Mean Rank		4.2727	5.0000	5.0909	3.9091	4.4545	3.6364	5.9091	3.6364
Final Ranking		4	6	7	3	5	1	8	1

Table 20

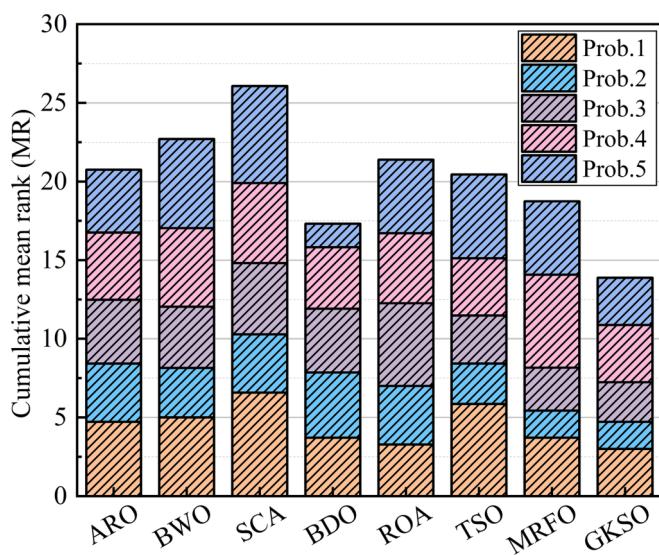
Comparison of results of solving Prob. 5 with different algorithms (#45–#50).

#	Index	Algorithms							
		ARO	BWO	SCA	BDO	ROA	TSO	MRFO	GKSO
45	MeanRank	7.4410E-01	9.3350E-01	9.1210E-01	8.8990E-01	9.3680E-01	9.1200E-01	9.1100E-01	7.7490E-01
	1	7	6	3	8	5	4	2	
46	MeanRank	5.2810E-01	2.7800E-01	4.3000E-01	2.4940E-01	2.7710E-01	2.7710E-01	3.4430E-01	2.7710E-01
	8	5	7	1	2	2	6	2	
47	MeanRank	3.9320E-01	7.6380E-01	4.6430E-01	2.6760E-01	7.4240E-01	8.8670E-01	1.1850E-01	5.2620E-01
	3	7	4	2	6	8	1	5	
48	MeanRank	5.0930E-01	9.2180E-01	7.6990E-01	2.3950E-01	7.7870E-01	9.3680E-01	7.5000E-01	5.5620E-01
	2	7	5	1	6	8	4	3	
49	MeanRank	4.0910E-01	3.6060E-01	4.0750E-01	1.8000E-01	3.2300E-01	3.6060E-01	4.0000E-01	3.6060E-01
	8	3	7	1	2	3	6	3	
50	MeanRank	2.8100E-01	3.7000E-01	5.4170E-01	1.3930E-01	3.6400E-01	3.7190E-01	5.0000E-01	3.5700E-01
	2	5	8	1	4	6	7	3	
Mean RankFinal Ranking		4.0000	5.6667	6.1667	1.5000	4.6667	5.3333	4.6667	3.0000
		3	7	8	1	4	6	4	2

Table 21

Contrast results of different algorithms for solving various types engineering OPs (Prob. 1–Prob. 5).

Prob.	Index	Algorithms							
		ARO	BWO	SCA	BDO	ROA	TSO	MRFO	GKSO
1	MR	4.7143	5.0000	6.5714	3.7143	3.2857	5.8571	3.7143	3.0000
2	MR	3.7143	3.1429	3.7143	4.1429	3.7143	2.5714	1.7143	1.7143
3	MR	4.0526	3.8947	4.5263	4.0526	5.2632	3.0526	2.7368	2.5263
4	MR	4.2727	5.0000	5.0909	3.9091	4.4545	3.6364	5.9091	3.6364
5	MR	4.0000	5.6667	6.1667	1.5000	4.6667	5.3333	4.6667	3.0000
Mean MRFinal Ranking		4.1508	4.5409	5.2139	3.4638	4.2769	4.0902	3.7482	2.7754
		5	7	8	2	6	4	3	1

**Fig. 19.** Cumulative average ranking of different algorithms in 5 different engineering OP categories.

changing the maximum fitness evaluation quantity on different dimensions of CEC2022. Statistical results indicate that GKS0 performs better in the competition between two different types of algorithms, and the algorithm can return better (near optimal) solutions. However, GKS0 has certain limitations in numerical optimization. Although it only has one control parameter m , it has a higher computational cost and more number of Me . Moreover, the proposed GKS0 evaluates the performance difference with the other seven optimizers through five different OPs in CEC2020 benchmark-constrained optimization functions, including 50 engineering case suites. The experimental results show that in most real-world OPs, GKS0 has a good ability to handle

various constraints, ranking 1st in terms of comprehensive ability among all algorithms. However, GKS0 also has certain limitations in engineering optimization. When faced with large-scale OPs with over 100 decision variables, the execution power of this algorithm is not outstanding enough. This requires the introduction of special mechanisms at the algorithmic level to handle large-scale constrained OPs. But in general, GKS0 has certain competitiveness and even advantages in dealing with different issues.

We have demonstrated the proposed GKS0's effectiveness in handling single objective continuous OPs, especially in the innovative work of comparing it with fish optimizers, where its performance outperforms many well-known fish algorithms such as MBF, WSO, and MRFO. However, GKS0 still has ideas that can be expanded. In future work, researchers can make more attempts. Firstly, in GKS0, the self-protection mechanism is matched to the fourth stage. However, during GKS hunting, this survival behavior is expected to occur at any stage. Therefore, how this situation can be more appropriately integrated in GKS will be an interesting topic, and we will further improve the self-protection mechanism in future research. After that, a multi-objective variant and binary version of GKS0 can be developed and executed to address high-dimensional multi-objective or large-scale real-world OPs. It is worth noting that for the control parameter $m = 1.5$, it needs to be dynamically adjusted according to the changes in the problem. Therefore, designing a flexible and adaptable version of m is a positive-going measure. Moreover, other initialization schemes [105], such as Opposition-based Learning mechanism and chaotic mapping, can also be introduced into GKS0 to better improve optimization performance. In addition, GKS0 can be modified and hybridized with other MAs to generate a new fused variant algorithm. This is a promising job worth considering for researchers. Finally, GKS0 can also be applied to various OPs in science and the real world, such as feature selection [106], hyper-parameter optimization [107], image segmentation [108], UAV path planning [109–111], curve and surface degree reduction [112], shape optimization [113] and complex engineering applications [114,115], etc.

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

Data will be made available on request.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 52375264).

References

- [1] A.E. Ezugwu, Advanced discrete firefly algorithm with adaptive mutation-based neighborhood search for scheduling unrelated parallel machines with sequence-dependent setup times, *Int. J. Intell. Syst.* 37 (8) (2021) 4612–4653.
- [2] A.E. Ezugwu, A.K. Shukla, R. Nath, et al., Metaheuristics: a comprehensive overview and classification along with bibliometric analysis, *Artif. Intell. Rev.* 54 (6) (2021) 4237–4316.
- [3] L. Abualigah, Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications, *Neural Comput. & Applic.* 33 (7) (2021) 2949–2972.
- [4] A. Mahdavi-Meymand, M. Zounemat-Kermani, Homonuclear Molecules Optimization meta-heuristic algorithm, *Knowl.-Based Syst.* 258 (2022), 110032.
- [5] A.E. Ezugwu, J.O. Agushaka, L. Abualigah, et al., Prairie Dog Optimization Algorithm, *Neural Comput. & Applic.* 34 (22) (2022) 20017–20065.
- [6] J.O. Agushaka, A.E. Ezugwu, Influence of probability distribution initialization methods on the Performance of Advanced Arithmetic Optimization Algorithm with Application to Unrelated Parallel Machine Scheduling Problem, *Concurrency Comput.-Pract. Exper.* 34 (22) (2022) e6871.
- [7] J.H. Holland, Genetic Algorithms, *Sci. Am.* 267 (1) (1992) 66–72.
- [8] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of ICNN95-International Conference on Neural Networks.* 4 (1995) 1942–1948.
- [9] X.L. Li, F. Lu, G.H. Tian, J.X. Qian, Applications of artificial fish school algorithm in combinatorial optimization problems, *J. Shandong Univ. (Eng. Sci.)* 34 (5) (2004) 64–67.
- [10] Y. Yuan, Q. Shen, S. Wang, Coronavirus Mask Protection Algorithm: A New Bio-inspired Optimization Algorithm and Its Applications, *J. Bionic Eng.* 20 (4) (2023) 1747–1765.
- [11] M.A. Al-Betar, Z.A.A. Alyasser, M.A. Awadallah, I. Abu Doush, Coronavirus herd immunity optimizer (CHIO), *Neural Comput. & Applic.* 33 (10) (2021) 5011–5042.
- [12] S. Aslan, S. Demirci, Immune plasma algorithm: A novel meta-heuristic for optimization problems, *IEEE Access* 8 (2020) 220227–220245.
- [13] V. Goodarzimehr, S. Talatahari, S. Shojaee, S. Hamzehei-Javaran, Special Relativity Search for applied mechanics and engineering, *Comput. Methods Appl. Mech. Eng.* 403 (2023), 115734.
- [14] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.* 1 (1) (1997) 67–82.
- [15] A.H. Rabie, N.A. Mansour, A.I. Saleh, Leopard seal optimization (LSO): A natural inspired meta-heuristic algorithm, *Commun. Nonlinear Sci. Numer. Simul.* 125 (2023), 107338.
- [16] K. Zervoudakis, S. Tsafarakis, A mayfly optimization algorithm, *Comput. Indus. Eng.* 145 (2020), 106559.
- [17] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: A novel optimization algorithm, *Knowl.-Based Syst.* 191 (2020), 105190.
- [18] J. Xue, B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, *Syst. Sci. Control Eng. Open Access J.* 8 (1) (2020) 22–34.
- [19] A. Faramarzi, M. Heidarinejad, S. Mirjalili, et al., Marine Predators Algorithm: A Nature-inspired Metaheuristic, *Expert Syst. Appl.* 152 (2020), 113377.
- [20] I. Ahmadianfar, O. Bozorg-Haddad, X.F. Chu, Gradient-based optimizer: A new metaheuristic optimization algorithm, *Inf. Sci.* 540 (2020) 131–159.
- [21] M. Braik, A. Sheta, H. Al-Hiary, A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm, *Neural Comput. & Applic.* 33 (7) (2021) 2515–2547.
- [22] G. Dhiman, M. Garg, A. Nagar, et al., A novel algorithm for global optimization: Rat Swarm Optimizer, *J. Ambient Intell. Hum. Comput.* 12 (8) (2021) 8457–8482.
- [23] Y.Y. Zhang, Z.G. Jin, S. Mirjalili, Generalized normal distribution optimization and its applications in parameter extraction of photovoltaic models, *Energ. Conver. Manage.* 224 (2020), 113301.
- [24] E. Bogar, S. Beyhan, Adolescent Identity Search Algorithm (AISA): A novel metaheuristic approach for solving optimization problems, *Appl. Soft Comput.* 95 (2020), 106503.
- [25] H. Ghasemian, F. Ghasemian, H. Vahdat-Nejad, Human urbanization algorithm: A novel metaheuristic approach, *Math. Comput. Simul.* 178 (2020) 1–15.
- [26] L. Abualigah, D. Yousri, M. Abd Elaziz, A.A. Ewees, M.A.A. Al-qaness, A. H. Gandomi, Aquila Optimizer: A novel meta-heuristic optimization algorithm, *Comput. Ind. Eng.* 157 (2021), 107250.
- [27] M.S. Braik, Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems, *Expert Syst. Appl.* 174 (2021), 114685.
- [28] Y.T. Yang, H.L. Chen, A.A. Heidari, A.H. Gandomi, Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts, *Expert Syst. Appl.* 177 (2021), 114864.
- [29] M. Jafari, E. Salajegheh, J. Salajegheh, Elephant clan optimization: A nature-inspired metaheuristic algorithm for the optimal design of structures, *Appl. Soft Comput.* 113 (2021), 107892.
- [30] A.T. Salawudeen, M.B. Mu'azu, Y.A. Sha'aban, A.E. Adedokun, A Novel Smell Agent Optimization (SAO): An extensive CEC study and engineering application, *Knowl.-Based Syst.* 232 (2021), 107486.
- [31] M. Ghasemi, A. Rahimnejad, R. Hemmati, E. Akbari, S. Andrew Gadsden, Wild Geese Algorithm: A novel algorithm for large scale optimization based on the natural life and death of wild geese, *Array.* 11 (2021), 100074.
- [32] H. Karami, M.V. Anaraki, S. Farzin, S. Mirjalili, Flow Direction Algorithm (FDA): A Novel Optimization Approach for Solving Optimization Problems, *Comput. Ind. Eng.* 156 (2021), 107224.
- [33] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Eng.* 376 (2021), 113609.
- [34] F.A. Hashim, E.H. Houssein, K. Hussain, M.S. Mabrouk, W. Al-Atabany, Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems, *Math. Comput. Simul.* 192 (2022) 84–110.
- [35] M.M. Noel, V. Muthiah-Nakarajan, G.B. Amali, A.S. Trivedi, A new biologically inspired global optimization algorithm based on firebug reproductive swarming behaviour, *Expert Syst. Appl.* 183 (2021), 115408.
- [36] D.B. Chen, Y.Y. Ge, Y.J. Wan, Y. Deng, Y. Chen, F. Zou, Poplar optimization algorithm: A new meta-heuristic optimization technique for numerical optimization and image segmentation, *Expert Syst. Appl.* 200 (2022), 117118.
- [37] N. Eslami, S. Yazdani, M. Mirzaei, E. Hadavandi, Aphid-Ant Mutualism: A novel nature-inspired metaheuristic algorithm for solving optimization problems, *Math. Comput. Simul.* 201 (2022) 362–395.
- [38] Y.X. Jiang, Q. Wu, S.K. Zhu, L.K. Zhang, Orca predation algorithm: A novel bio-inspired algorithm for global optimization problems, *Expert Syst. Appl.* 188 (2022), 116026.
- [39] M. Azizi, S. Talatahari, A.H. Gandomi, Fire Hawk Optimizer: a novel metaheuristic algorithm, *Artif. Intell. Rev.* 56 (1) (2023) 287–363.
- [40] M. Alimoradi, H. Azgomi, A. Asghari, Trees Social Relations Optimization Algorithm: A new Swarm-Based metaheuristic technique to solve continuous and discrete optimization problems, *Math. Comput. Simul.* 194 (2022) 629–664.
- [41] T.M. Shami, D. Grace, A. Burr, et al., Single candidate optimizer: a novel optimization algorithm, *Evol. Intel.* (2022).
- [42] F.A. Hashim, A.G. Hussien, Snake Optimizer: A novel meta-heuristic optimization algorithm, *Knowledge-Based Syst.* 242 (2022), 108320.
- [43] I. Ahmadianfar, A.A. Heidari, S. Noshadian, H.L. Chen, A.H. Gandomi, INFO: An efficient optimization algorithm based on weighted mean of vectors, *Expert Syst. Appl.* 195 (2022), 116516.
- [44] P. Chen, S.H. Zhou, Q. Zhang, N. Kasabov, A meta-inspired termite queen algorithm for global optimization and engineering design problems, *Eng. Appl. Artif. Intel.* 111 (2022), 104805.
- [45] M. Abdel-Basset, R. Mohamed, M. Jameel, M. Abouhawwash, Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems, *Knowl.-Based Syst.* 262 (2023), 110248.
- [46] M. Abdel-Basset, D. El-Shahat, M. Jameel, M. Abouhawwash, Young's double-slit experiment optimizer: A novel metaheuristic optimization algorithm for global and constraint optimization problems, *Comput. Methods Appl. Mech. Eng.* 403 (2023), 115652.
- [47] M. Dehghani, P. Trojovský, Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems, *Frontiers. Mech. Eng.* 8 (2) (2022).
- [48] M. Abdel-Basset, D. El-Shahat, M. Jameel, et al., Exponential distribution optimizer (EDO): a novel math-inspired algorithm for global optimization and engineering problems, *Artif. Intell. Rev.* (2023).
- [49] M. Abdel-Basset, R. Mohamed, M. Jameel, et al., Spider wasp optimizer: a novel meta-heuristic optimization algorithm, *Artif. Intell. Rev.* (2023).
- [50] H.A. Shehadeh, Chernobyl disaster optimizer (CDO): a novel meta-heuristic method for global optimization, *Neural Comput. Appl.* 35 (15) (2023) 10733–10749.
- [51] M. Azizi, U. Aickelin, H.A. Khorshidi, et al., Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization, *Sci. Rep.* 13 (1) (2023).
- [52] M. Abdel-Basset, R. Mohamed, M. Zidan, M. Jameel, M. Abouhawwash, Mantis Search Algorithm: A novel bio-inspired algorithm for global optimization and engineering design problems, *Comput. Methods Appl. Mech. Eng.* 415 (2023), 116200.
- [53] J. Kocoń, I. Cichecki, O. Kaszyca, M. Kochanek, D. Szydło, J. Baran, J. Bielaniewicz, M. Gruza, A. Janz, K. Kanclerz, A. Kocoń, B. Koptyra, W. Mieleszczko-Kowszewicz, P. Miłkowski, M. Oleksy, M. Piasecki, Ł. Radliński, K. Wojtasik, S. Woźniak, P. Kazienko, ChatGPT: Jack of all trades, master of none, *Information Fusion.* 99 (2023), 101861.
- [54] J.O. Agushaka, A.E. Ezugwu, L. Abualigah, Dwarf Mongoose Optimization Algorithm, *Comput. Methods Appl. Mech. Eng.* 391 (2022), 114570.

- [55] L. Abualigah, M. Abd Elaziz, P. Sumari, Z.W. Geem, A.H. Gandomi, Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer, *Expert Syst. Appl.* 191 (2022), 116158.
- [56] S.J. Zhao, T.R. Zhang, S.L. Ma, M. Chen, Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications, *Eng. Appl. Artif. Intel.* 114 (2022), 105075.
- [57] L.Y. Wang, Q.J. Cao, Z.X. Zhang, S. Mirjalili, W.G. Zhao, Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems, *Eng. Appl. Artif. Intel.* 114 (2022), 105082.
- [58] L. Wang, R.X. Yang, H.Q. Ni, W. Ye, M.R. Fei, P.M. Pardalos, A human learning optimization algorithm and its application to multi-dimensional knapsack problems, *Appl. Soft Comput.* 34 (2015) 736–743.
- [59] Y.L. Yuan, J.J. Ren, S. Wang, Z.X. Wang, X.K. Mu, W. Zhao, Alpine skiing optimization: A new bio-inspired optimization algorithm, *Adv. Eng. Softw.* 170 (2022), 103158.
- [60] S. Mirjalili, SCA: A Sine Cosine Algorithm for solving optimization problems, *Knowl.-Based Syst.* 96 (2016) 120–133.
- [61] M. Abdechiri, M.R. Meybodi, H. Bahrami, Gases Brownian Motion Optimization: an Algorithm for Optimization (GBMO), *Appl. Soft Comput.* 13 (5) (2013) 2932–2946.
- [62] R. Storn, Differential evolution-a simple and efficient heuristic for global optimization over continuous space, *J. Glob. Optim.* 11 (1997).
- [63] D. Simon, Biogeography-Based Optimization, *IEEE Trans. Evol. Comput.* 12 (6) (2008) 702–713.
- [64] H.R. Lourenco, O.C. Martin, T. Stützle, Iterated local search, *Handbook of Metaheuristics*. (2003) 320–353.
- [65] W.G. Zong, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, *Simulation* 76 (2) (2001) 60–68.
- [66] W. Zhao, L. Wang, Z. Zhang, Supply-Demand-Based Optimization: A Novel Economics-Inspired Algorithm for Global Optimization, *IEEE Access* 7 (2019) 73182–73206.
- [67] Y. Tan, et al., Fireworks Algorithm for Optimization. 1st International Conference on Swarm Intelligence. (2010) 355–64.
- [68] M. Mirrashid, H. Naderpour, Transit search: An optimization algorithm based on exoplanet exploration, *Results Control Optim.* 7 (2022), 100127.
- [69] Y.L. Yuan, Q.K. Yang, J.J. Ren, J.K. Fan, Q.L. Shen, X.B. Wang, Y. Zhao, Learning-imitation strategy-assisted alpine skiing optimization for the boom of offshore drilling platform, *Ocean Eng.* 278 (2023), 114317.
- [70] Y.L. Yuan, Q.L. Shen, W.H. Xi, S. Wang, J.J. Ren, J.G. Yu, Q.K. Yang, Multidisciplinary design optimization of dynamic positioning system for semi-submersible platform, *Ocean Eng.* 285 (2023), 115426.
- [71] S. Aslan, S. Demirci, An improved immune plasma algorithm with a regional pandemic restriction, *SIViP* 16 (8) (2022) 2093–2101.
- [72] A.A. Dehkordi, A.S. Sadiq, S. Mirjalili, K.Z. Ghafoor, Nonlinear-based Chaotic Harris Hawks Optimizer: Algorithm and Internet of Vehicles application, *Appl. Soft Comput.* 109 (2021), 107574.
- [73] S. Ahmed, K.H. Sheikh, S. Mirjalili, R. Sarkar, Binary Simulated Normal Distribution Optimizer for feature selection: Theory and application in COVID-19 datasets, *Expert Syst. Appl.* 200 (2022), 116834.
- [74] Y.L. Yuan, S. Wang, L.Y. Lv, X.G. Song, An adaptive resistance and stamina strategy-based dragonfly algorithm for solving engineering optimization problems, *Eng. Comput.* 38 (5) (2021) 2228–2251.
- [75] Y.L. Yuan, X.K. Mu, X.Y. Shao, J.J. Ren, Y. Zhao, Z.X. Wang, Optimization of an auto drum fashioned brake using the elite opposition-based learning and chaotic k-best gravitational search strategy based grey wolf optimizer algorithm, *Appl. Soft Comput.* 123 (2022), 108947.
- [76] S. Arslan, S. Aslan, A new lattice based artificial bee colony algorithm for EEG noise minimization, *J. Faculty Eng. Arch. Gazi Univ.* 38 (1) (2023) 15–27.
- [77] S. Shadranian, H.R. Naji, V.K. Bardsiri, The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems, *Eng. Appl. Artif. Intell.* 80 (2019) 20–34.
- [78] A.M. Yibre, B. Koçer, Improving artificial algae algorithm performance by predicting candidate solution quality, *Expert Syst. Appl.* 150 (2020), 113298.
- [79] S. Zhang, X.C. Ye, X.J. Gan, Y.J. Lv, Growth characteristic of pangasius sanitwongsei cultured artificially, *J. Southern Agric.* 46 (9) (2015) 1710–1714.
- [80] J.X. Yang, X.Y. Chen, Y.R. Chen, On the population status and migration of pangasid catfishes in Lancangjiang River basin, China, *Zool. Res.* 28 (1) (2007) 63–67.
- [81] B. Yoğurtçuoglu, F.G. Ekmekçi, First record of the giant pangasius, *Pangasius sanitwongsei* (Actinopterygii: Siluriformes: Pangasiidae), from central Anatolia, Turkey, *Acta Ichthyologica Et Piscatoria.* 48 (3) (2018) 241–244.
- [82] T. Mäkinen, O.L.F. Weyl, K.A. van der Walt, E.R. Swartz, First record of an introduction of the giant pangasius, *Pangasius sanitwongsei* Smith 1931, into an African river, *Afr. Zool.* 48 (2) (2013) 388–391.
- [83] Z. Hogan, U. Na-Nakorn, H. Kong, Threatened fishes of the world: *Pangasius sanitwongsei* Smith 1931 (Siluriformes: Pangasiidae), *Environ. Biol. Fishes.* 84 (3) (2009) 305–306.
- [84] L. Xie, T. Han, H. Zhou, Z.R. Zhang, B. Han, A.D. Tang, Tuna Swarm Optimization: A Novel Swarm-Based Metaheuristic Algorithm for Global Optimization, *Comput. Intell. Neurosci.* 9210050 (2021).
- [85] J.D. Koehn, A.J. Hobday, M.S. Pratchett, B.M. Gillanders, Climate change and Australian marine and freshwater environments, fishes and fisheries: synthesis and options for adaptation, *Mar. Freshw. Res.* 62 (9) (2011) 1148–1164.
- [86] J.J. O'Connor, E.K. Fobert, M. Besson, H. Jacob, D. Lechini, Live fast, die young: Behavioural and physiological impacts of light pollution on a marine fish during larval recruitment, *Mar. Pollut. Bull.* 146 (2019) 908–914.
- [87] J.J. Liang, B.Y. Qu, D.W. Gong, C.T. Yue, Problem Definitions and Evaluation Criteria for the CEC 2019 Special Session on Multimodal Multiobjective Optimization, (2019).
- [88] D. Yazdani, J. Branke, M.N. Omidvar, X.D. Li, C.H. Li, M. Mavrovouniotis, T.T. Nguyen, S.X. Yang, X. Yao, IEEE CEC 2022 competition on dynamic optimization problems generated by generalized moving peaks benchmark, (2021).
- [89] E. Jahani, M. Chizari, Tackling global optimization problems with a novel algorithm – Mouth Brooding Fish algorithm, *Appl. Soft Comput.* 62 (2018) 987–1002.
- [90] H.M. Jia, X.X. Peng, C.B. Lang, Remora optimization algorithm, *Expert Syst. Appl.* 185 (2021), 115665.
- [91] M. Braik, A. Hammouri, J. Atwan, M.A.A. Al-Betar, M.A. Awadallah, White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems, *Knowl.-Based Syst.* 243 (2022), 108457.
- [92] G. Hu, M. Li, X.F. Wang, G. Wei, C.T. Chang, An enhanced manta ray foraging optimization algorithm for shape optimization of complex CCG-Ball curves, *Knowl.-Based Syst.* 240 (2022), 108071.
- [93] D. Zaldívar, B. Morales, A. Rodríguez, A. Valdivia-G, E. Cuevas, M. Pérez-Cisneros, A novel bio-inspired optimization model based on Yellow Saddle Goatfish behavior, *Biosystems* 174 (2018) 1–21.
- [94] H. Shayanfar, F.S. Gharehchopogh, Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems, *Appl. Soft Comput.* 71 (2018) 728–746.
- [95] B. Abdollahzadeh, F.S. Gharehchopogh, S. Mirjalili, Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems, *Int. J. Intell. Syst.* 36 (10) (2021) 5887–5958.
- [96] B. Abdollahzadeh, F.S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems, *Comput. Ind. Eng.* 158 (2021), 107408.
- [97] B. Abdollahzadeh, F.S. Gharehchopogh, N. Khodadadi, S. Mirjalili, Mountain Gazelle Optimizer: A new Nature-inspired Metaheuristic Algorithm for Global Optimization Problems, *Adv. Eng. Softw.* 174 (2022), 103282.
- [98] C.T. Zhong, G. Li, Z. Meng, Beluga whale optimization: A novel nature-inspired metaheuristic algorithm, *Knowl.-Based Syst.* 251 (2022), 109215.
- [99] A. Srivastava, D.K. Das, A bottleneck dolphin optimizer: An application to solve dynamic emission economic dispatch problem in the microgrid, *Knowl.-Based Syst.* 243 (2022), 108455.
- [100] M. Dehghani, Z. Montazeri, E. Trojovská, P. Trojovsky, Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems, *Knowl.-Based Syst.* 259 (2023), 110011.
- [101] S. Aslan, T. Oktay, Path Planning of an Unmanned Combat Aerial Vehicle with an Extended-Treatment-Approach-Based Immune Plasma Algorithm, *Aerospace* 10 (5) (2023) 487.
- [102] H. Tang, J. Lee, Adaptive initialization LSHADE algorithm enhanced with gradient-based repair for real-world constrained optimization, *Knowl.-Based Syst.* 246 (2022), 108696.
- [103] A. Kumar, G. Wu, M.Z. Ali, R. Mallipeddi, P.N. Suganthan, S. Das, A test-suite of non-convex constrained optimization problems from the real-world and some baseline results, *Swarm Evol. Comput.* 56 (2020), 100693.
- [104] G. Hu, Y.X. Guo, J.Y. Zhong, G. Wei, IYDSE: Ameliorated Young's double-slit experiment optimizer for applied mechanics and engineering, *Comput. Methods Appl. Mech. Eng.* 412 (2023), 116062.
- [105] Q. Li, S.Y. Liu, X.S. Yang, Influence of initialization on the performance of metaheuristic optimizers, *Appl. Soft Comput.* 91 (2020), 106193.
- [106] G. Hu, B. Du, X.F. Wang, G. Wei, An enhanced black widow optimization algorithm for feature selection, *Knowl.-Based Syst.* 235 (2022), 107638.
- [107] B. Mohan, J. Badra, A novel automated SuperLearner using a genetic algorithm-based hyperparameter optimization, *Adv. Eng. Softw.* 175 (2023), 103358.
- [108] E.H. Houssein, B. E-d, D. Helmy, A.A. Oliva, H.S. Elngar, A novel Black Widow Optimization algorithm for multilevel thresholding image segmentation, *Expert Syst. Appl.* 167 (2021), 114159.
- [109] G. Hu, J.Y. Zhong, G. Wei, SaCHBA-PDN: Modified honey badger algorithm with multi-strategy for UAV path planning, *Expert Syst. Appl.* 223 (2023), 119941.
- [110] S. Aslan, T. Erkin, A multi-population immune plasma algorithm for path planning of unmanned combat aerial vehicle, *Adv. Eng. Inf.* 55 (2023), 101829.
- [111] S. Aslan, T. Erkin, An immune plasma algorithm based approach for UCAV path planning, *J. King Saud Univ. – Comput. Inform. Sci.* 35 (2023) 56–69.
- [112] G. Hu, R. Yang, G. Wei, Hybrid chameleon swarm algorithm with multi-strategy: A case study of degree reduction for disk Wang-Ball curves, *Math. Comput. Simul.* 206 (2023) 709–769.
- [113] G. Hu, M. Li, J.Y. Zhong, Combined cubic generalized ball surfaces: Construction and shape optimization using an enhanced JS algorithm, *Adv. Eng. Softw.* 176 (2023), 103404.
- [114] G. Hu, R. Yang, X.Q. Qin, G. Wei, MCSA: Multi-strategy boosted chameleon-inspired optimization algorithm for engineering applications, *Comput. Meth. Appl. Mech. Eng.* 403 (2023), 115676.
- [115] G. Hu, J.Y. Zhong, G. Wei, C.T. Chang, DTCSMO: An efficient hybrid starling murmuration optimizer for engineering applications, *Comput. Meth. Appl. Mech. Eng.* 405 (2023), 115878.