Expert Systems With Applications 177 (2021) 114864

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts

Yutao Yang [a], Huiling Chen [a,*], Ali Asghar Heidari [b,c,*,1], Amir H Gandomi [d]

[a] College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, Zhejiang 325035, China
[b] School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran 1439957131, Iran
[c] Department of Computer Science, School of Computing, National University of Singapore, Singapore 117417, Singapore
[d] Faculty of Engineering & Information Technology, University of Technology Sydney, NSW 2007, Australia

ARTICLE INFO

Keywords:
Hunger Games Search
Optimization
Swarm-intelligence
Metaheuristic
Engineering design problems

ABSTRACT

A recent set of overused population-based methods have been published in recent years. Despite their popularity, most of them have uncertain, immature performance, partially done verifications, similar overused metaphors, similar immature exploration and exploitation components and operations, and an insecure tradeoff between exploration and exploitation trends in most of the new real-world cases. Therefore, all users need to extensively modify and adjust their operations based on main evolutionary methods to reach faster convergence, more stable balance, and high-quality results. To move the optimization community one step ahead toward more focus on performance rather than change of metaphor, a general-purpose population-based optimization technique called Hunger Games Search (HGS) is proposed in this research with a simple structure, special stability features and very competitive performance to realize the solutions of both constrained and unconstrained problems more effectively. The proposed HGS is designed according to the hunger-driven activities and behavioural choice of animals. This dynamic, fitness-wise search method follows a simple concept of "Hunger" as the most crucial homeostatic motivation and reason for behaviours, decisions, and actions in the life of all animals to make the process of optimization more understandable and consistent for new users and decision-makers. The Hunger Games Search incorporates the concept of hunger into the feature process; in other words, an adaptive weight based on the concept of hunger is designed and employed to simulate the effect of hunger on each search step. It follows the computationally logical rules (games) utilized by almost all animals and these rival activities and games are often adaptive evolutionary by securing higher chances of survival and food acquisition. This method's main feature is its dynamic nature, simple structure, and high performance in terms of convergence and acceptable quality of solutions, proving to be more efficient than the current optimization methods. The effectiveness of HGS was verified by comparing HGS with a comprehensive set of popular and advanced algorithms on 23 well-known optimization functions and the IEEE CEC 2014 benchmark test suite. Also, the HGS was applied to several engineering problems to demonstrate its applicability. The results validate the effectiveness of the proposed optimizer compared to popular essential optimizers, several advanced variants of the existing methods, and several CEC winners and powerful differential evolution (DE)-based methods abbreviated as LSHADE, SPS_L_SHADE_EIG, LSHADE_cnEpSi, SHADE, SADE, MPEDE, and JDE methods in handling many single-objective problems. We designed this open-source population-based method to be a standard tool for optimization in different areas of artificial intelligence and machine learning with several new exploratory and exploitative features, high performance, and high optimization capacity. The method is very flexible and scalable to be extended to fit more form of optimization cases in both structural aspects and application sides. This paper's source codes, supplementary files, Latex and office source files, sources of plots, a brief version and pseudocode, and an open-source software toolkit for solving optimization problems with Hunger Games Search and online web service for any question, feedback, suggestion, and idea on HGS algorithm will be available to the public at https://aliasgharheidari.com/HGS.html.

* Corresponding authors.
E-mail addresses: yutaoyang.style@foxmail.com (Y. Yang), chenhuiling.jlu@gmail.com (H. Chen), as_heidari@ut.ac.ir, aliasgha@comp.nus.edu.sg, t0917038@u.nus.edu (A.A. Heidari), gandomi@uts.edu.au (A.H. Gandomi).
[1] https://aliasgharheidari.com

https://doi.org/10.1016/j.eswa.2021.114864
Received 19 September 2019; Received in revised form 28 February 2021; Accepted 4 March 2021
Available online 10 March 2021

1. Introduction

The real-world applications in expert systems, information systems, and knowledge-based systems often have a limited feature space and constraints based on the priorities and budget limits of the project owners. Decision-makers, developers and computer scientists need to find some feasible, explainable and sufficient details and solutions during a reasonable time using any family of deterministic or approximated algorithms for problems in different areas such as image segmentation (Abd Elaziz, Heidari, Fujita, & Moayedi, 2020; Zhao, Liu, Yu, Heidari, Wang, Liang, et al., 2020; Zhao, Liu, Yu, Heidari, Wang, Oliva, et al., 2020), location-based services (Li, Zhu, & Wang, 2019), opportunistic networks (Fu, Fortino, Li, Pace, & Yang, 2019), supply chain development (Chen, He, Guan, Lu, & Li, 2017), hydrothermal systems (Deng, Zhang, Sharma, & Nie, 2019), engineering applications (Ba et al., 2020; Liang et al., 2020), video deblurring (Wang, Zhang et al., 2020; Zhang, Jiang, Wang, Huang, & Zhao, 2020), social recommendation and QOS-aware service composition (Li & Lin, 2020; Li, Chen, Chen, & Tong, 2017; Li, Zheng, Chen, Song, & Chen, 2014), image recovery and alignment (Zhang, Wang, Zhou, & Ma, 2019), recognizing landmark architectures (Li, Zhang, & Zhang, 2016), human articulated body recognition (Li, Liu, Zhang, & Ye, 2016), secure encryption (Zhou et al., 2019), image filtering (Zhao, Gao, Wang, Pan, & Graphics, 2016; Zhao, Jiang, Jin, Du, & Li, 2018), image editing (Li et al., 2016; Li, Huang, Zhao, Wang, & Hu, 2020; Zhou, Tian, Zhu, Jin, & Sun, 2019), structural topology optimization (Zhang, Li, Zhong, & Xiang, 2014), scheduling problem (Pang, Zhou, Tsai, & Chou, 2018; Zhou, Pang, Chen, & Chou, 2018), face recognition and micro-expression recognition (Wang, Chen, Yan, Chen, & Fu, 2014; Wang et al., 2017), gold price prediction (Wen, Yang, Gong, & Lai, 2017), epileptic seizure detection (Li, et al., 2019, 2020), and wireless communication and network systems (Wu, Yang, Xu, Zhao, & Liu, 2014; Xiong et al., 2020; Xu, Shi, & Networks, 2015), to name a few potential areas for future users in optimization and artificial intelligence (AI) community. On the other hand, the effectiveness and complexity of the developed solvers is a central concern when the characteristics of the problems get more dynamic or complicated in terms of multimodality, uncertainty and vagueness of feature space. For instance, we can point to a set of applications in cross-field and computer science such as bankruptcy prediction (Wang, Chen, Li et al., 2017; Yu et al., 2021), prediction problems in educational field (Wei et al., 2020; Zhu, Ma et al., 2020), brain disease diagnosis (Fei, Wang, Ying, Hu, & Shi, 2020), thyroid cancer diagnosis (Xia et al., 2017), tuberculous pleural effusion diagnosis (Li et al., 2018), paraquat-poisoned patients diagnosis (Hu et al., 2017; Hu, Hong, Ma, Wang, & Chen, 2015; Zhao et al., 2019), parkinson disease diagnosis (Chen et al., 2016), and other medical problems (Chen et al., 2015; Liu et al., 2016; Shen et al., 2016; Sun et al., 2013; Wang & Chen, 2020; Wang, Chen, Yang et al., 2017), decision-making (Wu et al., 2020), and smart agriculture (Song, Zhong et al., 2020). Optimization is a "should" behind most of the AI and industrial problems in different disciplines such as deep learning (Wang, Xu, Li, Wang, & Song, 2018). It can be in the form of single-objective that we need to prepare all objective in a single known function, but it has been extended to many more forms such as multiobjective (Cao, Zhao, Yang, et al., 2019), robust optimization , many objectives (Cao et al., 2020), fuzzy optimization (Huang & Feng, 2019), large scale optimization (Cao et al., 2020), and memetic methods. There are also two philosophical viewpoints to deal with problems and mathematical models that one of them rely on the utilization of the gradient and deterministic equations when solving the problem (Long et al., 2015) and another viewpoint has a trial and error nature using recursive sensing and evaluating the landscape of the problem based on some approximated metrics and info about the problem basin or in a stochastic way. Evolutionary and swarm-based optimization method or metaheuristic methods are widely used approach in this class with many applications (Chen, Li, et al., 2020; Chen, Wang, & Zhao, 2020; Chen, Xu, Wang, & Zhao, 2019; Luo et al., 2019, 2018; Song, Wang et al., 2020; Tu et al., 2020; Wang, Chen et al., 2020; Yu, Zhao, Wang, Chen, & Li, 2020; Zhang, Xu et al., 2020; Zhang, Liu et al., 2020).

Finding optimal solutions to multimodal rotated, or composition problems is a difficult task without having any gradient information about an objective function. Over the past few years, users have become more interested in estimating the best solutions, then utilizing these solutions depending on their accuracy level (Jiang et al., 2017). Hence, the meta-heuristic algorithms (MAs) have attracted substantial attention, and they have been applied to various fields of machine learning, engineering, and science. The main reason for such a trend is that there is an overflow of new problems in the real world and, as such, increasing demands for these solvers when the problems become more challenging. The characteristics of MAs, such as avoiding local optimum, simplicity, and gradient-free steps, makes it possible to provide satisfactory solutions to such complex problems, which typically have many local optima and challenging search space. Dealing with multimodal spaces with iterative exploratory and exploitative procedures is the central feature of all MAs in literature.

Nevertheless, there are also some gaps, concerns, and drawbacks within the previous swarm-based optimization methods (Bai, Guo, Zhou, Zhang, & Zhang, 2021). Recently, some popular methods have been proposed that are based on the characters of animals. However, various studies revealed that performance of these methods were not studied deeply in the original work and their mathematical models also suffer from structural defects, mediocre performance, problematic verification methods , the apparent similarity in their structure, and slightly modified components (Hu et al., 2021). As per our rich experience on these methods (Chen, Heidari, Zhao, Zhang, & Chen, 2020; Chen, Yang, Heidari, & Zhao, 2019; Yu, Heidari, & Chen, 2020), such issues affect their reliability in the optimization community without sufficient attention to the performance aspects, complexity, the tuning of parameters, comparison with advanced and high-tech optimizers, verification using CEC competition sets, and wise interactions among the components. These aspects play significant roles when decision-makers or practitioners need to deal with some real-world problems (Xue et al., 2020). These disputes motivated us to investigate the algorithmic behaviours further and develop a more stable logic, especially considering that these popular methods require much effort and modifications to jump out of local optima and stagnation and their shortcomings. Although general users in industry and inexperienced code users can barely detect these issues, these methods are still difficult to understand. Hence, we attempted to highlight more aspects in this research and compared them to other methods to shift the preferences of the field toward the performance.

The focus of almost all methods is to iteratively evolve the population that appeared in the genetic algorithm (GA) (Holland, 1992) and particle swarm optimizer (PSO), which were later divided into the evolutionary algorithm (EA) and Swarm Intelligence (SI) optimizers. Biological evolutionary operations support the logic of evolutionary algorithms and can tackle optimization problems by three operations: selection, reorganization, and mutation. The GA (Holland, 1992) is a basic EA proposed by Holland based on Darwin's theory of evolution. Simulating organisms' evolution or the ideal solution can be performed in the solution space. The evolutionary process of the Differential Evolution (DE) algorithm (Storn & Price, 1997) is very similar to that of GA, but its specific definition of operation is different. At the same time, it uses the cooperative relationship between individuals within a group and the swarm intelligence generated by competition to guide the direction of evolution. Besides, EA includes Genetic Programming (GP) (Koza & Rice, 1992), Evolution Strategy (ES) (Hansen, Müller, & Koumoutsakos, 2003), and Evolutionary Programming (EP) (Yao, Liu, & Lin, 1999).

SI mainly simulates natural organisms' collective behaviour and uses social wisdom to search for optimal searching space cooperatively. Ant Colony Optimization (ACO) (Dorigo & Blum, 2005) simulates the food collection conducted in ant colonies and has been applied in many

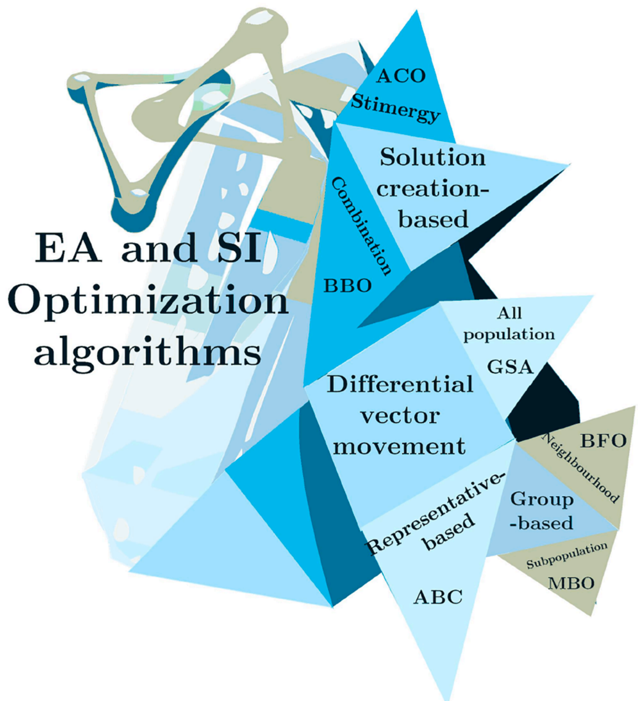

Fig. 1. Classification of optimizers based on the behaviour taxonomy.

discrete problems. PSO (Kennedy & Eberhart, 1995) mimics the regularity of bird clusters' activities, using information sharing among individuals in the group to move the whole group. In addition to the above two representative algorithms, the more recent SI algorithms are biogeography-based optimization (BBO) (Simon, 2008), Monarch Butterfly Optimization (MBO) (Wang, Deb, & Cui, 2019), Cuckoo Search (CS) (Yang & Deb, 2009), Artificial Bee Colony (ABC) (Karaboga & Basturk, 2007), and Harris Hawk Optimizer (HHO)[2] (Heidari, Mirjalili, Faris, Aljarah, Mafarja, & Chen, 2019). Another promising method is the slime mould algorithm (SMA)[3] (Li, Chen, Wang, Heidari, & Mirjalili, 2020), which has been recently developed based on slime mould and is gaining more attention from experts. In Fig. 1, the classification of methods based on algorithmic behaviours is shown. Please refer to the original research presented by (Molina et al., 2020) for complete data and further study.

Although MAs are divided into several categories, they share the same characteristics in that the search steps consist of exploration and exploration. In the first stage, we need to ensure the randomness of the search as much as possible and explore the search space broadly. In the second stage, we need to accurately focus on specific regions of the feature space found in the previous stage. A promising area focuses on the local search capacity, so balancing these two stages is crucial to the algorithm's performance.

Although many MAs have been proposed, there is no free lunch (Wolpert & Macready, 1997) in the world, and no algorithm can solve all optimization problems as the best method. Since each algorithm shows superiority in some specific optimization problems, researchers continuously work to explore and develop better algorithms. Hence, this work proposes a new meta-heuristic algorithm, Hunger Games Search (HGS), which is inspired by social animals' cooperative behaviour where search activity is proportional to their level of hunger. This algorithm is designed and implemented based on the common characteristics of social animals and their food search.

The remainder of this paper is structured as follows. Section 2 expounds on the enlightenment of HGS and establishes the corresponding mathematical model. Section 3 depicts the experiments involved in this work, qualitative analysis, and comparison with traditional and advanced algorithms on 23 benchmark functions, and IEEE CEC benchmark functions, and application to engineering problems. Section 4 summarizes the full text and future research direction.

2. Hunger Games search (HGS)

In this chapter, the HGS algorithm's details, along with its mathematical model, will be introduced.

2.1. Logic of search, behavioural choice, and hunger-driven games

Animal follows their sensory info based on some computational rules and in interaction with their environment (as a part of their environment) that these rules make the basis of their decisions and choices and support for the evolution of their cognitive architecture. It is verified that these computationally logical rules utilized by animals will often be adaptive evolutionary by securing higher chances of survival, reproduction, and food acquisition (Real, 1991). Hunger is responsible for one of the most crucial homeostatic motivations and reasons for behaviours, decisions, and actions in the life of animals. In spite of the wide variety of stimuli and competing demands that always and certainly impinge upon the quality of life of animals, they should select and pursue food sources when they face caloric insufficiency. To deal with this homeostatic imbalance, they must regularly search for food and move around their surroundings in ways that need switching between exploratory, defensive, and competing activities, indicating incredible smoothness in feeding strategies (Burnett et al., 2016).

The behavioural choice and choice of activity is universal in the animal monarchy, and it is a fundamental law to goal-oriented behaviours witnessed in nature. Various factors or a combination of them affect the behaviour of species, and the observed behaviours are subject to existing motivational state and the occurrence of stimuli in their locality (Reppucci & Veenema, 2020). For any animal, neuroscientists agree that the hunger[4] is a strong motivating force for activity, learning, and searching for food and it acts as a force toward changing the life condition to a more stable state[5] (Sutton & Krashes, 2020). Hunger can surpass and influence competing drives states such as thirstiness, nervousness, fear of hunters, and communal requirements, according to experiments in "*Hunger-Driven Motivational State Competition*" published at *Neuron* (Burnett et al., 2016). Hence, neuroscientists discovered that hunger possibly is at the top of the motivation hierarchy[6]. Hunger also trumps communal desires for animals when they can find the food and consume it (Burnett et al., 2016).

Social life helps animals to avoid predators and find food sources, both other animals and vegetables, as they work in natural collaboration, which enhances their chance of survival. This is the nature of evolution, whereby healthier animals can find sources of food better and have a greater chance of survival over weaker animals. This can be

[2] The info and source codes of the HHO algorithm are publicly available at https://aliasgharheidari.com/HHO.html

[3] The info and source codes of the SMA algorithm are publicly available at https://aliasgharheidari.com/SMA.html

[4] For a deep meaning of the hunger word, interested readers can also refer to (Cannon & Washburn, 1912)

[5] To read more about the motivations, preferences and choices of animals, interested readers can read more at: https://www.nature.com/scitable/knowledge/library/measuring-animal-preferences-and-choice-behavior-23590718/

[6] For more info and learning, interested readers can watch a supplementary video at https://tinyurl.com/aliasgharheidaridotcom or https://www.sciencedirect.com/science/article/pii/S0896627316305256

termed as a *hunger games* in nature. Any wrong decision may change the game's outcome, leading to the death of an individual or even extinction of an entire species. For example, after hunting, ravens and rats tell their companions that their next meal reduces the uncertainty of their next meal (Jarvandi, Booth, & Thibault, 2007). The daily behaviour of animals is highly influenced by some motivational situations, such as hunger and nervousness of being killed by hunters (Gotceitas & Godin, 1991). Hunger is a characteristic of "not eating" for a long time (Miller, Bailey, & Stevenson, 1950), whereby the stronger the hunger, the stronger the craving for food, and the more active the organism will be in searching for food in a short time before it gets too late and causes starvation or death (Friedman & Stricker, 1976). Otherwise, the chance of survival will be too low, and the animal dies. Hence, when the source of food is limited, there is a logical game between hungry animals to find the source of food and win the situation (O'brien, Browman, & Evans, 1990). The game is thus based on the logical decisions and motions of species.

2.2. Mathematical model

In this sub-section, the mathematical model and proposed HGS method are described in detail. Please note that we are constrained to build a mathematical model according to the hunger-driven activities and behavioural choice, and it should be as simple as possible and at the same time, most efficient performance.

2.2.1. Approach food

Social animals often cooperate with each other during foraging, but the possibility that a few individuals do not participate in the collaboration is not excluded (Clutton-Brock, 2009).

The following game instructions represent the central equation of the HGS algorithm for individual cooperative communication and foraging behaviour:

$$\overrightarrow{X(t+1)} = \begin{cases} Game_1: \overrightarrow{X(t)} \cdot (1 + randn(1)), r_1 < l \\ Game_2: \overrightarrow{W_1} \cdot \overrightarrow{X_b} + \overrightarrow{R} \cdot \overrightarrow{W_2} \cdot \left| \overrightarrow{X_b} - \overrightarrow{X(t)} \right|, r_1 > l, r_2 > E \\ Game_3: \overrightarrow{W_1} \cdot \overrightarrow{X_b} - \overrightarrow{R} \cdot \overrightarrow{W_2} \cdot \left| \overrightarrow{X_b} - \overrightarrow{X(t)} \right|, r_1 > l, r_2 < E \end{cases} \tag{2.1}$$

where $\overrightarrow{R}$ is in the range of $[-a, a]$;

r_1 and r_2 represent two random numbers, which are in the range of $[0, 1]$;
$randn(1)$ is a random number satisfying normal distribution;
t indicates the current iterations;
$\overrightarrow{W_1}$ and $\overrightarrow{W_2}$ represent the weights of hunger; which we designed them based on the fact of hunger-driven signals (Betley et al., 2015);
$\overrightarrow{X_b}$ represents the location of the best individual of this iteration;
$\overrightarrow{X(t)}$ represents each individual's location;
the value of l will be discussed in the parameter setting experiment, and it is a parameter which is designed to improve the algorithm.
$\overrightarrow{X(t)} \cdot (1 + randn(1))$ represents how an agent can search for food hungrily and randomly at the current location;
$\left| \overrightarrow{X_b} - \overrightarrow{X(t)} \right|$ models the *range of activity* of the current individual in the current time and it is multiplied by $\overrightarrow{W_2}$ to affect the influence of hunger on the range of activity. Since an individual will stop

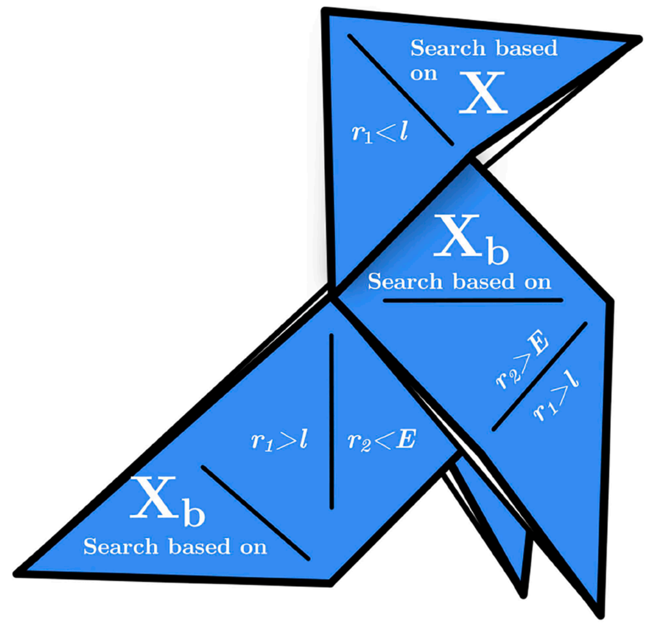

Fig. 2. The logic of Hunger Games Search (HGS) algorithm during optimization.

searching when it be not hungry anymore, $\overrightarrow{R}$ is a ranging controller added to limit the *range of activity*, in which the range of $\overrightarrow{R}$ is gradually reduced to 0. Adding or subtracting the *range of activity* based on $\overrightarrow{W_1} \cdot \overrightarrow{X_b}$ simulates the current individual informed by its peers when arriving at the food location, and then searching for food again at the current location after the acquisition of the food. $\overrightarrow{W_1}$ is introduced as the error in grasping the actual position in reality. The definition of the formula of E, which is a variation control for all positions, is as follows:

$$E = \operatorname{sech}(|F(i) - BF|) \tag{2.2}$$

where $i \in 1, 2, \cdots, n$, $F(i)$ represents the fitness value of each individual;
BF is the best fitness obtained in the current iteration process (so far);
Sech is a hyperbolic function $\left(\operatorname{sech}(x) = \frac{2}{e^x + e^{-x}}\right)$.
The formula for $\overrightarrow{R}$ is as follows:

$$\overrightarrow{R} = 2 \times shrink \times rand - shrink \tag{2.3}$$

$$shrink = 2 \times \left(1 - \frac{t}{T}\right) \tag{2.4}$$

where $rand$ is a random number in the range of $[0, 1]$; and T stands for the maximum number of iterations.

Fig. 2 displays the process of searching and logic of HGS in the spaces based on the rule in **Eq.** (2.1).

As can be seen in the graph, the search directions can be divided into two categories according to the classification of source points:

Search on the basis of $\overrightarrow{X}$: The first game instruction simulates the self-dependent one, which it has no teamwork spirit, and not involved in

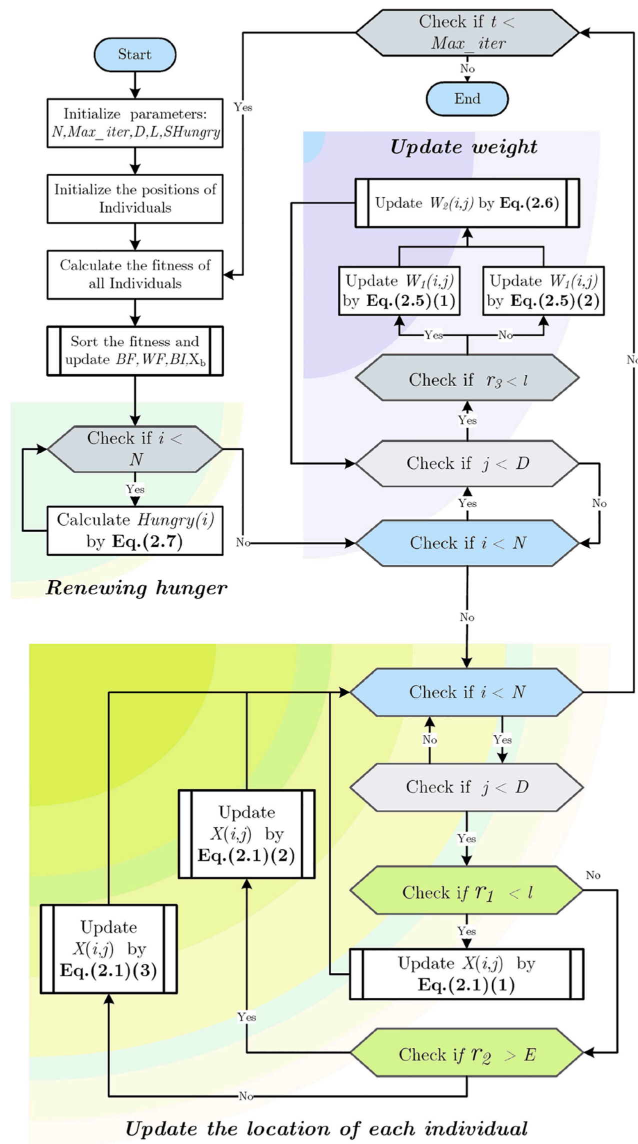

Fig. 3. Flowchart of Hunger Games Search (HGS) algorithm.

the cooperation phase and just wants to search for food hungrily.

Search on the basis of $\overrightarrow{X_b}$: The second game instruction is closely related to the variables $\overrightarrow{R}$, $\overrightarrow{W_1}$ and $\overrightarrow{W_2}$. By a refinement of these three factors, the individual's position can be evolved within the feature space. This method simulates the cooperation between several entities when they search for food.

The laws or rules in **Eq.** (2.1) allows individuals to explore possible locations near the optimal solution and locations far away from the optimal solution, which guarantees the search of all locations inside the boundaries of solution space to a certain extent (diversification). The same concept can also be applied to high-dimensional search space.

2.2.2. *Hunger role*

In this part, the starvation characteristics of individuals in search are simulated by a proposed model, mathematically.

The formula of $\overrightarrow{W_1}$ in **Eq.** (2.1) is as follows:

$$\overrightarrow{W_1(i)} = \begin{cases} hungry(i) \cdot \dfrac{N}{SHungry} \times r_4, r_3 < l \\ 1 r_3 > l \end{cases} \tag{2.5}$$

The formula of $\overrightarrow{W_2}$ in **Eq.** (2.1) is shown as follows:

$$\overrightarrow{W_2(i)} = (1 - exp(-|hungry(i) - SHungry|)) \times r_5 \times 2 \tag{2.6}$$

where *hungry* represents the *hunger* of each individual;

N represents the number of individuals;

SHungry is the sum of hungry feelings of all individuals, that is *sum*(*hungry*);

r_3,r_4 and r_5 are random numbers in the range of $[0,1]$.

The formula for *hungry*(i) is provided below:

$$hungry(i) = \begin{cases} 0, AllFitness(i) == BF \\ hungry(i) + H, AllFitness(i)! = BF \end{cases} \tag{2.7}$$

where *AllFitness*(i) preserves the fitness of each individual in the current iteration. In each iteration, the best individual's hunger was set to 0. For other individuals, a new hunger(H)is added based on the *original hunger*. Hence, we understand that the corresponding H of different individuals will be different.

The formula for H can be seen as follows:

$$TH = \frac{F(i) - BF}{WF - BF} \times r_6 \times 2 \times (UB - LB) \tag{2.8}$$

$$H = \begin{cases} LH \times (1 + r), TH < LH \\ TH, TH \geq LH \end{cases} \tag{2.9}$$

where r_6 is a random number in the range of $[0,1]$;

$F(i)$ represents the fitness value of each individual;

BF is the best fitness obtained in the current iteration process (so far);

WF stands for the worst fitness obtained in the current iteration process (so far);

UB and LB indicate the upper and lower bounds of the feature space, respectively.

The *hunger sensation* (Friedman, Ulrich, & Mattes, 1999) H is limited to a lower bound, LH, In order to make the algorithm get a better performance, we control the upper and lower limits of hunger, and the value of LH will be discussed in the parameter setting experiment.

Since *hunger* may have both positive and negative effects on the *range of activity*, $\overrightarrow{W_1}$ and $\overrightarrow{W_2}$are simulated.

In **Eq.** (2.8), the difference between UB and LB is used to reflect the individual's greatest hunger in different environments;

$F(i) - BF$ represents the amount of food that an individual still needs not to be hungry anymore; In every iteration, the hunger of individual will change.

$WF - DF$ yields the total foraging capacity of an individual in the current process;

$\frac{F(i)-DF}{WF-DF}$ represents the hunger ratio;

$r_6 \times 2$ gives the positive or negative effects of environmental factors on hunger.

While the HGS algorithm proposed can yield the commonness of social animals, it still has much room for improvement. For instance, the algorithm can be improved by mapping it according to a living organism

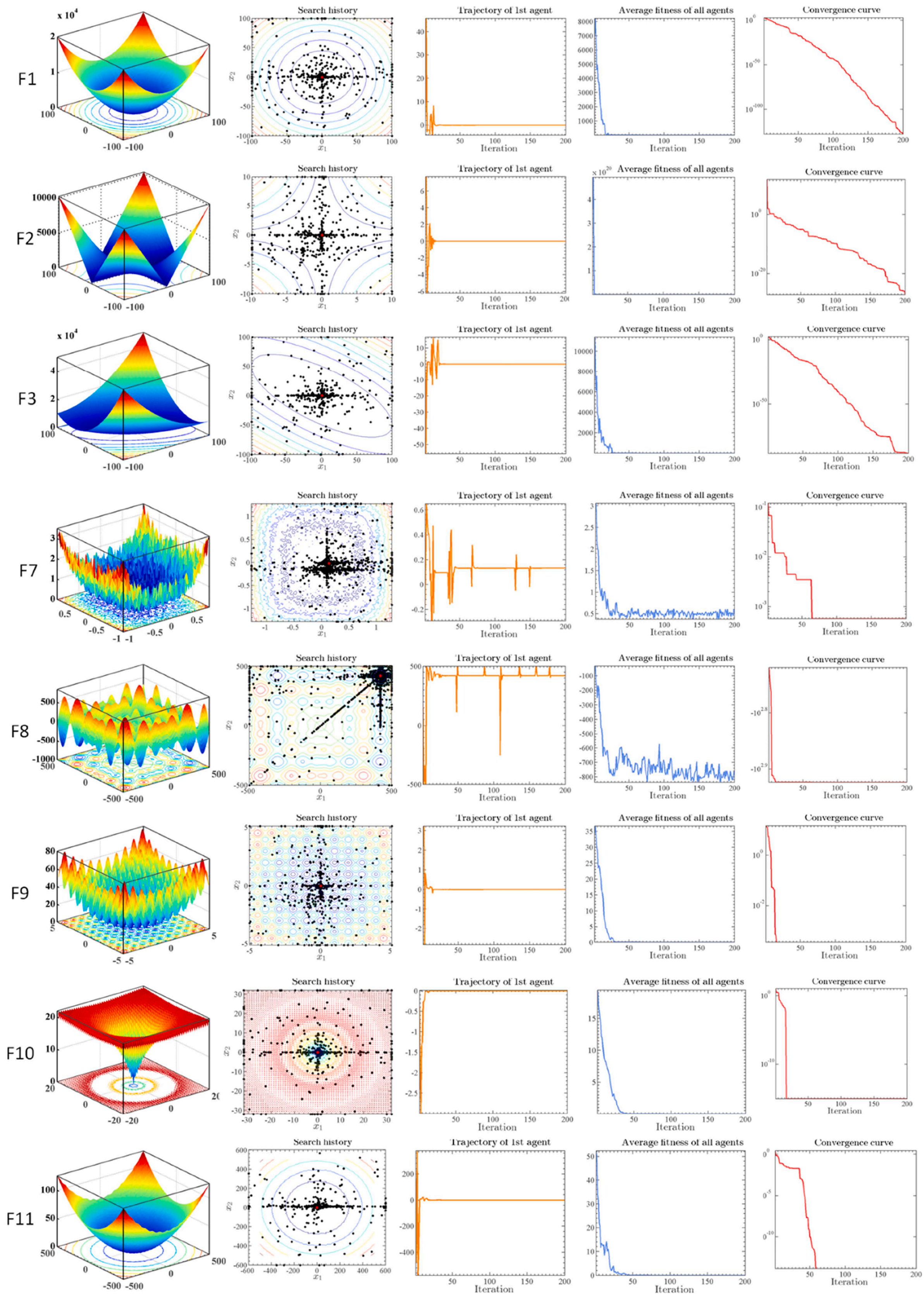

Fig. 4. Qualitative analysis of HGS on some typical functions.

Table 1
Description of the 23 benchmark functions.

ID	Function Equation	Dim	Range	f_{min}
F1	$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100,100]	0
F2	$f_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	30	[−10,10]	0
F3	$f_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	30	[−100,100]	0
F4	$f_4(x) = max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	30	[−100,100]	0
F5	$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[−30,30]	0
F6	$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[−100,100]	0
F7	$f_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	30	[−1.28,1.28]	0
F8	$f_8(x) = \sum_{i=1}^{n} -x_i sin(\sqrt{\lvert x_i\rvert})$	30	[−500,500]	−418.982
F9	$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10cos(2\pi x_i) + 10]$	30	[−5.12,5.12]	0
F10	$f_{10}(x) = -20exp\left\{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i}\right\} - exp\left\{\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right\} + 20 + e$	30	[−32,32]	0
F11	$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[−600,600]	0
F12	$f_{12}(x) = \frac{\pi}{n}\{10sin(ay_1) + \sum_{i=1}^{n-1} (y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2 + \sum_{i=1}^{n} \mu(x_i, 10, 100, 4)\}$	30	[−50,50]	0
F13	$f_{13}(x) = 0.1\{sin^2(3\pi x_i) + \sum_{i=1}^{n} (x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)] + \sum_{i=1}^{n} \mu(x_i, 5, 100, 4)$	30	[−50,50]	0
F14	$f_{14}(x) = \left(\frac{1}{5000} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i + a_{ij})^6}\right)^{-1}$	2	[−65,65]	1
F15	$f_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1({b_i}^2 + b_i x_2)}{{b_i}^2 + b_i x_3 + x_4}]^2$	4	[−5,5]	0.00030
F16	$f_{16}(x) = 4{x_1}^2 - 2.1{x_1}^4 + \frac{1}{3}{x_1}^6 + x_1 x_2 - 4{x_2}^2 + 4{x_2}^4$	2	[−5,5]	−1.0316
F17	$f_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}{x_1}^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	2	[−5,5]	0.398
F18	$f_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3{x_1}^2 - 14x_2 + 6x_1x_2 + 3{x_2}^2)\right] \times \left[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12{x_1}^2 + 48x_2 - 36x_1x_2 + 27{x_2}^2)\right]$	2	[−2,2]	3
F19	$f_{19}(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[1,3]	−3.86
F20	$f_{20}(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0,1]	−3.32
F21	$f_{21}(x) = -\sum_{i=1}^{5} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.1532
F22	$f_{22}(x) = -\sum_{i=1}^{7} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.4028
F23	$f_{23}(x) = -\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.5363

and integrating the living organism's unique characteristics. It can also be improved by adding other mechanisms. We simplify the algorithm as much as possible to maximize its scalability.

Algorithm 1 shows the pseudo-code of the proposed Hunger Games Search. Also, the flowchart is represented in Fig. 3.

Algorithm 1 Pseudo-code of Hunger Games Search (HGS)

```
Initialize the parameters N, T,l,D,SHungry
Initialize the positions of Individuals X_i(i = 1,2,···,N)
While (t ≤ T)
  Calculate the fitness of all Individuals
  UpdateBF, WF, X_b, BI
  Calculate the Hungry by Eq. (2.7)
  Calculate the W_1 by Eq. (2.5)
  Calculate the W_2 by Eq. (2.6)
  For eachIndividuals
    Calculate E by Eq. (2.2)
    UpdateR by Eq. (2.3)
    UpdatepositionsbyEq.(2.1)
  End For
  t = t + 1
End While
ReturnBF, X_b
```

2.3. Theoretical and structural qualities of the hunger Games search

As a gradient-free, population-based optimizer, the proposed HGS exhibits efficient performance due to the following unique advantages:

- It is a population-based method with stochastic switching elements that enrich its main exploratory and exploitative behaviours and flexibility of HGS in dealing with challenging problem landscapes.
- The adaptive and time-varying mechanisms of HGS allow this method to handle multi-modality, and local optima problems more effectively.
- The consideration of hunger ratio and influence of hunger on the range of activity make the HGS more flexible and capable of changing the performance in a fitness-wise fashion.
- The application of individual fitness values enables HGS to consider historical info if it is required to change the behaviour.
- Parameters l and E assist HGS in evolving the initial positions and search mode to ensure the exploration of the whole solution space as far as possible and enhance the diversification capacity of the algorithm to a great extent.
- The hunger weights $\overrightarrow{W_1}$ and $\overrightarrow{W_2}$ increase the perturbation of HGS during the search process and prevent the algorithm from trapping in a local optimum.
- The parameter $\overrightarrow{R}$ ensures that the search step of HGS is reduced at a specific rate; therefore satisfying the need to explore the target solution space in a broad range in the early stage and exploit the depth of the target search basin in the later stages.
- The Hunger Games Search can evolve the search agents with regards to best solutions (X_b) and normal solutions (X), which is a simple idea to ensure more exploration patterns and more coverage on the hidden areas of the feature space.The structure and logic of Hunger

Table 2
Parameter settings of the involved MAs.

Class	Algorithm	Parameter settings
Well-established	**BBO**	$elitism = 2; \lambda_{lower} = 0; \lambda_{upper} = 1; stepsize = 1$
	PSO	$c_1 = 2; c_2 = 2; vMax = 6; w = 1$
	DE	$scalingfactor = 0.5; crossoverprobability = 0.5$
	FA	$\alpha = 0.5; \beta = 0.2; \gamma = 1$
	BA	$A = 0.5; r = 0.5$
	FPA	$U\ N(0, \sigma^2); V\ N(0, 1); p = 0.5$
Recent methods	**SCA**	$A = 2$
	SSA	$c_1 \in [01]; c_2 \in [01];$
	GWO	$a = [2, 0]$
	MFO	$b = 1; t = [-1, 1]; a \in [-1, -2]$
	WOA	$a_1 = [2, 0]; a_2 = [-2, -1]; b = 1$
	GOA	$cmax = 1; cmin = 0.00001; b = 1$
	DA	$\beta = 1.5$
	ALO	$k = 500$
	MVO	$existenceprobability \in [0.21]; travellingdistancerate \in [0.61]$

Games Search are straightforward, and it is easy to be integrated with other evolutionary mechanisms for dealing with new practical problems in science and engineering.

- Despite the simple equations and compared to the existing methods, the Hunger Games Search has a very superior performance with high-quality results compared to well-known basic and advanced methods for studied benchmark problems.
- The codes of Hunger Games Search will be publicly available in different languages, and users can easily access the software codes and apply it to their target problem based on functional programming.
- An online, public web service at https://aliasgharheidari.com/HGS.html will be responsible for all users regarding any assistance and required supplementary material.

2.4. Computational complexity analysis

The proposed Hunger Games Search mainly includes the following parts: initialization, fitness evaluation, sorting, hunger updating, weight updating, and location updating. In the associated formulas, N indicates the number of individuals in the population, D is the dimension of the problem, and T represents the maximum quantity of iterations. During the initial stage, the computational complexity of fitness evaluation and hunger update are both $O(N)$, the computational complexity of sorting is $O(NlogN)$, and the computational complexity of weight and location update is $O(N \times D)$. From the above analysis, we can acquire the complexity of the whole algorithm: $O(N^*(1 + T^*N^*(2 + logN + 2^*D)))$.

3. Experiments and results

In this chapter, the proposed HGS algorithm is compared against some well-established counterparts. All experiments were conducted on a Windows Server 2008 R2 operating system with Intel (R) Xeon (R) CPU E5-2650 v4 (2.20 GHz) and 128 GB of RAM. All algorithms were coded in the MATLAB R2014b for a fair comparison.

3.1. Qualitative analysis

Fig. 4 shows the qualitative analysis of 23 well-known benchmark functions using the HGS algorithm, which includes the search history, trajectory of the first individual, average fitness of all individuals, and convergence behaviour. The search history shows the location and distribution of individuals in each iteration. The first individual's trajectory reveals the motion patterns of the first individual in the whole iteration process. The average fitness of all individuals monitors how the average fitness of the entire population changes during optimization. The convergence behaviour reveals the changing trend of optimal fitness and indirectly shows how well exploratory trends change to exploitative drifts.

By observing the individual's historical position, we can first observe that the individual has explored major portions of the search space, showcasing that the algorithm has a strong search ability and can avoid falling into a locally optimal solution. Simultaneously, we also see that most of the search locations are around the optimal solution, which indicates that the algorithm can accurately progress in the target area, and the convergence speed is fast. The algorithm has a good measurement between the two phases of exploration and exploration, in which we can detect the advantages of HGS.

The trajectory graph shows that individuals have strong fluctuations in the initial stage of the search, and the range of fluctuation coverage exceeds 50% of the solution space. This proves that the search ability of the HGS algorithm is very strong, and it can focus on high-quality solutions. As the number of iterations increases, the fluctuation tends to be more stable, which indicates that the algorithm has found a promising region, and it is still exploring the region. For some functions, such as F7 and F8, it is apparent that the fluctuation tends to stabilize and then oscillation occurs, meaning that the HGS algorithm can jump out of local optimum and avoid falling into local optima, which is also a validation of the balanced performance of the proposed algorithm.

The algorithm tends to converge very quickly in the early stages of iteration by monitoring the overall average fitness. Although the downward trend slows down with the iteration and is accompanied by variations, the average fitness gradually decreases, reflecting the well-prepared search and high searching capabilities of the algorithm. The convergence curve reveals the convergence speed of the algorithm and the time point of conversion between exploration and exploration. The convergence curves show that HGS can demonstrate a fast tendency in dealing with F8-F10, and there is no stagnation problem.

3.2. Validation on commonly used benchmark functions

In this part, we tested the proposed HGS algorithm on 23 benchmark unimodal and multi-modal functions. Details of these 23 functions can be found in Table 1, where Dim denotes the dimensions of the functions, Range refers to the definition domain of the function, and f_{min} reveals the optimal solution of the function.

One point is so critical in the verification of computational intelligence methods, and it is the detailed report of the used parameters for a fair, justifiable comparative analysis and the same conditions of test (Chen et al., 2020; Shi et al., 2018). This matter is to ensure the results of any kind of algorithm are gathered in the same condition and with no bias toward any specific method that used a better testing condition, as it followed by reference literature as well (Fan et al., 2020; Huang et al., 2020; Ni et al., 2020; Zhang et al., 2018).

For the experimental results' credibility, all experiments were conducted under the same conditions: population size was set to 30; and maximum iterations and dimensions were set to 1000 and 30, respectively. At the same time, to exclude the influence of random factors, we tested each algorithm 30 times. For this paper, the Friedman test (Derrac, García, Molina, & Herrera, 2011) and the Wilcoxon sign-rank test (García, Fernández, Luengo, & Herrera, 2010) were applied to identify the algorithms' significant differences. The Friedman test is a nonparametric statistical program that allows us to perform further analysis through the algorithm's average performance ranking. The Wilcoxon sign-rank test is used as a statistical significance test, where a p-value lower than 0.05 reveals that HGS performs significantly better than its competitors.

3.2.1. Comparison with basic optimizers

In this part, HGS was compared with 15 other methods that can be categorized into two classes: well-established methods and recent methods. The recent methods include Sine Cosine Algorithm (SCA) (Mirjalili, 2016), Salp Swarm Algorithm (SSA) (Mirjalili et al., 2017), Grey Wolf Optimizer (GWO) (Mirjalili, Mirjalili, & Lewis, 2014), Moth-flame Optimization (MFO) (Mirjalili, 2015), Whale Optimization Algorithm (WOA) (Mirjalili & Lewis, 2016), Grasshopper Optimization Algorithm (GOA) (Saremi, Mirjalili, & Lewis, 2017), Dragonfly Algorithm (DA) (Mirjalili, 2015b), Ant Lion Optimizer (ALO) (Mirjalili, 2015a), and Multi-Verse Optimizer (MVO) (Mirjalili, Mirjalili, & Hatamlou, 2015). The well-established methods include Biogeography-based Optimization (BBO) (Simon, 2008), Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), Differential Evolution (DE) (Storn & Price, 1997), Firefly Algorithm (FA) (Yang, 2009), Bat Algorithm (BA) (Yang, 2010), and Flower Pollination Algorithm (FPA) (Yang, Karamanoglu, & Xingshi, 2014). For complete descriptions of those methods, please refer to the original research. The parameter settings of these algorithms are shown in Table 2.

The data in Table 3 represent the results of comparing HGS with other traditional MAs, where "+", "−" and "=" indicate that HGS performs better, worse, and equal to the corresponding algorithm, respectively. Avg, which is the average ranking result of the algorithm, is based on the Freidman test. From the table, we can intuitively find that HGS ranks first. For each opponent, it is difficult to defeat HGS on most 23 benchmark functions. Although DE defeats HGS in the largest number of functions, only five, and other algorithms do not even defeat HGS on anyone. The average value of our method is only 2.17, which is much smaller than other algorithms. Compared with the second-ranked DE, the average value of HGS is about half of DE. We can conclude that the performance of HGS is superior to the other counterparts.

Table A.1 in Appendix A shows the consequences of the Wilcoxon sign-rank test performed by HGS and other algorithms. Most p-values are less than 0.05, accounting for 93.0% of all data. Even in SCA, PSO, BA, and FPA, all p-values are less than 0.05. Although the numbers of p-values that are higher than 0.05 are the largest on DE and MFO, there are only five cases. This fact further shows that HGS has a strong statistical significance compared to the other methods.

Inspecting the results in Fig. 5 shows that the convergence rate of HGS is fast. From F1 to F4, we can see that HGS converges the fastest among all the algorithms, other algorithms converge quite slowly, and some even fall into local optimum. F5 and F9-F11 indicate that HGS has high accuracy in solving problems and can quickly find the global optimum at the beginning of the iteration. Although some algorithms' convergence speed is also very competitive in some stages, the accuracy of the solution of those methods is not as high as that of HGS, and the solution found by HGS has a higher quality. Based on the results of F7 and F8, the convergence speed of HGS slows down, but it still finds the global optimum first compared to the other algorithms. Some algorithms even fall into local optimum at the beginning of the iterations. Observing the performance algorithms on F22 and F23 functions, it can be concluded that HGS has a strong ability for global exploration. l can effectively switch between the two modes of starvation, and LH intuitively defines the minimum value of an individual's hunger. To prevent HGS from falling into local optima when faced with some multimodal landscapes, both of them directly affect hunger weights, which contribute to the improved rates of HGS in the iterative process and a better balance of the search and discovery stages. In the search phase, the solution space can be searched as complete as possible, so that the algorithm can achieve the effect of fast convergence at the early stage. In the mining stage, the optimal solution can be found nearby, which ensures the accuracy of the solution.

3.2.2. Comparison with improved metaheuristic methods

To further illustrate the effectiveness of the HGS algorithm, we compared HGS with ten state-of-the-art advanced algorithms: IWOA (Tubishat, Abushariah, Idris, & Aljarah, 2018), OBWOA (Elsayed abd el aziz and Oliva, 2018), ACWOA (Khashan, El-Hosseini, & Haikal, 2018), SCADE (Nenavath & Jatoth, 2018), CGSCA (Kumar, Hussain, Singh, & Panigrahi, 2017), m_SCA (Gupta & Deep, 2018), RCBA (Liang, Liu, Shen, Li, & Man, 2018), CBA (Adarsh, Raghunathan, Jayabarathi, & Yang, 2016), and CDLOBA (Yong, He, Li, & Zhou, 2018). For the full names of these methods and complete descriptions, please refer to the original works.

Based on the test data in Table 4, it can be recognized that HGS exhibits powerful performance on multimodal functions, especially on fixed dimension multimodal functions. The average value, based on Friedman test's value, is only 1.78, which is much smaller than other algorithms and is only about a fifth of the maximum average of CDLOBA. None of the five algorithms, IWOA, CGSCA, RCBA, CBA, or CDLOBA, can beat HGS on 23 benchmark functions. Although SCADE defeats HGS in dealing with some functions, there are only five cases. This observation clarifies that HGS has strong optimization ability in terms of exploration and exploitation trends. One of the effective mechanisms resulting in the improved solutions of the proposed HGS is that it is equipped with two rates, l and E. These features assist HGS in changing the initial positions and search mode, which ensure the in-depth exploration of the whole solution space as far as possible and enhance the exploratory traits of the algorithm to a great extent. Also, hunger weights can emphasize the perturbation trends of the HGS optimizer during iteration. This feature also reduces the change in stagnation due to the existence of several local optima.

Table A.2 in Appendix A reveals the p-value of HGS and its comparisons on all test functions. From the table, we can see that all values in CDLOBA are less than 0.05. The CGSCA, RCBA, and CBA have only one data value greater than 0.05 at most. Although OBWOA values greater than 0.05 are the most, there are only seven cases. We can also see from the table that the difference between values higher than 0.05 and 0.05 is not significant. These test results indicate that the HGS algorithm is significantly superior compared to the other algorithms.

The convergence curves of HGS are depicted in Fig. 6, which shows that the speed and accuracy of HGS are better than its competitors. On the F4 test function, the convergence rate of HGS is relatively constant,

Table 3
Comparison results of HGS algorithm on the 23 benchmark functions with traditional algorithms.

	F1		F2		F3	
Algorithm	AVG	STD	AVG	STD	AVG	STD
HGS	**5.10E-304**	0.00E + 00	6.00E-168	0.00E + 00	**1.20E-167**	0.00E + 00
SCA	7.19E-02	2.66E-01	2.65E-05	5.47E-05	3.76E + 03	3.61E + 03
SSA	1.15E-08	2.52E-09	7.03E-01	5.97E-01	2.73E + 02	1.96E + 02
GWO	3.85E-59	6.10E-59	1.20E-34	1.48E-34	2.00E-15	4.12E-15
MFO	2.67E + 03	4.50E + 03	3.57E + 01	1.98E + 01	2.28E + 04	1.41E + 04
WOA	8.30E-150	3.40E-149	8.10E-105	1.90E-104	1.92E + 04	1.15E + 04
GOA	7.90E + 00	5.18E + 00	8.00E + 00	1.05E + 01	2.05E + 03	9.88E + 02
DA	1.10E + 03	5.10E + 02	1.48E + 01	5.47E + 00	1.20E + 04	7.10E + 03
ALO	1.25E-05	8.89E-06	4.26E + 01	4.93E + 01	1.22E + 03	5.52E + 02
MVO	3.03E-01	8.98E-02	3.71E-01	1.03E-01	4.49E + 01	1.89E + 01
BBO	1.28E + 01	5.11E + 00	**0.00E + 00**	0.00E + 00	7.72E + 03	2.67E + 03
PSO	1.26E + 02	1.52E + 01	7.29E + 01	1.56E + 01	4.20E + 02	8.46E + 01
DE	2.86E-12	1.42E-12	3.71E-08	1.17E-08	2.46E + 04	4.20E + 03
FA	2.81E-03	7.78E-04	1.47E-01	7.82E-02	7.93E + 02	3.95E + 02
BA	1.48E + 01	2.02E + 00	2.53E + 03	1.37E + 04	6.70E + 01	1.40E + 01
FPA	2.41E + 02	7.87E + 01	1.56E + 01	3.68E + 00	3.67E + 02	1.27E + 02

	F4		F5		F6	
Algorithm	AVG	STD	AVG	STD	AVG	STD
HGS	**2.50E-137**	9.80E-137	**1.92E + 01**	9.77E + 01	7.78E-07	1.17E-06
SCA	1.95E + 01	1.08E + 01	2.80E + 02	5.07E + 02	4.63E + 00	4.82E-01
SSA	8.97E + 00	4.01E + 00	1.37E + 02	1.77E + 02	1.17E-08	3.42E-09
GWO	1.31E-14	1.59E-14	2.68E + 01	7.76E-01	6.34E-01	3.50E-01
MFO	6.86E + 01	7.66E + 00	6.47E + 03	2.27E + 04	1.66E + 03	4.59E + 03
WOA	4.40E + 01	2.89E + 01	2.71E + 01	4.67E-01	9.51E-02	1.14E-01
GOA	1.09E + 01	3.50E + 00	1.21E + 03	1.50E + 03	7.33E + 00	6.11E + 00
DA	2.42E + 01	7.91E + 00	1.25E + 05	1.02E + 05	1.20E + 00	5.20E + 02
ALO	1.18E + 01	3.13E + 00	1.34E + 02	2.22E + 02	8.28E-06	7.32E-06
MVO	9.40E-01	3.14E-01	3.98E + 02	7.26E + 02	3.07E-01	9.51E-02
BBO	3.83E + 01	1.25E + 01	4.34E + 02	2.46E + 02	1.70E + 01	8.78E + 00
PSO	4.47E + 00	1.95E-01	1.45E + 05	3.43E + 04	1.30E + 02	1.26E + 01
DE	1.92E + 00	3.59E-01	4.46E + 01	2.20E + 01	**2.62E-12**	1.36E-12
FA	6.62E-02	1.51E-02	2.02E + 02	3.94E + 02	2.81E-03	8.27E-04
BA	1.90E + 00	2.27E-01	4.05E + 03	1.29E + 03	1.47E + 01	1.84E + 00
FPA	1.59E + 01	3.40E + 00	7.52E + 03	4.57E + 03	2.22E + 02	8.13E + 01

	F7		F8		F9	
Algorithm	AVG	STD	AVG	STD	AVG	STD
HGS	**3.43E-04**	4.66E-04	**−1.26E + 04**	1.40E-01	**0.00E + 00**	0.00E + 00
SCA	3.95E-02	3.76E-02	−4.00E + 03	3.04E + 02	9.00E + 00	1.23E + 01
SSA	8.58E-02	2.84E-02	−7.58E + 03	8.00E + 02	6.22E + 01	2.09E + 01
GWO	8.84E-04	4.59E-04	−5.98E + 03	5.83E + 02	9.40E-01	3.04E + 00
MFO	2.12E + 00	4.15E + 00	−8.58E + 03	9.21E + 02	1.55E + 02	3.86E + 01
WOA	1.50E + 00	1.99E-03	−1.16E + 044	1.30E + 03	0.00E + 00	0.00E + 00
GOA	2.10E-02	8.21E-03	−7.49E + 03	7.50E + 02	9.61E + 01	3.46E + 01
DA	3.28E-01	1.74E-01	−5.61E + 03	6.62E + 02	1.48E + 02	3.10E + 02
ALO	1.01E-01	4.17E-02	−5.61E + 03	5.34E + 02	7.78E + 01	1.95E + 01
MVO	1.93E-02	8.42E-03	−7.59E + 03	5.93E + 02	1.08E + 02	2.81E + 01
BBO	7.76E-04	8.31E-04	−1.25E + 04	1.16E + 01	0.00E + 00	0.00E + 00
PSO	1.10E + 02	3.11E + 01	−7.09E + 03	8.76E + 02	3.66E + 02	2.13E + 01
DE	2.72E-02	5.77E-03	−1.25E + 04	9.61E + 01	5.94E + 01	6.45E + 00
FA	6.27E-03	2.27E-03	−6.98E + 03	5.10E + 02	4.01E + 01	1.16E + 01
BA	1.59E + 01	9.77E + 00	−7.11E + 03	6.34E + 02	2.63E + 02	2.47E + 01
FPA	1.47E-01	5.47E-02	−7.99E + 03	1.87E + 02	1.12E + 02	2.08E + 01

	F10		F11		F12	
Algorithm	AVG	STD	AVG	STD	AVG	STD
HGS	**8.88E-16**	0.00E + 00	**0.00E + 00**	0.00E + 00	1.07E-08	1.61E-08
SCA	1.47E + 01	8.94E + 00	3.79E-01	2.70E-01	5.98E + 01	3.08E + 02
SSA	1.99E + 00	5.25E-01	1.04E-02	9.53E-03	5.65E + 00	2.82E + 00
GWO	1.59E-14	2.90E-15	3.07E-03	7.78E-03	4.04E-02	2.10E-02
MFO	1.28E + 01	7.57E + 00	3.01E + 01	4.94E + 01	8.53E + 06	4.67E + 07
WOA	3.61E-15	2.22E-15	0.00E + 00	0.00E + 00	7.51E-03	8.42E-03
GOA	4.18E + 00	9.64E-01	8.04E-01	1.70E-01	6.05E + 00	2.59E + 00
DA	8.87E + 00	1.69E + 00	1.27E + 01	7.85E + 00	1.95E + 02	6.17E + 02
ALO	1.97E + 00	5.68E-01	1.55E-02	1.49E-02	9.83E + 00	4.73E + 00
MVO	1.10E + 00	7.29E-01	5.66E-01	1.35E-01	1.07E + 00	1.20E + 00
BBO	4.55E-01	4.17E-01	1.13E + 00	5.14E-02	4.52E-02	4.21E-02
PSO	8.28E + 00	4.02E-01	1.03E + 00	1.08E-02	4.86E + 00	7.09E-01
DE	4.48E-07	1.20E-07	3.82E-10	1.38E-09	**3.56E-13**	2.55E-13
FA	1.38E-02	2.44E-03	4.00E-03	2.36E-03	2.51E-05	1.17E-05

(*continued on next page*)

Table 3 (*continued*)

	F1		F2			F3	
Algorithm	AVG	STD	AVG	STD		AVG	STD
BA	4.36E + 00	1.90E-01	5.59E-01	6.27E-02		1.20E + 01	4.37E + 00
FPA	8.68E + 00	1.09E + 00	3.07E + 00	6.95E-01		4.36E + 00	1.03E + 00
	F13		F14			F15	
Algorithm	AVG	STD	AVG	STD		AVG	STD
HGS	9.91E-08	6.74E-08	1.97E + 00	2.98E + 00		6.80E-04	3.47E-04
SCA	1.01E + 01	3.09E + 01	1.66E + 00	9.49E-01		9.00E-04	3.72E-04
SSA	3.09E + 00	7.64E + 00	9.98E-01	2.31E-16		1.54E-03	3.56E-03
GWO	5.40E-01	2.47E-01	4.07E + 00	3.68E + 00		5.10E-03	8.57E-03
MFO	1.37E + 07	7.49E + 07	2.81E + 00	2.01E + 00		1.90E-03	3.76E-03
WOA	2.38E-01	2.08E-01	2.56E + 00	3.33E + 00		6.76E-04	3.49E-04
GOA	2.43E + 01	1.43E + 01	**9.98E-01**	4.03E-16		9.66E-03	1.26E-02
DA	3.38E + 04	6.34E + 04	9.98E-01	5.02E-10		2.59E-03	4.72E-03
ALO	8.36E-01	2.29E + 00	1.62E + 00	1.26E + 00		8.44E-04	2.07E-04
MVO	7.76E-02	5.78E-02	9.98E-01	9.70E-12		5.33E-03	8.44E-03
BBO	3.67E-01	2.06E-01	9.98E-01	5.65E-16		6.70E-02	1.54E-02
PSO	2.25E + 01	3.34E + 00	3.56E + 00	2.49E + 00		1.23E-03	4.14E-04
DE	**1.69E-12**	1.14E-12	1.16E + 00	9.00E + 00		6.91E-04	1.70E-04
FA	3.40E-04	1.80E-04	1.35E + 00	5.98E-01		9.26E-04	1.16E-03
BA	2.25E + 00	3.03E-01	4.43E + 00	4.03E + 00		8.79E-03	1.34E-02
FPA	2.56E + 01	5.57E + 00	9.98E-01	2.79E-08		**4.38E-04**	1.11E-04
	F16		F17			F18	
Algorithm	AVG	STD	AVG	STD		AVG	STD
HGS	**−1.03E + 00**	5.45E-16	**3.98E-01**	0.00E + 00		3.00E + 00	3.56E-15
SCA	−1.03E + 00	2.37E-05	3.98E-01	1.00E-03		3.00E + 00	3.93E-05
SSA	−1.03E + 00	7.61E-15	3.98E-01	3.81E-15		3.00E + 00	4.50E-14
GWO	−1.03E + 00	8.53E-09	3.98E-01	3.42E-07		3.00E + 00	1.03E-05
MFO	**−1.03E + 00**	6.78E-16	**3.98E-01**	0.00E + 00		3.00E + 00	2.14E-15
WOA	−1.03E + 00	1.44E-10	3.98E-01	8.35E-07		3.00E + 00	2.54E-05
GOA	−1.03E + 00	2.39E-14	3.98E-01	2.59E-14		5.70E + 00	1.48E + 01
DA	−1.03E + 00	1.88E-06	3.98E-01	9.99E-07		3.00E + 00	6.12E-05
ALO	−1.03E + 00	5.83E-14	3.98E-01	1.30E-14		3.00E + 00	4.22E-13
MVO	−1.03E + 00	9.74E-08	3.98E-01	2.59E-08		5.70E + 00	1.48E + 01
BBO	0.00E + 00	0.00E + 00	6.45E-01	0.00E + 00		3.00E + 00	0.00E + 00
PSO	−1.03E + 00	1.28E-03	3.98E-01	6.87E-04		3.09E + 00	7.39E-02
DE	**−1.03E + 00**	6.78E-16	**3.98E-01**	0.00E + 00		**3.00E + 00**	1.21E-15
FA	−1.03E + 00	1.31E-09	3.98E-01	6.21E-10		3.00E + 00	9.77E-09
BA	−1.03E + 00	3.85E-04	3.98E-01	1.66E-04		3.04E + 00	3.41E-02
FPA	−1.03E + 00	1.09E-11	3.98E-01	3.69E-15		3.00E + 00	3.40E-12
	F19		F20			F21	
Algorithm	AVG	STD	AVG	STD		AVG	STD
HGS	**−3.86E + 00**	2.45E-15	−3.25E + 00	6.68E-02		−9.98E + 00	9.31E-01
SCA	−3.86E + 00	2.34E-03	−2.80E + 00	4.67E-01		−2.95E + 00	1.86E + 00
SSA	−3.86E + 00	4.92E-14	−3.24E + 00	5.66E-02		−7.72E + 00	2.92E + 00
GWO	−3.86E + 00	2.41E-03	−3.23E + 00	1.09E-01		−9.48E + 00	1.75E + 00
MFO	**−3.86E + 00**	2.71E-15	−3.20E + 00	4.90E-02		−7.06E + 00	3.46E + 00
WOA	−3.86E + 00	3.19E-03	−3.24E + 00	1.17E-01		−9.29E + 00	1.93E + 00
GOA	−3.81E + 00	1.11E-01	−3.26E + 00	6.13E-02		−6.72E + 00	3.39E + 00
DA	−3.86E + 00	1.01E-04	−3.26E + 00	6.90E-02		−6.85E + 00	2.81E + 00
ALO	−3.86E + 00	1.57E-14	−3.27E + 00	6.03E-02		−7.30E + 00	3.21E + 00
MVO	−3.86E + 00	5.92E-07	−3.27E + 00	5.95E-02		−7.71E + 00	2.70E + 00
BBO	−3.35E-01	1.69E-16	−1.66E-01	2.82E-17		−6.49E + 00	3.57E + 00
PSO	−3.86E + 00	1.05E-02	−2.78E + 00	2.08E-01		−4.00E + 00	1.23E + 00
DE	**−3.86E + 00**	2.71E-15	**−3.32E + 00**	9.45E-05		−9.63E + 00	1.90E + 00
FA	−3.86E + 00	3.91E-10	−3.29E + 00	5.57E-02		−9.66E + 00	1.90E + 00
BA	−3.84E + 00	1.27E-02	−2.91E + 00	1.09E-01		−4.56E + 00	2.41E + 00
FPA	−3.86E + 00	2.09E-11	−3.32E + 00	6.03E-03		**−1.02E + 01**	6.87E-05
	F22		F23				
Algorithm	AVG	STD	AVG	STD	+/−/=	Avg	Rank
HGS	−1.04E + 01	1.07E-05	**−1.05E + 01**	3.94E-15	~	2.17	**1**
SCA	−3.87E + 00	1.95E + 00	−4.14E + 00	1.77E + 00	22/1/0	11.39	13
SSA	−8.38E + 00	3.21E + 00	−9.65E + 00	2.35E + 00	21/1/1	6.83	6
GWO	−1.02E + 01	9.70E-01	−1.05E + 01	2.83E-04	20/0/3	6.48	4
MFO	−8.30E + 00	3.27E + 00	−8.43E + 00	3.29E + 00	18/0/5	11.13	12
WOA	−9.08E + 00	2.46E + 00	−8.16E + 00	3.00E + 00	19/0/4	6.61	5
GOA	−5.84E + 00	3.39E + 00	−5.74E + 00	3.80E + 00	21/0/2	10.26	11
DA	−8.42E + 00	2.87E + 00	−8.17E + 00	3.21E + 00	21/1/1	11.91	14
ALO	−6.37E + 00	3.22E + 00	−7.14E + 00	3.33E + 00	21/1/1	8.34	9

(*continued on next page*)

Table 3 (*continued*)

	F1		F2			F3	
Algorithm	AVG	STD	AVG	STD		AVG	STD
MVO	−8.33E + 00	3.06E + 00	−9.19E + 00	2.52E + 00	21/1/1	7.91	7
BBO	−8.42E + 00	3.14E + 00	−9.23E + 00	2.71E + 00	20/2/1	9.22	10
PSO	−5.09E + 00	1.27E + 00	−4.61E + 00	1.29E + 00	23/0/0	12.96	16
DE	−1.02E + 01	9.72E-01	−1.05E + 01	5.41E-02	12/5/6	4.04	2
FA	**−1.04E + 01**	8.88E-07	−1.05E + 01	6.40E-07	19/2/2	5.74	3
BA	−6.45E + 00	2.76E + 00	−6.02E + 00	3.02E + 00	23/0/0	12.09	15
FPA	−1.04E + 01	2.54E-03	−1.05E + 01	1.70E-03	19/4/0	8.30	8

and the global optimal solution is found at a very fast speed during the entire process. At the same time that HGS finds the optimal solution, some algorithms have just started to converge to some solutions. From curves of F5 and F9-F11, it can be observed that HGS finds the optimal solution at a very fast speed during the initial iterations, but some of the compared algorithms that have fallen into local optimum. From these results, we can infer that the HGS has a strong ability of exploration and exploration propensities, and the two phases have excellent stability due to the impacts of *L* and *LH*. In the search phase, both of them expand the search scope as much as possible and ensure that the individual can search in a small range in the mining stage.

3.3. Validation of IEEE CEC2014 functions

To further illustrate the performance of the HGS algorithm, we tested it on the IEEE CEC2014 benchmark set. The data set is divided into Unimodal Functions, Simple Multimodal Functions, Hybrid Functions, and Composition Functions. Details of the functions can be found in Table 5. In this part, for the reliability of the experiment, the conditions related to the test were adjusted the same as before: the maximum number of iterations was set to 1000, the population size and dimension were set to 30, and the involved algorithm was tested 30 times randomly on each function. The Friedman test (Derrac et al., 2011) and Wilcoxon sign-rank test (Derrac et al., 2011) were utilized to evaluate the experimental results.

3.3.1. Comparison with other optimizers

The proposed HGS was compared with 12 traditional MAs on the IEEE CEC 2014 dataset, including SCA (Mirjalili, 2016), SSA (Mirjalili et al., 2017), GWO (Mirjalili et al., 2014), MFO (Mirjalili, 2015c), WOA (Mirjalili & Lewis, 2016), GOA (Saremi, et al., 2017), DA (Mirjalili, 2015b), ALO (Mirjalili, 2015a), PSO (Kennedy & Eberhart, 1995), DE (Ma et al., 2021; Sun, Li, & Deng, 2021), BA, and FPA (Yang, et al., 2014). The parameter settings of the mentioned algorithms are listed in Table 2.

The detailed comparison results are listed in Table 6. We found that HGS ranks first among all algorithms, with a much smaller Avg. HGS shows a strong ability to search for optimal solutions on most of the functions. It is well known that DE exhibits excellent performance on contest datasets, but it only defeats HGS on eight functions, while HGS defeats HGS on 19 functions. As a fixed-dimensional multi-modal function, composition functions have a large number of local optima, which requires an algorithm with excellent performance. HGS ranks first in the composite functions, including F23, F24, F25, F27, F28, and F30, which shows that the overall performance of HGS is powerful so that it can perform a smoother transition between exploration and exploration trends.

Table A.3 in Appendix A lists the p-value of HGS versus the other algorithms. Among the 360 data sets, 318 are less than 0.05, comprising 88.1% of the total data. It is worth noting that these data sets are far less than 0.05. Although there are more than 0.05 data in ALO, there are only 7 data sets. The number of it in SCA and DA is even reduced to only one. This shows that HGS has statistical advantages over the other competitive MAs.

According to the analysis of Fig. 7, we see that the convergence speed of HGS in F8, F10, and F11 is fast, and the accuracy of the solution is very high. Some algorithms even fall into the local optimum in the middle of the iteration. F23-F25, F27, F28, F30 are composite functions with a large number of local optima. Interestingly, we can observe that the convergence speed of the HGS algorithm is superior and fast on these types of problems. The target region can be found in the initial iteration period, which shows that the exploratory trends of HGS are influential and can effectively avoid falling into local optimum. These rates more intuitively show that HGS has the right sense of balance between exploration and exploration. Composite cases can challenge the capacity of utilized methods in harmonizing the main searching phases. The results show that HGS yields superior results and satisfactory performance. The reason for the satisfactory efficacy of HGS is the high capacity of this method in harmonizing the diversity of solutions and focusing on the locality of high-quality solutions in later phases. These two reasons are based on the *L* and *LH* parameters, which weigh the change of individual search range in the process of iteration. The HGS has a useful feature that ensures the search steps of HGS will be concentrated based on a specific rate. This feature assists this method in exploring the solution space in-depth, while it explores the feature space extensively during the initial stages.

3.3.2. Comparison with advanced MAs

To further prove the effectiveness of the proposed HGS, we further compared HGS with some state-of-the-art advanced algorithms on CEC2014 benchmark functions.

Table 7 shows the comparison between HGS and the advanced MAs on the CEC2014 test suite. As shown from the results, we can intuitively see that HGS ranks first amongst ten algorithms and first on 17 functions, accounting for 56.7% of the total number of functions, concentrating on simple multimodal functions and composition functions. From this point, we can see that HGS has excellent performance. The average value of HGS is only 2.37, which is about half of the average value of m_SCA, which ranks second. This indicates that the search ability of HGS is efficient, and it can avoid falling into local optimum.

Table A.4 in Appendix A lists the p-value of HGS versus the other involved MAs. Data sets less than 0.05 in **Table A.4** in Appendix A accounted for 91.1% of the total, revealing that HGS has distinct statistical advantages compared with its competitors. For IWOA, HGS has statistical significance on all functions.

Fig. 8 shows the convergence curves of the algorithms. At the beginning of the iteration, the convergence speed of HGS is very fast. With the increase of iteration times, the convergence speed slows down, but it is still the first one to find the optimal solution with high accuracy. F6, F8-12, and F16 show that HGS has a distinct advantage over simple multimodal functions. F29-30 reveals that HGS can find a better solution to composition functions with much faster convergence than the other counterparts. In the search phase, *l* and *LH* can dynamically expand the scope of individual search with the iteration to ensure that the algorithm can search the solution space as much as possible and can converge quickly. In the mining stage, after finding the possible region of the optimal solution, the search scope can be reduced to achieve the purpose of excavation and ensure the high-precision solution. The combination

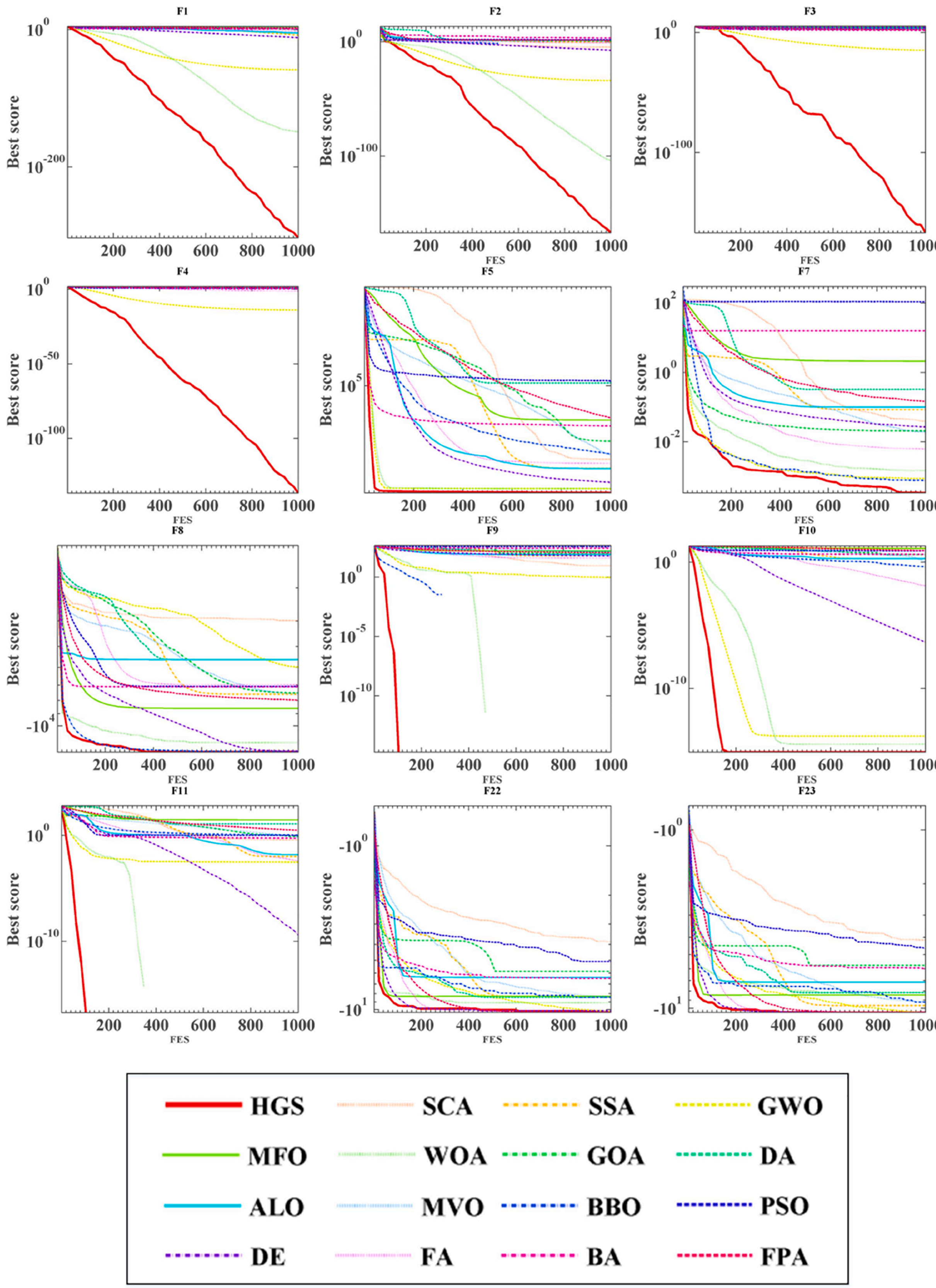

Fig. 5. Comparisons between HGS and traditional MAs.

Table 4
Comparison results on the 23 benchmark functions with advanced algorithms.

	F1		F2		F3	
Algorithm	AVG	STD	AVG	STD	AVG	STD
HGS	**0.00E + 00**	0.00E + 00	4.83E-166	0.00E + 00	1.54E-152	8.45E-152
IWOA	5.58E-150	2.2E-149	2.94E-102	9.1E-102	1.40E + 04	6.03E + 03
OBWOA	**0.00E + 00**	0.00E + 00	**6.25E-289**	0.00E + 00	1.49E + 04	1.09E + 04
ACWOA	6.87E-222	0.00E + 00	1.15E-113	5.69E-113	8.95E-177	0.00E + 00
SCADE	8.94E-217	0.00E + 00	3.58E-124	1.96E-123	**3.81E-183**	0.00E + 00
CGSCA	1.07E-57	5.85E-57	7.92E-33	4.32E-32	6.71E-44	3.67E-43
m_SCA	8.40E-48	3.48E-47	2.86E-31	1.54E-30	1.05E-11	5.76E-11
RCBA	1.62E-01	5.90E-02	9.94E + 00	2.74E + 01	9.54E + 01	3.12E + 01
CBA	2.90E-02	1.38E-01	1.49E + 02	6.65E + 02	7.90E + 01	4.02E + 01
CDLOBA	6.17E-03	1.90E-03	3.95E + 02	1.43E + 03	4.55E-01	1.35E + 00

	F4		F5		F6	
Algorithm	AVG	STD	AVG	STD	AVG	STD
HGS	**2.32E-132**	1.27E-131	**1.52E + 01**	1.18E + 01	**8.72E-07**	1.16E-06
IWOA	1.09E + 01	1.75E + 01	2.67E + 01	7.48E-01	2.30E-02	6.25E-02
OBWOA	3.24E + 01	1.90E + 01	2.72E + 01	7.07E-01	4.13E-01	2.57E-01
ACWOA	8.81E-99	3.96E-98	2.68E + 01	2.24E-01	2.72E-02	7.77E-03
SCADE	5.16E-38	1.84E-37	2.64E + 01	7.18E + 00	3.47E-06	2.65E-06
CGSCA	1.49E-24	8.18E-24	2.83E + 01	5.00E-01	4.85E + 00	2.43E-01
m_SCA	4.03E-14	1.29E-13	2.72E + 01	8.29E-01	2.48E + 00	5.24E-01
RCBA	1.01E + 01	6.13E + 00	3.11E + 02	4.86E + 02	1.76E-01	4.19E-02
CBA	1.85E + 01	7.79E + 00	2.98E + 02	4.86E + 02	1.07E-01	5.81E-01
CDLOBA	4.70E + 01	8.78E + 00	9.68E + 01	1.39E + 02	5.92E-03	1.67E-03

	F7		F8		F9	
Algorithm	AVG	STD	AVG	STD	AVG	STD
HGS	6.46E-04	9.46E-04	−1.25E + 04	6.05E + 02	**0.00E + 00**	0.00E + 00
IWOA	2.07E-03	2.36E-03	−1.08E + 04	1.76E + 03	**0.00E + 00**	0.00E + 00
OBWOA	1.53E-04	2.14E-04	−1.15E + 04	1.29E + 03	**0.00E + 00**	0.00E + 00
ACWOA	**7.06E-05**	5.92E-05	**−1.26E + 04**	4.09E + 01	**0.00E + 00**	0.00E + 00
SCADE	6.26E-04	3.29E-04	−1.24E + 04	2.89E + 02	**0.00E + 00**	0.00E + 00
CGSCA	5.28E-04	3.59E-04	−4.19E + 03	7.06E + 02	**0.00E + 00**	0.00E + 00
m_SCA	7.87E-04	6.22E-04	−6.13E + 03	8.70E + 02	**0.00E + 00**	0.00E + 00
RCBA	5.38E-01	1.92E-01	−7.42E + 03	7.53E + 02	8.23E + 01	2.18E + 01
CBA	3.03E-01	2.38E-01	−7.17E + 03	7.04E + 02	1.36E + 02	4.19E + 01
CDLOBA	2.82E + 01	3.50E + 01	−7.40E + 03	7.95E + 02	2.59E + 02	4.39E + 01

	F10		F11		F12	
Algorithm	AVG	STD	AVG	STD	AVG	STD
HGS	**8.88E-16**	**0.00E + 00**	**0.00E + 00**	0.00E + 00	**5.87E-09**	6.40E-09
IWOA	3.49E-15	2.27E-15	1.64E-03	6.28E-03	5.57E-03	1.07E-02
OBWOA	2.55E-15	2.03E-15	**0.00E + 00**	0.00E + 00	2.06E-02	1.06E-02
ACWOA	2.55E-15	1.80E-15	**0.00E + 00**	0.00E + 00	2.13E-03	1.12E-03
SCADE	**8.88E-16**	**0.00E + 00**	**0.00E + 00**	0.00E + 00	8.11E-08	1.09E-07
CGSCA	**8.88E-16**	**0.00E + 00**	**0.00E + 00**	0.00E + 00	5.40E-01	7.60E-02
m_SCA	3.44E + 00	7.60E + 00	4.93E-03	2.70E-02	1.68E-01	5.98E-02
RCBA	4.54E + 00	4.97E + 00	3.45E-02	1.75E-02	1.46E + 01	6.07E + 00
CBA	1.56E + 01	2.51E + 00	1.67E-01	1.33E-01	1.66E + 01	6.48E + 00
CDLOBA	1.95E + 01	9.51E-01	1.41E + 02	9.60E + 01	2.01E + 01	5.45E + 00

	F13		F14		F15	
Algorithm	AVG	STD	AVG	STD	AVG	STD
HGS	6.55E-03	3.59E-02	1.65E + 00	2.48E + 00	6.45E-04	2.24E-04
IWOA	1.53E-01	1.61E-01	2.34E + 00	2.95E + 00	6.30E-04	3.14E-04
OBWOA	7.36E-01	3.04E-01	3.29E + 00	3.51E + 00	6.90E-04	3.27E-04
ACWOA	5.65E-02	3.41E-02	2.01E + 00	2.49E + 00	3.35E-04	4.31E-05
SCADE	**1.05E-06**	7.27E-07	**9.98E-01**	**3.13E-08**	**3.15E-04**	2.84E-06
CGSCA	2.49E + 00	9.86E-02	1.66E + 00	9.51E-01	8.74E-04	4.41E-04
m_SCA	1.64E + 00	2.36E-01	1.39E + 00	8.07E-01	8.21E-04	4.20E-04
RCBA	9.86E-02	4.12E-02	5.88E + 00	4.90E + 00	7.26E-03	1.25E-02
CBA	4.63E + 01	1.67E + 01	3.32E + 00	3.48E + 00	4.85E-03	7.89E-03
CDLOBA	3.99E + 01	1.43E + 01	2.91E + 00	1.51E + 00	4.20E-03	7.36E-03

	F16		F17		F18	
Algorithm	AVG	STD	AVG	STD	AVG	STD
HGS	**−1.03E + 00**	5.13E-16	**3.98E-01**	0.00E + 00	**3.00E + 00**	2.16E-15
IWOA	−1.03E + 00	8.70E-11	3.98E-01	2.98E-07	3.00E + 00	3.01E-06
OBWOA	−1.03E + 00	6.51E-09	3.98E-01	1.11E-06	3.00E + 00	3.68E-05
ACWOA	−1.03E + 00	2.49E-04	4.01E-01	1.11E-02	3.00E + 00	1.29E-03
SCADE	−1.03E + 00	1.57E-05	3.98E-01	1.90E-04	3.00E + 00	1.26E-03

(continued on next page)

Table 4 (*continued*)

	F1		F2			F3	
Algorithm	AVG	STD	AVG	STD		AVG	STD
CGSCA	−1.03E + 00	4.16E-05	3.99E-01	1.78E-03		3.00E + 00	2.06E-05
m_SCA	−1.03E + 00	4.76E-08	3.98E-01	8.72E-07		3.00E + 00	5.70E-06
RCBA	−1.03E + 00	1.58E-06	3.98E-01	3.71E-07		3.00E + 00	7.90E-05
CBA	−1.03E + 00	1.92E-05	3.98E-01	7.20E-06		3.00E + 00	7.49E-04
CDLOBA	−1.03E + 00	2.48E-04	3.98E-01	8.03E-05		3.01E + 00	9.31E-03
	F19		F20			F21	
Algorithm	AVG	STD	AVG	STD		AVG	STD
HGS	**−3.86E + 00**	2.40E-15	−3.27E + 00	7.29E-02		**−1.02E + 01**	5.68E-15
IWOA	−3.86E + 00	2.42E-06	−3.26E + 00	6.38E-02		−9.14E + 00	2.35E + 00
OBWOA	−3.86E + 00	4.26E-06	**−3.28E + 00**	6.72E-02		−1.02E + 01	9.54E-05
ACWOA	−3.86E + 00	3.74E-03	−3.21E + 00	1.14E-01		−8.94E + 00	2.18E + 00
SCADE	−3.86E + 00	3.17E-03	−3.19E + 00	6.82E-02		−1.02E + 01	2.01E-04
CGSCA	−3.85E + 00	2.11E-03	−2.87E + 00	3.43E-01		−2.58E + 00	2.15E + 00
m_SCA	−3.86E + 00	2.29E-03	−3.21E + 00	8.39E-02		−9.13E + 00	2.05E + 00
RCBA	−3.86E + 00	4.30E-04	−3.27E + 00	6.11E-02		−8.22E + 00	2.86E + 00
CBA	−3.86E + 00	1.04E-03	−3.25E + 00	7.79E-02		−6.71E + 00	3.19E + 00
CDLOBA	−3.85E + 00	8.20E-03	−2.98E + 00	9.97E-02		−6.01E + 00	3.34E + 00
	F22		F23				
Algorithm	AVG	STD	AVG	STD	+/−/=	Avg	Rank
HGS	**−1.04E + 01**	1.19E-15	−1.02E + 01	1.37E + 00	~	1.78	**1**
IWOA	−8.65E + 00	2.77E + 00	−8.47E + 00	3.04E + 00	19/0/4	4.57	5
OBWOA	−1.04E + 01	9.23E-05	**−1.05E + 01**	9.49E-05	15/1/7	4.00	3
ACWOA	−9.85E + 00	1.61E + 00	−1.05E + 01	8.14E-02	16/3/4	4.09	4
SCADE	−1.04E + 01	1.58E-04	−1.05E + 01	1.64E-04	14/4/5	3.39	2
CGSCA	−3.78E + 00	2.07E + 00	−3.85E + 00	1.86E + 00	19/0/4	6.52	7
m_SCA	−9.52E + 00	2.00E + 00	−9.82E + 00	1.85E + 00	19/1/3	5.57	6
RCBA	−7.82E + 00	3.30E + 00	−8.49E + 00	3.25E + 00	22/0/1	7.00	8
CBA	−6.86E + 00	3.68E + 00	−6.63E + 00	4.01E + 00	21/0/2	7.87	9
CDLOBA	−6.76E + 00	3.48E + 00	−6.25E + 00	3.83E + 00	23/0/0	8.65	10

of these two phases can effectively balance the search and excavation phases.

3.4. Comparisons with DE variants

This chapter compares HGS with some improved versions of DE, including MPEDE (Wu, Mallipeddi, Suganthan, Wang, & Chen, 2015), SPS_L_SHADE_EIG (Guo, Tsai, Yang, & Hsu, 2015), LSHADE_cnEpSi (Awad, Ali, & Suganthan, 2017), SHADE (Tanabe & Fukunaga, 2014), SADE (Qin, Huang, & Suganthan, 2009), LSHADE (Tanabe & Fukunaga, 2014), JDE (Brest, Greiner, Boskovic, Mernik, & Zumer, 2006) and DE (Storn & Price, 1997) on 21 functions, which were selected from the first 13 of 23 benchmark functions and the last 8 composite functions of CEC2014 functions. All functions can be divided into three categories: single-mode (F1- F7), multimodal (F8-F13), and composite functions (F14-F21). In this experiment, the population size N was set to 30, the dimension of the optimization problem D was taken as 30, the maximum evaluation number *MaxFES* was taken as 300,000 times, and each algorithm was executed 30 times randomly.

Table 8 shows the comparison between HGS and the improved version of DE. The results show that the HGS algorithm ranks first among the ten algorithms and first among the 15 functions, accounting for 71.4% of the total number of functions. From this point, we can see that HGS exhibits excellent performance with an average value of only 2.33. These results indicate that the search ability of HGS is effective and can avoid falling into local optimum.

Based on the analysis in Fig. 9, we can observe that the convergence rate of HGS in F1, F2, and F11 is fast, the solution accuracy is very high, and the optimal solution is found in the early iteration stage. Through the convergence graphs of F10, F14, F15, F18, and F19, it can be found that although all the algorithms have fast convergence speed in the initial stage, the convergence accuracy is not as high as HGS. On F7 and F21, HGS has high convergence accuracy and can find the global optimum.

3.5. Parameter sensitivity analysis

In this chapter, we analyze the parameters involved in the algorithms: population size (N), the maximum number of iterations (T), parameter (l), moreover, and hunger threshold (LH). These parameters affect the convergence speed and accuracy of HGS. When testing$l$, we fixed LH to 100, and set l to start at 0.01, with a step of 0.01 between every two numbers, a total of 10 values. Similarly, when we analyzedLH, we initialized l to 0.08 and LH as 10, 100, 1000, and 1000. When testing l and LH, N and T were set to 30 and 1000, respectively, and remain unchanged. Each algorithm was tested 30 times. All experiments were conducted on 23 well-regarded benchmark functions.

The comparison results of the different values for parameter l are found in Table 9. From the table, we see that l has a significant influence on the performance of the algorithm. In the experiment, when l was 0.08, the performance is the best. Also, the maximum difference between the average values can reach 2.66. The average value of 0.01 is about 1.88 times that of 0.08.

Table 10 presents the comparison of different values of LH. Of the four values in this experiment, LHranked first when LH was 10000. Nevertheless, the influence of LH is less exaggerated than that of l.

From the above results, we can draw the following conclusions: L and LH's values have a certain impact on the search ability and solution accuracy of HGS. The balance between the two stages of exploration and exploration is closely related to these two parameters. Readers can set values for both variables according to specific conditions.

When testing the influence of N and T on HGS, we use F13 in 23 benchmark functions as the test examples. Note that N was set to 5, 10, 30, 50, 100, and 200, respectively, and T was initialized to 50, 100, 200, 500, 1000 and 2000. The test results can be visually observed in Fig. 10. The increase of $N$and T will improve the solution accuracy of HGS, but after reaching a certain level, this effect will become minimal. Given the long-time consumed when the value is too large, and the unsatisfactory experimental results are too small, the user can set it according to the

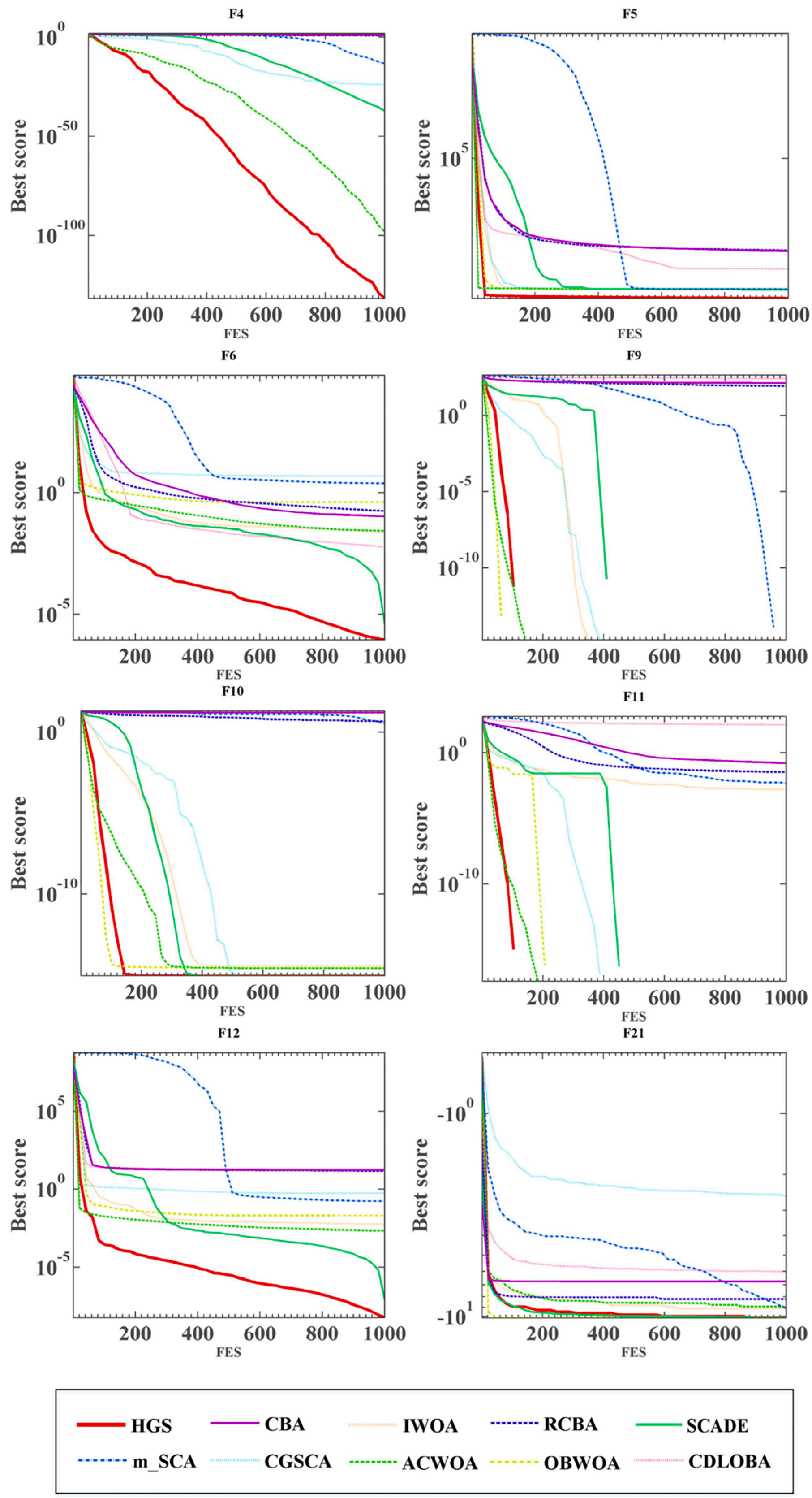

Fig. 6. Comparisons between HGS and advanced MAs.

Table 5
Description of the IEEE CEC2014 functions.

ID	Function Equation	Dim	Range	f_{min}
Unimodal Functions				
F1	RotatedHighConditionedEllipticFunction	30	[−100,100]	100
F2	RotatedBentCigarFunction	30	[−100,100]	200
F3	RotatedDiscusFunction	30	[-100,100]	300
Simple Multimodal Functions				
F4	ShiftedandRotatedRosenbrocksFunction	30	[−100,100]	400
F5	ShiftedandRotatedAckleysFunction	30	[−100,100]	500
F6	ShiftedandRotatedWeierstrassFunction	30	[−100,100]	600
F7	ShiftedandRotatedGriewanksFunction	30	[−100,100]	700
F8	ShiftedRastriginsFunction	30	[−100,100]	800
F9	ShiftedandRotatedRastriginsFunction	30	[−100,100]	900
F10	ShiftedSchwefelsFunction	30	[−100,100]	1000
F11	ShiftedandRotatedSchwefelsFunction	30	[−100,100]	1100
F12	ShiftedandRotatedKatsuuraFunction	30	[−100,100]	1200
F13	ShiftedandRotatedHappyCatFunction	30	[−100,100]	1300
F14	ShiftedandRotatedHGBatFunction	30	[−100,100]	1400
F15	ShiftedandRotatedExpandedGriewanksplusRosenbrocksFunction	30	[−100,100]	1500
F16	ShiftedandRotatedExpandedScaffersF6Function	30	[−100,100]	1600
Hybrid Functions				
F17	HybridFunction1(N = 3)	30	[−100,100]	1700
F18	HybridFunction2(N = 3)	30	[−100,100]	1800
F19	HybridFunction3(N = 4)	30	[−100,100]	1900
F20	HybridFunction4(N = 4)	30	[−100,100]	2000
F21	HybridFunction5(N = 5)	30	[−100,100]	2100
F22	HybridFunction6(N = 5)	30	[−100,100]	2200
Composition Functions				
F23	Compositionfunction1(N = 5)	30	[−100,100]	2300
F24	Compositionfunction2(N = 3)	30	[−100,100]	2400
F25	Compositionfunction3(N = 3)	30	[−100,100]	2500
F26	Compositionfunction4(N = 5)	30	[−100,100]	2600
F27	Compositionfunction5(N = 5)	30	[−100,100]	2700
F28	Compositionfunction6(N = 5)	30	[−100,100]	2800
F29	Compositionfunction7(N = 3)	30	[−100,100]	2900
F30	Compositionfunction8(N = 3)	30	[−100,100]	3000

experiment's actual needs.

3.6. Experiments on engineering design problems

It is well known that there are many constraints in practical problems (Liu, Guo, Lv, Qiao, & Azimi, 2020; Liu, Wang, Zhao, Su, & Chen, 2020; Peng, Chen, Zheng, & Liu, 2020; Peng, Zhang, Liu, Liu, & Qiao, 2020). In dealing with engineering scenarios, there is one main difference with global benchmark cases, and there is a concern on how to consider the restrictions and constraints of the variables and their impact on the minimization/maximization of the objective function (Liu et al., 2020; Ridha et al., 2021). Therefore, we further evaluated the efficiency of HGS by applying it to engineering problems. Several constraint handling methods were considered, including the death penalty, annealing, static, dynamic, co-evolutionary, and adaptive (Li et al., 2020). When searching individuals violate any constraints, the method assigns a large objective function value to them. In the optimization of the heuristic algorithm, this method will help to eliminate infeasible solutions automatically, so it is not necessary to calculate this scheme's infeasibility. The death penalty's most prominent advantages are simplicity and low time consumption (Li et al., 2020).

In this work, HGS was tested on four engineering constraints: welded beam, I-beam, and multiple disk clutch brake.

3.6.1. Welded beam design problem

The welded beam design problem aims to find the lowest consumption of welded beams under the four constraints of shear stress (τ), bending stress (θ), buckling load (P_c) and deflection (δ). The problem involves the following four variables: welding seam thickness (h); welding joint length (l); beam width (t); beam thickness (b). The mathematical model is as follows:

Consider $\vec{x} = [x_1, x_2, x_3, x_4] = [hltb]$

Minimize $f(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_4)$

Subject to $g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \le 0$

$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \le 0$$

$$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \le 0$$

$$g_4(\vec{x}) = x_1 - x_4 \le 0$$

$$g_5(\vec{x}) = P - P_C(\vec{x}) \le 0$$

$$g_6(\vec{x}) = 0.125 - x_1 \le 0$$

Variable range $0.1 \le x_1 \le 2$, $0.1 \le x_2 \le 10$, $0.1 \le x_3 \le 10$, $0.1 \le x_4 \le 2$

where $\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$ $\tau' = \frac{P}{\sqrt{2}x_1x_2}$ $\tau'' = \frac{MR}{J}$ $M = P(L + 0.5)$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$J = 2\left\{\frac{x_1x_2}{\sqrt{2}}\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

Table 6
Comparison results on the CEC2014 functions with traditional MAs.

	F1		F2		F3	
Algorithm	AVG	STD	AVG	STD	AVG	STD
HGS	9.56E + 06	7.25E + 06	4.53E + 06	1.46E + 07	1.62E + 04	1.57E + 04
SCA	4.24E + 08	1.39E + 08	2.57E + 10	4.62E + 09	5.95E + 04	1.06E + 04
SSA	2.39E + 07	1.41E + 07	9.64E + 03	9.52E + 03	7.31E + 04	2.22E + 04
GWO	9.07E + 07	5.57E + 07	3.16E + 09	2.60E + 09	4.55E + 04	1.07E + 04
MFO	8.69E + 07	9.20E + 07	1.41E + 10	1.08E + 10	1.09E + 05	6.24E + 04
WOA	1.31E + 08	3.88E + 07	2.26E + 09	1.15E + 09	9.91E + 04	4.99E + 04
GOA	3.29E + 07	1.45E + 07	1.87E + 07	1.11E + 07	6.58E + 04	2.69E + 04
DA	3.48E + 08	1.93E + 08	5.37E + 09	3.23E + 09	1.57E + 05	2.77E + 04
ALO	1.43E + 07	7.01E + 06	1.08E + 04	7.92E + 03	1.48E + 05	3.72E + 04
PSO	1.75E + 07	5.95E + 06	1.96E + 08	2.46E + 07	4.12E + 04	1.10E + 04
DE	1.01E + 08	3.01E + 07	**2.13E + 03**	4.05E + 03	**3.02E + 03**	2.46E + 03
BA	9.52E + 06	4.20E + 06	2.28E + 07	4.00E + 06	8.77E + 04	2.46E + 04
FPA	**7.38E + 06**	4.03E + 06	2.07E + 09	1.10E + 09	2.96E + 04	7.20E + 03

	F4		F5		F6	
Algorithm	AVG	STD	AVG	STD	AVG	STD
HGS	5.25E + 02	4.11E + 01	5.20E + 02	9.51E-02	6.20E + 02	3.26E + 00
SCA	2.42E + 03	7.11E + 02	5.21E + 02	5.65E-02	6.37E + 02	2.89E + 00
SSA	5.50E + 02	4.66E + 01	5.20E + 02	1.27E-01	6.26E + 02	3.47E + 00
GWO	7.02E + 02	1.55E + 02	5.21E + 02	5.54E-02	**6.16E + 02**	3.10E + 00
MFO	1.44E + 03	9.57E + 02	5.20E + 02	1.95E-01	6.24E + 02	3.55E + 00
WOA	8.72E + 02	1.56E + 02	5.21E + 02	1.25E-01	6.37E + 02	3.15E + 00
GOA	5.35E + 02	4.00E + 01	5.20E + 02	7.82E-02	6.22E + 02	3.67E + 00
DA	1.37E + 03	5.52E + 02	5.21E + 02	8.22E-02	6.37E + 02	3.50E + 00
ALO	5.46E + 02	3.55E + 01	**5.20E + 02**	1.52E-01	6.26E + 02	3.76E + 00
PSO	**4.82E + 02**	3.87E + 01	5.21E + 02	5.30E-02	6.25E + 02	2.98E + 00
DE	5.50E + 02	1.86E + 01	5.21E + 02	5.32E-02	6.29E + 02	1.27E + 00
BA	5.08E + 02	3.11E + 01	5.21E + 02	5.96E-02	6.36E + 02	3.06E + 00
FPA	6.79E + 02	7.50E + 01	5.21E + 02	5.08E-02	6.32E + 02	1.32E + 00

	F7		**F8**		**F9**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	7.01E + 02	2.34E + 00	**8.26E + 02**	1.01E + 01	1.04E + 03	3.16E + 01
SCA	9.33E + 02	3.28E + 01	1.07E + 03	2.43E + 01	1.20E + 03	2.61E + 01
SSA	**7.00E + 02**	1.37E-02	9.52E + 02	3.86E + 01	1.05E + 03	3.87E + 01
GWO	7.22E + 02	2.18E + 01	8.97E + 02	2.37E + 01	**1.01E + 03**	2.16E + 01
MFO	7.90E + 02	6.93E + 01	9.42E + 02	4.51E + 01	1.09E + 03	4.24E + 01
WOA	7.13E + 02	6.46E + 00	1.01E + 03	4.77E + 01	1.17E + 03	4.25E + 01
GOA	7.01E + 02	1.19E-01	9.51E + 02	4.05E + 01	1.07E + 03	4.85E + 01
DA	7.75E + 02	3.13E + 01	1.09E + 03	5.48E + 01	1.21E + 03	5.89E + 01
ALO	7.00E + 02	8.36E-03	9.23E + 02	3.29E + 01	1.05E + 03	4.32E + 01
PSO	7.03E + 02	2.09E-01	9.87E + 02	2.14E + 01	1.14E + 03	2.72E + 01
DE	7.00E + 02	1.82E-02	8.51E + 02	5.28E + 00	1.07E + 03	1.28E + 01
BA	7.01E + 02	3.01E-02	1.03E + 03	4.05E + 01	1.17E + 03	5.55E + 01
FPA	7.21E + 02	8.15E + 00	9.78E + 02	2.74E + 01	1.11E + 03	3.19E + 01

	F10		**F11**		**F12**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	**1.62E + 03**	3.00E + 02	**4.27E + 03**	5.46E + 02	**1.20E + 03**	1.09E-01
SCA	7.54E + 03	4.61E + 02	8.65E + 03	4.29E + 02	1.20E + 03	4.10E-01
SSA	4.91E + 03	7.59E + 02	5.21E + 03	8.92E + 02	1.20E + 03	3.93E-01
GWO	3.64E + 03	6.95E + 02	4.45E + 03	5.73E + 02	1.20E + 03	1.25E + 00
MFO	4.62E + 03	7.61E + 02	5.52E + 03	6.54E + 02	1.20E + 03	2.38E-01
WOA	5.79E + 03	7.30E + 02	6.95E + 03	8.14E + 02	1.20E + 03	5.47E-01
GOA	5.13E + 03	7.35E + 02	5.36E + 03	7.23E + 02	1.20E + 03	5.15E-01
DA	7.01E + 03	8.11E + 02	7.61E + 03	6.85E + 02	1.20E + 03	6.23E-01
ALO	4.55E + 03	6.54E + 02	5.43E + 03	7.00E + 02	1.20E + 03	3.24E-01
PSO	5.46E + 03	4.70E + 02	6.44E + 03	6.44E + 02	1.20E + 03	3.80E-01
DE	2.02E + 03	1.71E + 02	7.17E + 03	3.13E + 02	1.20E + 03	2.05E-01
BA	5.86E + 03	8.22E + 02	6.04E + 03	6.73E + 02	1.20E + 03	3.43E-01
FPA	4.49E + 03	2.32E + 02	5.54E + 03	3.38E + 02	1.20E + 03	2.09E-01

	F13		**F14**		**F15**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	1.30E + 03	1.46E-01	1.40E + 03	3.27E-01	1.52E + 03	4.76E + 00
SCA	1.30E + 03	3.85E-01	1.47E + 03	1.58E + 01	1.80E + 04	1.07E + 04
SSA	1.30E + 03	1.31E-01	1.40E + 03	1.89E-01	**1.51E + 03**	3.87E + 00
GWO	1.30E + 03	4.13E-01	1.41E + 03	1.02E + 01	1.92E + 03	6.96E + 02
MFO	1.30E + 03	1.43E + 00	1.43E + 03	2.11E + 01	1.22E + 05	2.74E + 05
WOA	1.30E + 03	1.00E-01	1.40E + 03	5.18E + 00	1.76E + 03	1.58E + 02

(*continued on next page*)

Table 6 (*continued*)

Algorithm	F1 AVG	F1 STD	F2 AVG	F2 STD	F3 AVG	F3 STD
GOA	1.30E + 03	1.47E-01	1.40E + 03	3.06E-01	1.52E + 03	4.78E + 00
DA	1.30E + 03	9.87E-01	1.42E + 03	1.04E + 01	8.86E + 03	1.08E + 04
ALO	**1.30E + 03**	8.46E-02	**1.40E + 03**	6.71E-02	1.51E + 03	4.22E + 00
PSO	1.30E + 03	1.01E-01	1.40E + 03	1.93E-01	1.52E + 03	1.89E + 00
DE	1.30E + 03	5.45E-02	1.40E + 03	7.80E-02	1.52E + 03	1.30E + 00
BA	1.30E + 03	1.28E-01	1.40E + 03	4.87E-02	1.53E + 03	3.53E + 00
FPA	1.30E + 03	7.47E-02	1.40E + 03	5.02E + 00	1.58E + 03	4.59E + 01

Algorithm	F16 AVG	F16 STD	F17 AVG	F17 STD	F18 AVG	F18 STD
HGS	**1.61E + 03**	5.62E-01	1.75E + 06	1.30E + 06	1.08E + 04	9.18E + 03
SCA	1.61E + 03	2.84E-01	1.59E + 07	7.68E + 06	2.78E + 08	1.17E + 08
SSA	1.61E + 03	7.18E-01	1.35E + 06	9.67E + 05	8.53E + 03	7.14E + 03
GWO	1.61E + 03	7.40E-01	4.02E + 06	4.60E + 06	1.82E + 07	2.64E + 07
MFO	1.61E + 03	5.11E-01	3.46E + 06	3.36E + 06	8.78E + 06	4.71E + 07
WOA	1.61E + 03	4.15E-01	1.51E + 07	9.17E + 06	5.03E + 05	4.83E + 05
GOA	1.61E + 03	4.80E-01	1.27E + 06	1.11E + 06	1.09E + 04	2.03E + 04
DA	1.61E + 03	3.02E-01	1.37E + 07	9.24E + 06	1.81E + 07	3.42E + 07
ALO	1.61E + 03	5.78E-01	1.44E + 06	9.30E + 05	**4.59E + 03**	2.46E + 03
PSO	1.61E + 03	4.09E-01	8.01E + 05	5.69E + 05	3.69E + 06	9.03E + 05
DE	1.61E + 03	2.38E-01	4.99E + 06	1.84E + 06	2.64E + 05	2.01E + 05
BA	1.61E + 03	2.48E-01	6.09E + 05	3.60E + 05	4.29E + 05	1.49E + 05
FPA	1.61E + 03	1.97E-01	**4.59E + 04**	3.75E + 04	7.70E + 03	3.84E + 03

Algorithm	F19 AVG	F19 STD	F20 AVG	F20 STD	F21 AVG	F21 STD
HGS	1.93E + 03	3.97E + 01	3.26E + 04	1.84E + 04	8.03E + 05	7.29E + 05
SCA	2.03E + 03	3.35E + 01	4.36E + 04	1.75E + 04	3.75E + 06	2.51E + 06
SSA	1.92E + 03	1.08E + 01	2.54E + 04	1.29E + 04	4.00E + 05	4.31E + 05
GWO	1.95E + 03	2.80E + 01	2.88E + 04	1.52E + 04	1.08E + 06	2.00E + 06
MFO	1.97E + 03	6.66E + 01	6.13E + 04	4.23E + 04	7.91E + 05	5.50E + 05
WOA	1.99E + 03	4.26E + 01	1.47E + 05	1.43E + 05	7.03E + 06	5.92E + 06
GOA	1.92E + 03	1.86E + 01	1.74E + 04	1.48E + 04	3.41E + 05	2.52E + 05
DA	1.99E + 03	5.67E + 01	1.50E + 05	1.41E + 05	4.21E + 06	6.33E + 06
ALO	1.92E + 03	2.26E + 01	4.20E + 04	2.13E + 04	4.23E + 05	3.63E + 05
PSO	1.92E + 03	2.49E + 00	1.78E + 04	8.48E + 03	2.72E + 05	1.84E + 05
DE	**1.91E + 03**	4.12E + 00	1.21E + 04	4.78E + 03	8.01E + 05	3.62E + 05
BA	1.92E + 03	1.59E + 01	2.58E + 04	1.35E + 04	2.38E + 05	1.58E + 05
FPA	1.92E + 03	9.13E + 00	**9.83E + 03**	4.28E + 03	**1.45E + 04**	3.40E + 03

Algorithm	F22 AVG	F22 STD	F23 AVG	F23 STD	F24 AVG	F24 STD
HGS	2.96E + 03	2.40E + 02	**2.50E + 03**	0.00E + 00	**2.60E + 03**	1.30E-03
SCA	3.27E + 03	1.67E + 02	2.71E + 03	2.26E + 01	2.61E + 03	1.90E + 01
SSA	2.81E + 03	1.91E + 02	2.63E + 03	7.71E + 00	2.64E + 03	6.13E + 00
GWO	2.68E + 03	1.91E + 02	2.64E + 03	9.93E + 00	2.60E + 03	1.08E-02
MFO	3.01E + 03	2.59E + 02	2.66E + 03	5.12E + 01	2.68E + 03	2.32E + 01
WOA	3.10E + 03	2.79E + 02	2.68E + 03	2.46E + 01	2.61E + 03	6.86E + 00
GOA	2.80E + 03	1.88E + 02	2.64E + 03	1.05E + 01	2.64E + 03	7.77E + 00
DA	3.37E + 03	2.85E + 02	2.72E + 03	3.51E + 01	2.67E + 03	1.04E + 01
ALO	3.03E + 03	2.77E + 02	2.63E + 03	7.06E + 00	2.66E + 03	1.05E + 01
PSO	2.97E + 03	2.20E + 02	2.62E + 03	1.50E + 00	2.63E + 03	4.71E + 00
DE	**2.64E + 03**	1.11E + 02	2.62E + 03	2.91E-03	2.63E + 03	2.20E + 00
BA	3.45E + 03	3.22E + 02	2.62E + 03	1.61E + 00	2.66E + 03	2.41E + 01
FPA	2.79E + 03	1.20E + 02	2.63E + 03	4.54E + 00	2.65E + 03	2.91E + 00

Algorithm	F25 AVG	F25 STD	F26 AVG	F26 STD	F27 AVG	F27 STD
HGS	**2.70E + 03**	0.00E + 00	2.74E + 03	4.94E + 01	**2.90E + 03**	0.00E + 00
SCA	2.74E + 03	1.13E + 01	2.70E + 03	4.01E-01	3.87E + 03	2.88E + 02
SSA	2.72E + 03	5.28E + 00	2.70E + 03	1.38E-01	3.57E + 03	1.50E + 02
GWO	2.71E + 03	5.50E + 00	2.74E + 03	4.93E + 01	3.37E + 03	1.49E + 02
MFO	2.72E + 03	8.26E + 00	2.70E + 03	1.18E + 00	3.60E + 03	2.22E + 02
WOA	2.72E + 03	1.81E + 01	2.73E + 03	6.52E + 01	3.84E + 03	3.69E + 02
GOA	2.71E + 03	3.99E + 00	2.78E + 03	5.52E + 01	3.44E + 03	2.36E + 02
DA	2.75E + 03	1.80E + 01	2.74E + 03	4.96E + 01	3.65E + 03	3.98E + 02
ALO	2.73E + 03	6.88E + 00	2.73E + 03	4.52E + 01	3.56E + 03	2.50E + 02
PSO	2.72E + 03	6.28E + 00	2.78E + 03	4.08E + 01	3.46E + 03	3.21E + 02
DE	2.72E + 03	3.45E + 00	2.70E + 03	6.78E-02	3.48E + 03	1.38E + 02
BA	2.73E + 03	1.28E + 01	2.71E + 03	5.60E + 01	4.00E + 03	3.15E + 02
FPA	2.71E + 03	2.74E + 00	**2.70E + 03**	2.46E-01	3.17E + 03	5.32E + 01

(*continued on next page*)

Table 6 (*continued*)

	F1		F2		F3	
Algorithm	AVG	STD	AVG	STD	AVG	STD
	F28		**F29**		**F30**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	**3.00E + 03**	0.00E + 00	8.55E + 05	2.59E + 06	**5.23E + 03**	2.63E + 03
SCA	5.65E + 03	4.49E + 02	3.25E + 07	1.51E + 07	5.59E + 05	2.18E + 05
SSA	4.28E + 03	4.16E + 02	4.81E + 06	7.30E + 06	3.71E + 04	2.10E + 04
GWO	4.06E + 03	3.53E + 02	2.05E + 06	4.36E + 06	7.75E + 04	4.45E + 04
MFO	3.95E + 03	2.04E + 02	3.54E + 06	4.04E + 06	5.66E + 04	5.40E + 04
WOA	5.59E + 03	6.37E + 02	1.35E + 07	9.72E + 06	2.91E + 05	2.41E + 05
GOA	4.39E + 03	4.04E + 02	3.30E + 06	1.02E + 07	5.36E + 04	2.94E + 04
DA	7.12E + 03	1.00E + 03	6.70E + 07	5.00E + 07	4.34E + 05	2.79E + 05
ALO	5.79E + 03	5.15E + 02	5.49E + 07	1.11E + 08	5.61E + 04	1.12E + 05
PSO	7.49E + 03	1.03E + 03	9.87E + 04	2.92E + 05	2.54E + 04	1.87E + 04
DE	3.72E + 03	2.88E + 01	**1.07E + 04**	1.37E + 04	1.13E + 04	2.17E + 03
BA	5.37E + 03	7.26E + 02	7.36E + 07	5.70E + 07	3.41E + 04	5.59E + 04
FPA	4.21E + 03	2.53E + 02	3.56E + 05	1.70E + 06	1.31E + 04	5.43E + 03

Algorithm	+/−/=	Avg	Rank
HGS	~	3.73	**1**
SCA	29/0/1	11.33	12
SSA	16/8/6	5.00	3
GWO	22/4/4	6.73	8
MFO	26/0/4	8.30	10
WOA	26/2/2	9.47	11
GOA	18/6/6	5.63	5
DA	29/0/1	11.60	13
ALO	17/6/7	5.67	6
PSO	21/6/3	6.40	7
DE	19/8/3	4.63	2
BA	20/6/4	7.50	9
FPA	20/7/3	5.00	3

$$\sigma(\overrightarrow{x}) = \frac{6PL}{x_4 {x_3}^2}, \delta(\overrightarrow{x}) = \frac{4PL^3}{Ex_3^3 x_4}$$

$$P_C(\overrightarrow{x}) = \frac{4.013E\sqrt{\frac{EGx_3^2x_4^6}{36}}}{6L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 60001b, L = 14 \in ..\delta_{max} = 0.25 \in ..$$

$$E = 30 \times 1^6 psi, G = 12 \times 10^6 psi$$

$$\tau_{max} = 13600psi, \sigma_{max} = 30000psi$$

On this subject, HGS was compared with HS (Lee & Geem, 2005) and CBO (Kaveh & Mahdavi, 2014). The results in Table 11 show that HGS has the best performance and solution. The main reason is the satisfactory performance of this method in harmonizing the diversity of solutions and later focusing on the locality of high-quality variable sets inside a constrained feature space.

3.6.2. *I-beam design problem*

The goal of this problem is to decrease the vertical deflection of the I-beams based on related parameters, including are length, height, and two thicknesses. The mathematical model of the problem is as follows:

Consider $\overrightarrow{x} = [x_1, x_2, x_3, x_4] = [bht_wt_f]$

Objective $f(\overrightarrow{x})_{min} = \frac{5000}{\frac{t_w(h-2t_f)^3}{12} + \frac{bt_f^3}{6} + 2bt_f\left(\frac{h-t_f}{2}\right)^2}$

Subject to $g(\overrightarrow{x}) = 2bt_w + t_w(h - 2t_f) \le 0$

Variable range $10 \le x_1 \le 50$

$$10 \le x_2 \le 80$$

$$0.9 \le x_3 \le 5$$

$$0.9 \le x_4 \le 5$$

Table 12 presents the comparisons between HGS and ARSM (Wang, 2003) IARSM (Wang, 2003), CS (Gandomi, Yang, & Alavi, 2013), and SOS (Cheng & Prayogo, 2014) on the I-beam problem. From the table, we see that HGS minimizes the vertical deflection of the I-beam more than the other four algorithms, demonstrating its superior efficacy for this engineering problem.

3.6.3. *Multiple disk clutch brake*

The objective of this minimization problem, categorized as a discrete optimization problem, is to use five discrete design variables to minimize the quality of multi-disc clutch brakes. The five variables are actuating force, inner and outer radius, number of 27 friction surfaces, and thickness of discs. The mathematical model for this problem is as follows:

$$f(x) = \Pi\left(r_0^2 - r_i^2\right)t(Z+1)\rho$$

subject to:

$$g_1(x) = r_0 - r_i - \Delta r \ge 0$$

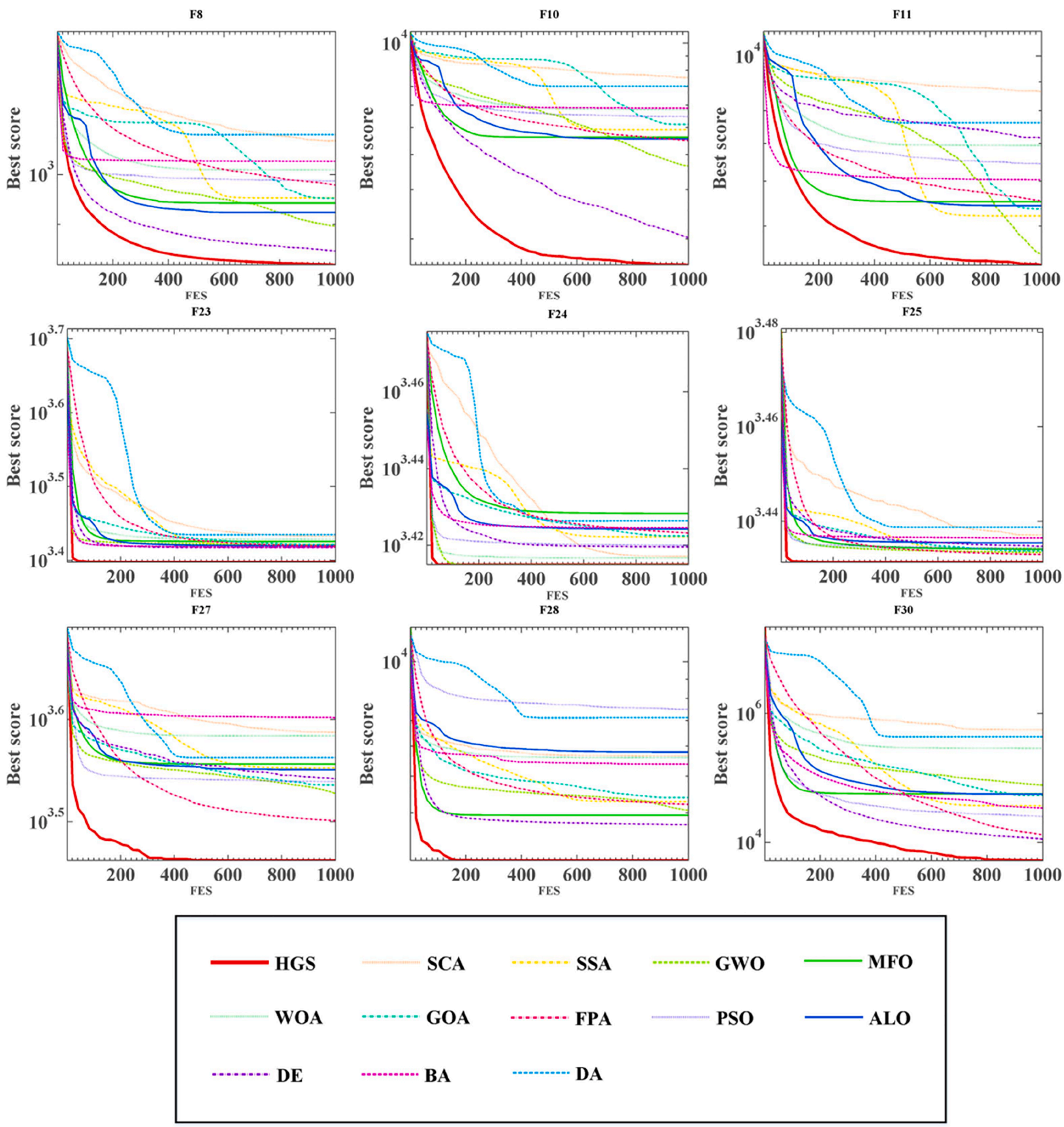

Fig. 7. Comparisons between HGS and traditional MAs.

Table 7
Comparison results on the CEC2014 functions with advanced algorithms.

	F1		F2		F3	
Algorithm	AVG	STD	AVG	STD	AVG	STD
HGS	1.13E + 07	8.82E + 06	7.84E + 07	4.28E + 08	**2.11E + 04**	1.95E + 04
IWOA	9.18E + 07	3.89E + 07	1.17E + 09	1.19E + 09	6.42E + 04	3.26E + 04
OBWOA	2.50E + 08	1.51E + 08	5.05E + 09	2.43E + 09	6.58E + 04	2.30E + 04
ACWOA	2.67E + 08	5.76E + 07	2.06E + 10	4.93E + 09	6.74E + 04	7.29E + 03
SCADE	5.71E + 08	1.01E + 08	3.90E + 10	5.66E + 09	6.55E + 04	5.78E + 03
CGSCA	4.73E + 08	1.28E + 08	3.00E + 10	4.90E + 09	6.11E + 04	6.93E + 03
m_SCA	1.25E + 08	7.83E + 07	1.27E + 10	4.72E + 09	4.16E + 04	8.98E + 03
RCBA	6.02E + 06	2.07E + 06	3.31E + 05	8.07E + 04	8.18E + 04	2.81E + 04
CBA	1.37E + 07	5.83E + 06	9.63E + 05	2.21E + 06	1.31E + 05	3.63E + 04
CDLOBA	**5.81E + 06**	4.51E + 06	**1.84E + 04**	1.10E + 04	1.27E + 05	3.24E + 04
	F4		**F5**		**F6**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	5.27E + 02	3.05E + 01	**5.20E + 02**	5.99E-02	**6.20E + 02**	2.42E + 00
IWOA	7.51E + 02	1.17E + 02	5.21E + 02	1.35E-01	6.36E + 02	3.42E + 00
OBWOA	1.30E + 03	6.16E + 02	5.21E + 02	7.76E-02	6.37E + 02	2.65E + 00
ACWOA	2.09E + 03	4.11E + 02	5.21E + 02	9.28E-02	6.36E + 02	2.51E + 00
SCADE	3.67E + 03	8.98E + 02	5.21E + 02	4.17E-02	6.36E + 02	2.70E + 00
CGSCA	2.79E + 03	6.10E + 02	5.21E + 02	7.53E-02	6.37E + 02	2.19E + 00
m_SCA	9.97E + 02	2.82E + 02	5.21E + 02	9.85E-02	6.25E + 02	2.73E + 00
RCBA	5.09E + 02	3.82E + 01	5.20E + 02	1.35E-01	6.39E + 02	3.58E + 00
CBA	5.10E + 02	3.09E + 01	5.20E + 02	2.53E-01	6.40E + 02	3.60E + 00
CDLOBA	**5.04E + 02**	3.90E + 01	5.21E + 02	8.92E-02	6.36E + 02	3.19E + 00
	F7		**F8**		**F9**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	7.02E + 02	3.95E + 00	**8.31E + 02**	1.65E + 01	**1.05E + 03**	2.66E + 01
IWOA	7.06E + 02	2.19E + 00	9.88E + 02	3.40E + 01	1.15E + 03	5.24E + 01
OBWOA	7.30E + 02	2.46E + 01	1.04E + 03	3.84E + 01	1.18E + 03	4.62E + 01
ACWOA	8.39E + 02	3.89E + 01	1.04E + 03	3.44E + 01	1.18E + 03	2.82E + 01
SCADE	9.73E + 02	4.14E + 01	1.09E + 03	1.30E + 01	1.22E + 03	2.04E + 01
CGSCA	9.67E + 02	4.72E + 01	1.09E + 03	1.73E + 01	1.21E + 03	1.87E + 01
m_SCA	7.88E + 02	3.66E + 01	9.72E + 02	2.50E + 01	1.08E + 03	2.98E + 01
RCBA	7.01E + 02	1.25E-01	1.02E + 03	6.47E + 01	1.17E + 03	5.66E + 01
CBA	7.00E + 02	1.65E-01	1.00E + 03	5.48E + 01	1.16E + 03	7.33E + 01
CDLOBA	**7.00E + 02**	1.21E-02	1.06E + 03	5.24E + 01	1.25E + 03	5.52E + 01
	F10		**F11**		**F12**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	**1.59E + 03**	2.57E + 02	**4.08E + 03**	5.73E + 02	**1.20E + 03**	8.38E-02
IWOA	4.27E + 03	6.70E + 02	6.37E + 03	6.45E + 02	1.20E + 03	5.27E-01
OBWOA	5.68E + 03	7.69E + 02	7.08E + 03	8.67E + 02	1.20E + 03	5.71E-01
ACWOA	6.00E + 03	6.50E + 02	7.47E + 03	9.30E + 02	1.20E + 03	6.24E-01
SCADE	7.73E + 03	2.71E + 02	8.62E + 03	3.42E + 02	1.20E + 03	3.17E-01
CGSCA	7.66E + 03	4.46E + 02	8.61E + 03	4.07E + 02	1.20E + 03	3.28E-01
m_SCA	5.22E + 03	7.60E + 02	6.02E + 03	5.98E + 02	1.20E + 03	5.33E-01
RCBA	5.48E + 03	6.67E + 02	5.62E + 03	6.81E + 02	1.20E + 03	3.96E-01
CBA	5.72E + 03	8.05E + 02	5.80E + 03	6.13E + 02	1.20E + 03	7.78E-01
CDLOBA	5.34E + 03	6.88E + 02	5.58E + 03	5.70E + 02	1.20E + 03	2.86E-01
	F13		**F14**		**F15**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	1.30E + 03	1.44E-01	1.40E + 03	3.37E-01	**1.51E + 03**	3.83E + 00
IWOA	1.30E + 03	1.15E-01	1.40E + 03	1.83E + 00	1.62E + 03	8.55E + 01
OBWOA	1.30E + 03	6.21E-01	1.42E + 03	1.10E + 01	5.86E + 03	7.68E + 03
ACWOA	1.30E + 03	4.47E-01	1.46E + 03	1.31E + 01	3.63E + 03	1.15E + 03
SCADE	1.30E + 03	4.19E-01	1.51E + 03	1.25E + 01	3.75E + 04	1.19E + 04
CGSCA	1.30E + 03	4.29E-01	1.48E + 03	1.53E + 01	2.66E + 04	1.73E + 04
m_SCA	1.30E + 03	9.94E-01	1.43E + 03	1.23E + 01	3.82E + 03	3.40E + 03
RCBA	1.30E + 03	1.54E-01	**1.40E + 03**	9.53E-02	1.54E + 03	8.65E + 00
CBA	1.30E + 03	1.56E-01	1.40E + 03	1.66E-01	1.57E + 03	1.72E + 01
CDLOBA	**1.30E + 03**	1.25E-01	1.40E + 03	1.10E-01	1.72E + 03	7.39E + 01
	F16		**F17**		**F18**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	**1.61E + 03**	8.07E-01	1.96E + 06	1.74E + 06	1.65E + 04	1.10E + 04
IWOA	1.61E + 03	5.60E-01	1.12E + 07	8.85E + 06	1.72E + 05	2.73E + 05
OBWOA	1.61E + 03	3.90E-01	1.72E + 07	1.58E + 07	9.68E + 06	1.84E + 07
ACWOA	1.61E + 03	5.28E-01	4.77E + 07	2.06E + 07	2.05E + 08	1.06E + 08
SCADE	1.61E + 03	1.92E-01	2.41E + 07	1.06E + 07	3.15E + 08	2.12E + 08

(continued on next page)

Table 7 (*continued*)

	F1		F2		F3	
Algorithm	AVG	STD	AVG	STD	AVG	STD
CGSCA	1.61E + 03	1.82E-01	1.44E + 07	5.09E + 06	3.13E + 08	2.18E + 08
m_SCA	1.61E + 03	5.20E-01	3.59E + 06	2.74E + 06	4.10E + 07	4.92E + 07
RCBA	1.61E + 03	3.65E-01	6.30E + 05	3.67E + 05	**9.35E + 03**	1.23E + 04
CBA	1.61E + 03	5.39E-01	9.21E + 05	5.88E + 05	2.42E + 04	4.54E + 04
CDLOBA	1.61E + 03	3.26E-01	**2.27E + 05**	2.14E + 05	1.54E + 04	7.00E + 03

	F19		**F20**		**F21**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	**1.93E + 03**	3.94E + 01	2.80E + 04	1.96E + 04	6.44E + 05	5.18E + 05
IWOA	1.97E + 03	3.81E + 01	6.33E + 04	4.90E + 04	3.82E + 06	3.79E + 06
OBWOA	2.02E + 03	6.08E + 01	7.78E + 04	3.55E + 04	6.30E + 06	5.52E + 06
ACWOA	2.06E + 03	5.10E + 01	1.01E + 05	4.48E + 04	1.54E + 07	1.25E + 07
SCADE	2.05E + 03	2.51E + 01	5.07E + 04	2.44E + 04	4.67E + 06	2.50E + 06
CGSCA	2.02E + 03	2.35E + 01	5.90E + 04	2.70E + 04	4.15E + 06	1.99E + 06
m_SCA	1.97E + 03	3.32E + 01	**2.20E + 04**	7.70E + 03	6.38E + 05	6.70E + 05
RCBA	1.94E + 03	3.64E + 01	2.91E + 04	1.48E + 04	3.85E + 05	3.23E + 05
CBA	1.94E + 03	3.54E + 01	4.92E + 04	2.75E + 04	4.24E + 05	3.66E + 05
CDLOBA	1.98E + 03	4.23E + 01	4.91E + 04	2.37E + 04	**1.59E + 05**	1.42E + 05

	F22		**F23**		**F24**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	2.93E + 03	2.63E + 02	**2.50E + 03**	0.00E + 00	2.60E + 03	1.18E-03
IWOA	3.09E + 03	2.60E + 02	2.64E + 03	5.02E + 01	2.62E + 03	4.04E + 01
OBWOA	3.17E + 03	2.91E + 02	2.69E + 03	2.09E + 01	2.60E + 03	4.12E + 00
ACWOA	3.37E + 03	3.34E + 02	2.53E + 03	7.94E + 01	2.60E + 03	7.79E-05
SCADE	3.33E + 03	1.62E + 02	**2.50E + 03**	0.00E + 00	**2.60E + 03**	9.05E-06
CGSCA	3.37E + 03	1.82E + 02	**2.50E + 03**	0.00E + 00	2.60E + 03	3.13E-04
m_SCA	**2.71E + 03**	1.78E + 02	2.65E + 03	1.46E + 01	2.60E + 03	5.70E-03
RCBA	3.52E + 03	3.02E + 02	2.62E + 03	1.83E + 00	2.69E + 03	3.41E + 01
CBA	3.52E + 03	3.12E + 02	2.62E + 03	1.87E + 00	2.68E + 03	3.02E + 01
CDLOBA	3.39E + 03	2.42E + 02	2.62E + 03	5.52E + 00	2.72E + 03	4.80E + 01

	F25		**F26**		**F27**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	**2.70E + 03**	0.00E + 00	2.74E + 03	4.94E + 01	**2.90E + 03**	0.00E + 00
IWOA	2.72E + 03	1.52E + 01	2.72E + 03	3.77E + 01	3.84E + 03	2.89E + 02
OBWOA	2.71E + 03	1.25E + 01	2.72E + 03	3.77E + 01	3.46E + 03	2.97E + 02
ACWOA	**2.70E + 03**	0.00E + 00	2.76E + 03	4.84E + 01	3.89E + 03	2.80E + 02
SCADE	**2.70E + 03**	0.00E + 00	2.70E + 03	6.94E-01	3.43E + 03	3.83E + 02
CGSCA	**2.70E + 03**	0.00E + 00	2.70E + 03	4.96E-01	**2.90E + 03**	0.00E + 00
m_SCA	2.72E + 03	5.39E + 00	**2.70E + 03**	6.57E-01	3.28E + 03	2.36E + 02
RCBA	2.74E + 03	1.64E + 01	2.73E + 03	4.65E + 01	4.04E + 03	3.85E + 02
CBA	2.73E + 03	1.41E + 01	2.73E + 03	6.76E + 01	3.99E + 03	4.51E + 02
CDLOBA	2.73E + 03	1.21E + 01	2.74E + 03	9.25E + 01	3.87E + 03	3.99E + 02

	F28		**F29**		**F30**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	**3.00E + 03**	0.00E + 00	**2.86E + 05**	1.53E + 06	**5.43E + 03**	2.92E + 03
IWOA	5.18E + 03	6.98E + 02	1.08E + 07	7.09E + 06	1.19E + 05	1.00E + 05
OBWOA	5.44E + 03	1.21E + 03	1.22E + 07	8.07E + 06	3.69E + 05	2.17E + 05
ACWOA	4.11E + 03	1.51E + 03	4.27E + 07	3.78E + 07	1.05E + 06	7.13E + 05
SCADE	5.83E + 03	2.63E + 02	2.94E + 07	1.24E + 07	7.63E + 05	1.94E + 05
CGSCA	**3.00E + 03**	0.00E + 00	3.40E + 06	7.18E + 06	2.78E + 05	2.61E + 05
m_SCA	4.27E + 03	3.10E + 02	7.16E + 06	9.73E + 06	1.09E + 05	6.47E + 04
RCBA	5.70E + 03	9.81E + 02	3.62E + 07	5.22E + 07	4.18E + 04	1.01E + 05
CBA	5.78E + 03	9.81E + 02	6.70E + 07	5.33E + 07	6.34E + 04	9.45E + 04
CDLOBA	5.53E + 03	8.42E + 02	2.26E + 07	2.60E + 07	6.61E + 04	1.08E + 05

Algorithm	+/−/=	Avg	Rank
HGS	~	2.37	**1**
IWOA	27/3/0	4.9	4
OBWOA	28/1/1	6.47	7
ACWOA	27/1/2	7.20	9
SCADE	26/0/4	7.53	10
CGSCA	23/1/6	6.57	8
m_SCA	26/2/2	4.47	2
RCBA	19/8/3	4.80	3
CBA	19/5/6	5.40	6
CDLOBA	20/8/2	4.93	5

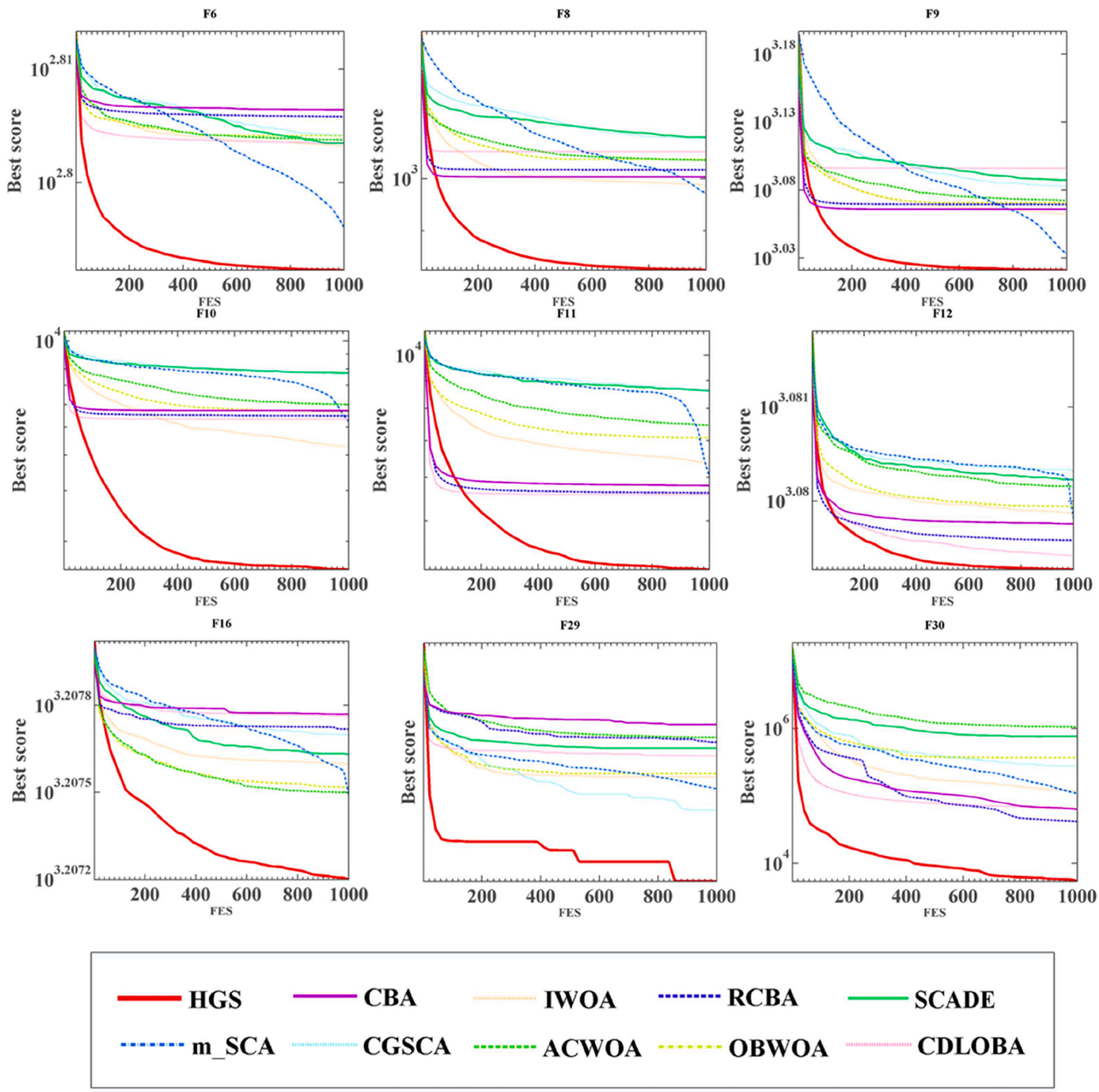

Fig. 8. Comparisons between HGS and advanced MAs.

Table 8
Comparison results with traditional DE variants.

	F1		F2		F3	
Algorithm	AVG	STD	AVG	STD	AVG	STD
HGS	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
MPEDE	1.86E-226	0.00E + 00	1.43E-111	6.59E-111	1.08E-35	4.20E-35
SPS_L_SHADE_EIG	2.86E-241	0.00E + 00	5.81E-123	2.19E-122	1.84E-38	8.22E-38
LSHADE_cnEpSi	5.09E-197	0.00E + 00	5.56E-69	3.04E-68	2.92E-49	1.32E-48
SHADE	2.08E-224	0.00E + 00	4.20E-95	2.02E-94	4.83E-52	2.62E-51
SADE	3.10E-151	1.68E-150	7.73E-104	4.23E-103	4.33E-06	8.07E-06
LSHADE	1.79E-206	0.00E + 00	2.48E-86	9.48E-86	8.06E-45	2.10E-44
JDE	2.66E-197	0.00E + 00	1.78E-123	3.83E-123	2.47E-12	4.73E-12
DE	1.41E-159	2.59E-159	1.53E-94	1.68E-94	1.49E + 03	8.15E + 02
	F4		**F5**		**F6**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	**0.00E + 00**	**0.00E + 00**	1.11E + 01	8.39E + 00	6.06E-10	8.67E-10
MPEDE	3.48E-05	1.02E-04	9.30E-01	1.71E + 00	3.60E-33	4.51E-33
SPS_L_SHADE_EIG	6.13E-09	1.17E-08	6.64E-01	1.51E + 00	**0.00E + 00**	**0.00E + 00**
LSHADE_cnEpSi	3.18E-04	9.16E-04	9.30E-01	1.71E + 00	1.26E-32	1.73E-32
SHADE	9.30E-18	3.76E-17	7.97E-01	1.62E + 00	4.11E-34	1.07E-33
SADE	1.19E-05	3.28E-05	2.19E + 01	2.44E + 01	1.95E-33	3.29E-33
LSHADE	1.33E-04	1.95E-04	9.30E-01	1.71E + 00	2.05E-33	3.74E-33
JDE	1.78E + 01	4.95E + 00	**5.32E-01**	**1.38E + 00**	2.05E-33	3.06E-33
DE	3.47E-14	1.46E-13	3.27E + 01	2.07E + 01	**0.00E + 00**	**0.00E + 00**
	F7		**F8**		**F9**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	**1.44E-05**	**2.50E-05**	−1.26E + 04	1.25E-03	**0.00E + 00**	**0.00E + 00**
MPEDE	2.72E-03	1.34E-03	−1.18E + 04	3.62E + 02	8.23E + 00	5.45E + 00
SPS_L_SHADE_EIG	3.27E-03	1.56E-03	−1.25E + 04	8.07E + 01	3.32E-02	1.82E-01
LSHADE_cnEpSi	9.89E-03	8.74E-03	−1.28E + 04	2.91E + 02	3.65E-01	8.05E-01
SHADE	2.50E-03	1.29E-03	−1.22E + 04	1.52E + 02	6.63E-14	3.02E-14
SADE	4.31E-03	2.08E-03	−1.26E + 04	3.00E + 01	1.09E + 00	9.90E-01
LSHADE	6.99E-03	3.96E-03	**−1.89E + 03**	**4.55E + 01**	3.58E + 00	1.00E + 01
JDE	2.08E-03	8.28E-04	−1.25E + 04	1.11E + 02	6.63E-02	2.52E-01
DE	2.44E-03	5.34E-04	−1.25E + 04	1.21E + 02	6.63E-02	2.52E-01
	F10		**F11**		**F12**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	8.88E-16	0.00E + 00	**0.00E + 00**	**0.00E + 00**	2.50E-14	5.92E-14
MPEDE	1.76E + 00	9.89E-01	1.50E-02	1.97E-02	1.94E-01	3.32E-01
SPS_L_SHADE_EIG	7.76E-15	9.01E-16	9.04E-04	2.78E-03	1.57E-32	5.57E-48
LSHADE_cnEpSi	3.37E + 00	8.40E-01	1.88E-02	2.00E-02	1.31E-01	3.16E-01
SHADE	2.77E-01	4.73E-01	6.40E-03	1.03E-02	3.80E-02	1.07E-01
SADE	**1.01E + 00**	**8.08E-01**	1.91E-02	2.80E-02	1.73E-02	7.74E-02
LSHADE	3.37E-14	3.58E-15	1.31E-02	1.91E-02	1.61E + 00	2.17E + 00
JDE	7.88E-15	3.43E-15	8.23E-03	1.99E-02	3.46E-03	1.89E-02
DE	7.64E-15	1.08E-15	0.00E + 00	0.00E + 00	**1.57E-32**	**5.57E-48**
	F13		**F14**		**F15**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	1.36E-12	4.00E-12	**2.50E + 03**	**0.00E + 00**	**2.60E + 03**	**0.00E + 00**
MPEDE	2.09E-01	7.85E-01	2.62E + 03	1.21E-12	2.64E + 03	8.00E + 00
SPS_L_SHADE_EIG	**1.35E-32**	**5.57E-48**	2.62E + 03	1.48E-12	2.63E + 03	1.20E + 00
LSHADE_cnEpSi	1.45E + 00	3.58E + 00	2.61E + 03	1.28E-01	2.64E + 03	5.43E + 00
SHADE	1.80E-03	4.94E-03	2.62E + 03	2.09E-12	2.64E + 03	6.70E + 00
SADE	3.66E-04	2.01E-03	2.62E + 03	1.34E-12	2.63E + 03	4.53E + 00
LSHADE	2.93E-03	4.94E-03	2.62E + 03	1.98E-12	2.64E + 03	6.58E + 00
JDE	1.55E-32	4.86E-33	2.62E + 03	1.52E-12	2.63E + 03	5.92E + 00
DE	**1.35E-32**	**5.57E-48**	2.62E + 03	1.39E-12	2.63E + 03	3.09E + 00
	F16		**F17**		**F18**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	**2.70E + 03**	**0.00E + 00**	2.76E + 03	5.00E + 01	**2.90E + 03**	**0.00E + 00**
MPEDE	2.71E + 03	4.94E + 00	2.72E + 03	4.29E + 01	3.27E + 03	1.38E + 02
SPS_L_SHADE_EIG	**2.70E + 03**	**2.56E + 00**	**2.71E + 03**	**3.45E + 01**	3.11E + 03	4.25E + 01
LSHADE_cnEpSi	2.71E + 03	3.34E + 00	2.73E + 03	4.48E + 01	3.30E + 03	1.19E + 02
SHADE	2.71E + 03	2.96E + 00	**2.71E + 03**	**3.44E + 01**	3.15E + 03	7.33E + 01
SADE	2.71E + 03	1.80E + 00	2.73E + 03	4.78E + 01	3.19E + 03	7.07E + 01
LSHADE	2.71E + 03	2.99E + 00	**2.71E + 03**	**3.04E + 01**	3.26E + 03	9.41E + 01
JDE	2.71E + 03	1.94E + 00	2.70E + 03	1.82E + 01	3.11E + 03	5.81E + 01
DE	2.71E + 03	8.27E-01	2.70E + 03	4.72E-02	3.22E + 03	7.34E + 01

(*continued on next page*)

Table 8 (*continued*)

	F1		F2		F3	
Algorithm	AVG	STD	AVG	STD	AVG	STD
	F19		**F20**		**F21**	
Algorithm	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
HGS	**3.00E + 03**	**0.00E + 00**	**3.10E + 03**	**0.00E + 00**	**3.20E + 03**	**0.00E + 00**
MPEDE	3.86E + 03	1.89E + 02	5.67E + 05	3.09E + 06	5.52E + 03	1.09E + 03
SPS_L_SHADE_EIG	3.71E + 03	1.11E + 02	6.40E + 05	2.44E + 06	4.77E + 03	1.07E + 03
LSHADE_cnEpSi	3.97E + 03	3.03E + 02	1.30E + 06	3.99E + 06	6.23E + 03	1.59E + 03
SHADE	3.71E + 03	1.18E + 02	7.07E + 05	2.71E + 06	5.73E + 03	9.44E + 02
SADE	3.72E + 03	4.13E + 01	4.06E + 03	2.21E + 02	5.10E + 03	5.14E + 02
LSHADE	3.71E + 03	1.09E + 02	3.71E + 03	1.48E + 02	5.38E + 03	1.06E + 03
JDE	3.66E + 03	4.52E + 01	3.67E + 03	3.72E + 01	5.27E + 03	9.90E + 02
DE	3.63E + 03	2.16E + 01	6.59E + 03	1.01E + 04	6.56E + 03	1.32E + 03

Algorithm	+/−/=	Avg	Rank
HGS	~	**2.33**	**1**
MPEDE	16/3/2	6.76	8
SPS_L_SHADE_EIG	13/5/3	3.05	2
LSHADE_cnEpSi	16/4/1	7.10	9
SHADE	17/3/1	4.57	5
SADE	18/2/1	5.95	6
LSHADE	17/3/1	6.00	7
JDE	16/4/1	3.95	3
DE	15/4/2	4.29	4

$$g_2(x) = l_{max} - (Z+1)(t+\delta) \geq 0$$

$$g_3(x) = P_{max} - P_{rz} \geq 0$$

$$g_4(x) = P_{max}v_{sr\,max} - P_{rz}v_{sr} \geq 0$$

$$g_5(x) = v_{sr\,max} - v_{sr} \geq 0$$

$$g_6 = T_{max} - T \geq 0$$

$$g_7(x) = M_h - sM_s \geq 0$$

$$g_8(x) = T \geq 0$$

$$M_h = \frac{2}{3}\mu FZ\frac{r_0^3 - r_i^2}{r_0^2 - r_i^3}\ P_{rz} = \frac{F}{\Pi\left(r_0^2 - r_i^2\right)}\ v_{rz} = \frac{2\Pi n\left(r_0^3 - r_i^3\right)}{90\left(r_0^2 - r_i^2\right)}\ T = \frac{I_z\Pi n}{30\left(M_h + M_f\right)}$$

$\Delta r = 20mm$ $I_z = 55kgmm^2$ $P_{max} = 1MPa$ $F_{max} = 1000N$ $T_{max} = 15s$ $\mu = 0.5$ $s = 1.5$ $M_s = 40Nm$ $M_f = 3Nm$ $n = 250rpm$ $v_{sr\,max} = 10m/s$ $l_{max} = 30mm$ $r_{imin} = 60$ $r_{imax} = 80$ $r_{0min} = 90$ $r_{0max} = 110$ $t_{min} = 1.5$ $t_{max} = 3$ $F_{min} = 600$ $F_{max} = 1000$ $Z_{min} = 2$ $Z_{max} = 9$

In order to minimize the quality of multi-disc clutch brakes, HGS was compared with WCA (Eskandar, Sadollah, Bahreininejad, & Hamdi, 2012), PVS (Rao, Savsani, & Vakharia, 2011), and TLBO (Savsani & Savsani, 2016). Details of the comparison results can be found in Table 13. From the table, we can see that the quality of HGS is nearly equal to or better than other algorithms, which shows that HGS has better vital optimization ability and can find more high-quality solutions.

4. Conclusions and future perspectives

This study presents a novel population-based model to tackle optimization problems based on social animals' characteristics in searching for food. More specifically, in each iteration, the algorithm searches around the optimal location, in the same manner, that animals forage, where the weights, or hunger values, mimic the impact of hunger on an animal's individual activity. Qualitative analysis of the algorithm was carried out using four indicators, including search history, the trajectory of the first dimension, average fitness, and convergence curve. The proposed Hunger Games Search (HGS) was validated on a comprehensive collection of 23 benchmark functions and IEEE CEC2014 functions. The Wilcoxon sign rank test and the Freidman test were utilized to assess the statistical significance between HGS and other well-known MAs. The experimental results show that HGS has an efficient searching ability compared with other algorithms, and it can quickly find and develop the target solution space. Overall, HGS is very good at balancing exploration and exploration. Simultaneously, to confirm the applicability of HGS to practical problems, four engineering problems were considered, including welded beam, I-beam, and multiple disk clutch brake. From the experimental results, HGS can satisfy the optimization effect of production engineering problems and significantly reduce manufacturing costs.

In this paper, we followed the most straightforward rules for developing HGS to make it easier to expand and integrate with existing artificial intelligence methods. There are many windows for the future directions of this new efficient HGS algorithm. First, researchers can investigate the effectiveness of this open-source HGS code for solving real-world problems in parameter optimization for machine learning models, binary feature selection, and image segmentation. Another window is how to enhance the performance of the basic version proposed in this research. Possible chances and future proposals are the application of oppositional based learning (OBL), Orthogonal learning (OL), chaotic signals instead of random variables, applying evolutionary population dynamic (EPD), advantages of mutation and crossover on the exploration and exploitation cores of the method, ensemble mutation-based strategies, role of levy flight, application of greedy search, co-evolutionary methods, quantum computing, parallel computing, ranking-based schemes, random spare schemes, multi-population structures, dimension-wise operations, and their various combination[7]. Lastly, the proposed Hunger Games Search is a single-objective approach in the currently released version, and the next task can be to develop the binary, multiobjective and many-objective variants of the developed Hunger Games Search to deal with more variety of multi-objective, binary, and many-objective problems.

[7] Literature and guidance on most of these mechanisms are publicly available at https://aliasgharheidari.com/Publications.html and https://www.researchgate.net/profile/Ali_Asghar_Heidari

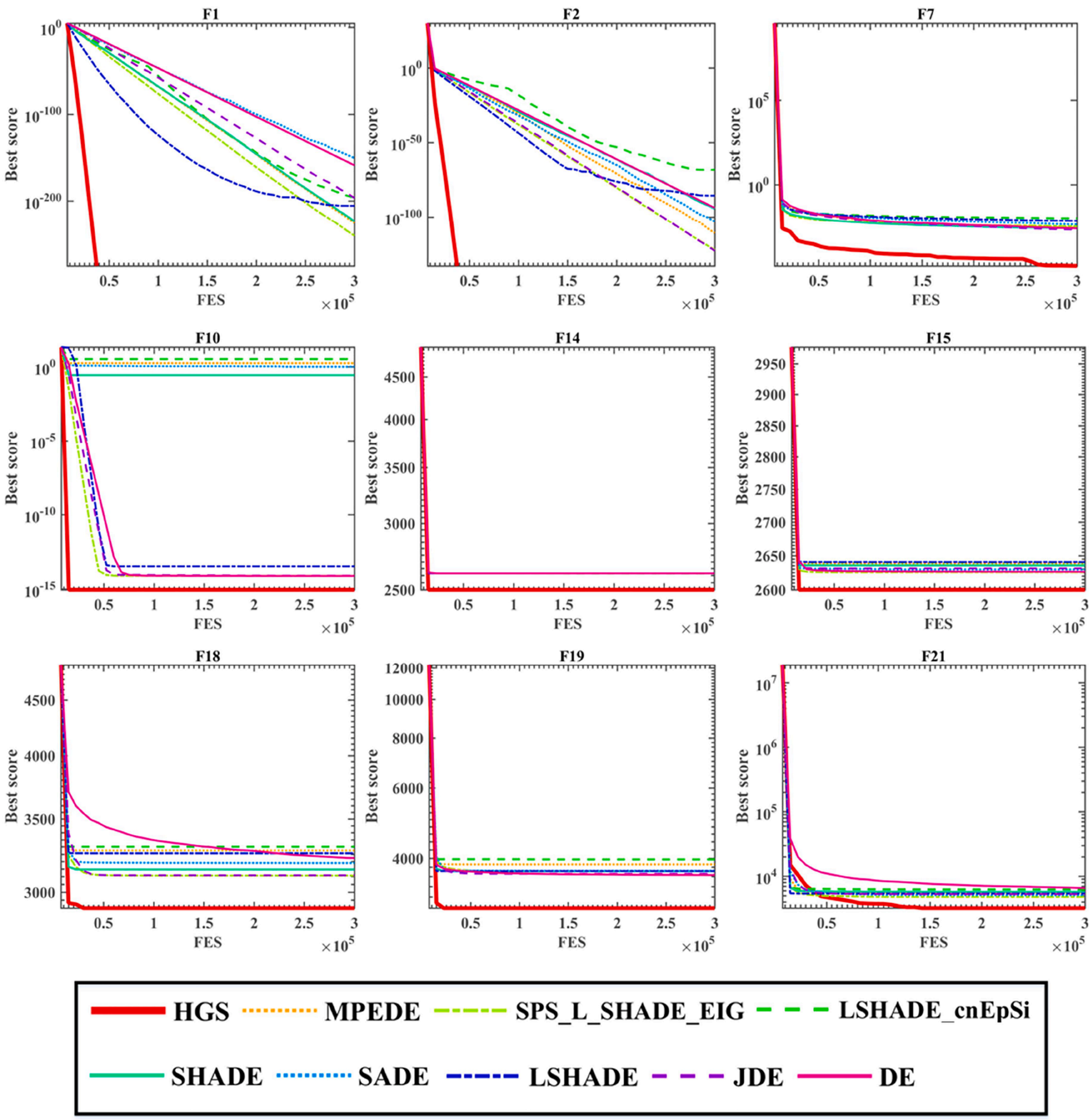

Fig. 9. Comparisons between HGS and DE variants.

Table 9
Ranking of results with different values of parameter.l

Fun	$l = \mathbf{0.01}$	$l = \mathbf{0.02}$	$l = \mathbf{0.03}$	$l = \mathbf{0.04}$	$l = \mathbf{0.05}$	$l = \mathbf{0.06}$	$l = \mathbf{0.07}$	$l = \mathbf{0.08}$	$l = \mathbf{0.09}$	$l = \mathbf{0.1}$
ARV	5.70	4.96	4.96	3.26	3.13	3.91	3.2	3.04	3.17	4.43
Rank	10	8	8	5	2	6	4	1	3	7

Table 10
Ranking of results with different values of parameter.LH

Fun	$LH = 10$	$LH = 100$	$LH = 1000$	$LH = 10000$
ARV	2.17	2.04	2.06	2.00
Rank	4	2	3	1

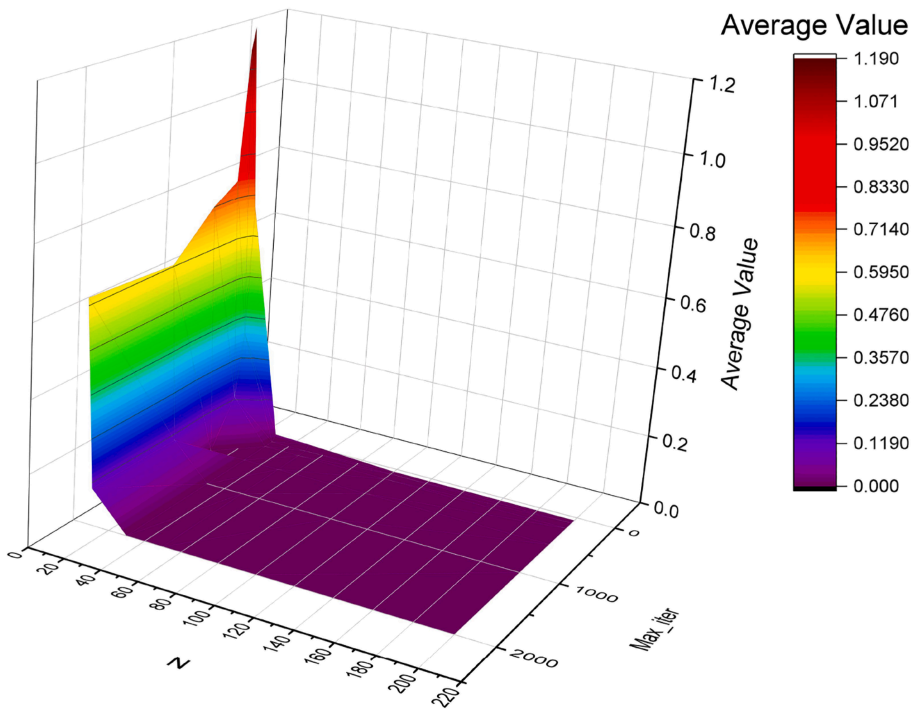

Fig. 10. The influence of N and T (it is shown by *Max_iter* in the above plot).

Table 11
Results of welded beam design problem compared with other methods.

Algorithm	Optimal values for variables				Optimum cost
	h	l	t	b	
HGS	0.26	5.1025	8.03961	0.26	2.302076
HS(Lee & Geem, 2005)	0.2442	6.2231	8.2915	0.2433	2.3807
CBO (Kaveh & Mahdavi, 2014)	0.2434	6.2552	8.2915	0.2444	2.38411

Table 12
Results of I-beam design problem compared with other methods.

Algorithm	Optimal values for variables				Optimum cost
	B	h			
HGS	50	80	0.9	2.321792	0.013074
ARSM(G. G. Wang, 2003)	48.42	79.99	0.90	2.40	0.0157
IARSM(G. G. Wang, 2003)	48.4200	79.9900	0.9000	2.4000	0.1310
CS(Gandomi et al., 2013)	50	80	0.9	2.321675	0.0130747
SOS(Cheng & Prayogo, 2014)	50	80	0.9	2.32179	0.0130741

Table 13
Results of Multiple disk clutch brake compared with other methods.

Algorithm			t	F	Z	Optimal cost
HGS	70	90	**1**	1000	3	0.313657
WCA(Eskandar et al., 2012)	70	90	1	910	3	0.313656
PVS(Rao et al., 2011)	70	90	1	980	3	0.31366
TLBO(Savsani & Savsani, 2016)	70	90	1	810	3	0.313656

CRediT authorship contribution statement

Yutao Yang: Conceptualization, Validation, Methodology, Formal analysis, Resources, Writing - Review & Editing, Writing - original draft, Software, Visualization, Investigation. **Huiling Chen:** Conceptualization, Resources, Project administration, Validation, Methodology, Formal analysis, Investigation, Funding acquisition, Supervision. **Ali Asghar Heidari:** Methodology, Conceptualization, Formal analysis, Validation, Data Curation, Resources, Writing - Review & Editing, Writing - original draft, Software, Visualization, Investigation. **Amir H Gandomi:** Writing - Review & Editing.

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.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (62076185, U1809209). We acknowledge the comments the anonymous reviewers and respected editor that enhanced the quality of this research. We also acknowledge Seyedali Mirjalili for editing the first version of the manuscript.

Appendix A

Tables A.1-A.4 Describing the corresponding p-values of the four experiments

Table A.1 The p-value of the Wilcoxon test obtained from comparison with traditional algorithms on 23 benchmark functions

Function	SCA	SSA	GWO	MFO	WOA	GOA	DA	ALO
F1	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F2	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F3	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F4	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F5	1.73E-06	1.92E-06	1.73E-06	1.92E-06	1.73E-06	1.73E-06	1.73E-06	2.88E-06
F6	1.73E-06	1.92E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	8.47E-06
F7	1.73E-06	1.73E-06	1.36E-04	1.73E-06	3.32E-04	1.73E-06	1.73E-06	1.73E-06
F8	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F9	1.73E-06	1.73E-06	3.91E-03	1.73E-06	1.00E + 00	1.73E-06	1.73E-06	1.73E-06
F10	1.73E-06	1.73E-06	5.98E-07	1.73E-06	2.30E-05	1.73E-06	1.73E-06	1.73E-06
F11	1.73E-06	1.73E-06	6.25E-02	1.73E-06	1.00E + 00	1.73E-06	1.73E-06	1.73E-06
F12	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F13	1.73E-06	3.61E-03	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F14	2.77E-03	7.27E-01	7.71E-04	6.65E-02	1.20E-03	1.07E-01	2.06E-02	3.72E-02
F15	1.32E-02	2.30E-02	7.66E-01	6.42E-03	9.92E-01	5.71E-04	2.60E-05	2.70E-02
F16	1.73E-06	5.05E-06	1.73E-06	1.00E + 00	1.73E-06	2.55E-06	1.96E-04	2.52E-06
F17	1.73E-06	1.22E-04	1.73E-06	1.00E + 00	1.73E-06	1.71E-06	5.96E-05	5.76E-05
F18	1.73E-06	1.70E-06	1.73E-06	3.11E-01	1.73E-06	1.73E-06	3.15E-06	1.73E-06
F19	1.73E-06	1.80E-06	1.73E-06	1.00E + 00	1.73E-06	1.73E-06	1.73E-06	1.35E-06
F20	1.73E-06	2.56E-02	1.31E-01	4.80E-03	2.45E-01	1.99E-01	2.54E-01	3.71E-01
F21	1.73E-06	1.02E-05	3.11E-05	4.88E-04	1.97E-05	1.13E-05	1.73E-06	1.02E-05
F22	1.73E-06	1.36E-05	1.73E-06	3.91E-03	1.73E-06	1.73E-06	1.73E-06	5.22E-06
F23	1.73E-06	1.73E-06	1.73E-06	3.91E-03	1.73E-06	1.73E-06	1.73E-06	1.73E-06

Table A.1(continued) The p-value of Wilcoxon test obtained from comparison with traditional algorithms on 23 benchmark functions

Function	MVO	BBO	PSO	DE	FA	BA	FPA
F1	1.73E-06	1.69E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F2	1.73E-06	1.56E-02	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F3	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F4	1.73E-06	1.72E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F5	1.73E-06	1.92E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F6	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F7	1.73E-06	3.33E-02	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F8	1.73E-06	1.73E-06	1.73E-06	3.88E-06	1.73E-06	1.73E-06	1.73E-06
F9	1.73E-06	1.00E + 00	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F10	1.73E-06	1.52E-04	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F11	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F12	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F13	1.73E-06	1.92E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F14	2.77E-03	9.97E-04	2.77E-03	9.38E-02	2.77E-03	7.71E-04	2.77E-03
F15	4.39E-03	1.72E-06	4.73E-06	8.13E-01	5.58E-01	7.69E-06	2.26E-03
F16	1.73E-06	4.32E-08	1.73E-06	1.00E + 00	1.73E-06	1.73E-06	1.73E-06
F17	1.73E-06	4.32E-08	1.73E-06	1.00E + 00	1.73E-06	1.73E-06	3.13E-02
F18	1.73E-06	5.80E-07	1.73E-06	2.91E-03	1.73E-06	1.73E-06	1.73E-06
F19	1.73E-06	4.32E-08	1.73E-06	1.00E + 00	1.73E-06	1.73E-06	1.73E-06
F20	6.73E-01	9.51E-07	1.73E-06	2.48E-04	7.81E-01	1.73E-06	8.73E-03
F21	1.73E-06	7.36E-06	1.73E-06	3.22E-02	1.73E-06	1.73E-06	3.11E-05
F22	1.73E-06	6.03E-07	1.73E-06	1.95E-02	3.11E-05	1.73E-06	1.73E-06
F23	1.73E-06	3.34E-07	1.73E-06	1.00E + 00	1.73E-06	1.73E-06	1.73E-06

Table A.2 The p-value of Wilcoxon test obtained from comparison with advanced algorithms on 23 benchmark functions

Fun	IWOA	OBWOA	ACWOA	SCADE	CGSCA	m_SCA	RCBA	CBA	CDLOBA
F1	1.73E-06	1.00E + 00	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F2	1.73E-06	3.96E-01	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F3	1.73E-06	1.73E-06	3.39E-01	8.29E-01	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F4	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F5	1.73E-06	1.73E-06	1.73E-06	5.79E-05	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F6	1.73E-06	1.73E-06	1.73E-06	1.49E-05	1.73E-06	1.73E-06	1.73E-06	5.45E-02	1.73E-06
F7	2.58E-03	5.19E-02	1.71E-03	6.44E-01	7.19E-01	1.31E-01	1.73E-06	1.73E-06	1.73E-06
F8	1.13E-05	1.80E-05	2.45E-01	2.83E-04	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F9	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.73E-06	1.73E-06	1.73E-06
F10	3.81E-05	2.44E-04	1.22E-04	1.00E + 00	1.00E + 00	1.51E-06	1.73E-06	1.73E-06	1.73E-06
F11	5.00E-01	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.73E-06	1.73E-06	1.73E-06
F12	1.73E-06	1.73E-06	1.73E-06	1.92E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F13	1.24E-05	1.73E-06	2.84E-05	3.72E-05	1.73E-06	1.73E-06	2.16E-05	1.73E-06	1.73E-06
F14	2.22E-04	1.60E-04	2.61E-04	3.59E-04	3.59E-04	3.59E-04	1.97E-05	2.41E-04	3.59E-04
F15	6.58E-01	3.93E-01	1.24E-05	7.69E-06	4.72E-02	2.70E-02	1.59E-03	2.37E-05	3.52E-06

(*continued on next page*)

(*continued*)

Fun	IWOA	OBWOA	ACWOA	SCADE	CGSCA	m_SCA	RCBA	CBA	CDLOBA
F16	2.56E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F17	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F18	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F19	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F20	1.16E-01	2.71E-01	5.32E-03	3.59E-04	1.73E-06	2.11E-03	2.06E-01	1.92E-01	1.73E-06
F21	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F22	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F23	6.89E-05	3.59E-04	3.59E-04	3.59E-04	1.73E-06	2.83E-04	1.15E-04	8.47E-06	7.69E-06

Table A.3P-value of Wilcoxon test obtained from HGS versus other traditional algorithms on IEEE CEC2014 functions

Function	SCA	SSA	GWO	MFO	WOA	GOA
F1	1.73E-06	4.45E-05	1.73E-06	1.92E-06	1.73E-06	2.13E-06
F2	1.73E-06	3.41E-05	1.73E-06	1.73E-06	1.73E-06	2.22E-04
F3	3.88E-06	1.73E-06	1.80E-05	1.73E-06	1.92E-06	1.73E-06
F4	1.73E-06	2.85E-02	1.92E-06	2.13E-06	1.73E-06	5.17E-01
F5	1.73E-06	1.71E-01	1.73E-06	2.22E-04	1.73E-06	1.31E-01
F6	1.73E-06	3.52E-06	1.96E-03	1.60E-04	1.73E-06	1.85E-02
F7	1.73E-06	1.73E-06	2.60E-06	1.92E-06	1.92E-06	1.85E-02
F8	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F9	1.73E-06	5.17E-01	8.19E-05	1.36E-04	1.92E-06	2.43E-02
F10	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F11	1.73E-06	3.41E-05	1.47E-01	3.88E-06	1.73E-06	1.49E-05
F12	1.73E-06	5.75E-06	4.73E-06	3.72E-05	1.73E-06	2.35E-06
F13	1.73E-06	3.38E-03	2.77E-03	4.45E-05	2.60E-05	1.20E-03
F14	1.73E-06	1.02E-05	1.40E-02	3.52E-06	3.71E-01	1.24E-05
F15	1.73E-06	3.16E-02	2.35E-06	1.92E-06	1.73E-06	7.52E-02
F16	1.73E-06	1.06E-04	2.70E-02	2.35E-06	1.73E-06	6.98E-06
F17	1.73E-06	1.59E-01	4.39E-03	6.04E-03	1.73E-06	1.65E-01
F18	1.73E-06	1.92E-01	6.34E-06	2.29E-01	1.73E-06	1.78E-01
F19	1.92E-06	3.71E-01	3.00E-02	2.85E-02	2.60E-05	3.39E-01
F20	3.68E-02	6.87E-02	4.53E-01	8.94E-04	2.88E-06	2.96E-03
F21	4.73E-06	1.04E-02	4.78E-01	6.73E-01	5.22E-06	8.94E-04
F22	4.29E-06	1.40E-02	1.49E-05	4.65E-01	5.71E-02	9.84E-03
F23	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F24	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F25	1.73E-06	1.73E-06	1.73E-06	1.73E-06	2.93E-04	1.73E-06
F26	2.06E-01	2.13E-06	8.45E-01	1.71E-01	1.11E-02	2.05E-04
F27	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F28	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F29	1.73E-06	1.25E-04	3.06E-04	3.32E-04	1.73E-06	2.58E-03
F30	1.73E-06	1.73E-06	1.73E-06	2.35E-06	1.73E-06	1.73E-06

Table A.3(continued) The p-value of Wilcoxon test obtained from HGS versus other traditional algorithms on IEEE CEC2014 functions

Function	DA	ALO	PSO	DE	BA	FPA
F1	1.73E-06	1.40E-02	2.61E-04	1.73E-06	8.29E-01	3.39E-01
F2	1.73E-06	4.45E-05	1.73E-06	2.60E-06	6.32E-05	1.73E-06
F3	1.73E-06	1.92E-06	3.11E-05	5.75E-06	1.73E-06	1.60E-04
F4	1.73E-06	2.18E-02	1.04E-03	9.84E-03	1.11E-01	1.73E-06
F5	1.73E-06	1.06E-01	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F6	1.73E-06	1.24E-05	3.18E-06	1.73E-06	1.73E-06	1.73E-06
F7	1.73E-06	1.73E-06	2.96E-03	1.73E-06	1.75E-02	1.73E-06
F8	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F9	1.73E-06	4.53E-01	1.73E-06	9.63E-04	1.92E-06	3.18E-06
F10	1.73E-06	1.73E-06	1.73E-06	1.02E-05	1.73E-06	1.73E-06
F11	1.73E-06	8.47E-06	1.92E-06	1.73E-06	2.35E-06	1.73E-06
F12	1.73E-06	2.60E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F13	4.20E-04	2.35E-06	1.64E-05	3.52E-06	3.41E-05	1.92E-06
F14	1.92E-06	2.35E-06	5.22E-06	7.69E-06	3.88E-06	8.73E-03
F15	1.73E-06	4.05E-01	4.20E-04	2.85E-02	3.18E-06	1.73E-06
F16	1.73E-06	2.37E-05	1.73E-06	2.88E-06	1.73E-06	1.73E-06
F17	2.60E-06	2.71E-01	1.59E-03	4.29E-06	8.19E-05	1.73E-06
F18	1.73E-06	6.16E-04	1.73E-06	1.73E-06	1.73E-06	2.62E-01
F19	5.79E-05	1.71E-01	3.71E-01	1.20E-01	2.71E-01	3.71E-01
F20	4.29E-06	4.49E-02	1.38E-03	1.64E-05	5.71E-02	4.29E-06
F21	1.60E-04	1.66E-02	1.48E-04	3.09E-01	3.72E-05	1.73E-06
F22	5.31E-05	3.82E-01	9.75E-01	3.41E-05	1.97E-05	2.77E-03
F23	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F24	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F25	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F26	2.13E-01	1.92E-01	3.11E-05	1.92E-06	4.45E-05	8.47E-06

(*continued on next page*)

(*continued*)

Function	DA	ALO	PSO	DE	BA	FPA
F27	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F28	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F29	1.73E-06	7.43E-05	8.45E-01	5.04E-01	2.88E-06	2.26E-03
F30	1.73E-06	1.73E-06	7.69E-06	3.52E-06	1.73E-06	2.13E-06

Table A.4P-value of Wilcoxon test obtained from comparison with advanced algorithms on IEEE CEC2014 functions

Fun	IWOA	OBWOA	ACWOA	SCADE	CGSCA	m_SCA	RCBA	CBA	CDLOBA
F1	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.29E-03	5.19E-02	2.58E-03
F2	2.37E-05	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.48E-02	4.53E-01	2.07E-02
F3	5.22E-06	2.88E-06	2.35E-06	3.18E-06	2.88E-06	1.60E-04	2.13E-06	1.73E-06	1.92E-06
F4	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	3.16E-02	3.16E-02	1.75E-02
F5	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.92E-06	1.59E-03	1.73E-06
F6	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.92E-06	1.73E-06	1.73E-06	1.73E-06
F7	2.11E-03	2.13E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	6.58E-01	8.47E-06	1.73E-06
F8	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F9	1.92E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	3.88E-04	1.73E-06	2.88E-06	1.73E-06
F10	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F11	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	2.35E-06	3.18E-06	1.73E-06	2.60E-06
F12	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	5.75E-06
F13	1.80E-05	2.29E-01	1.73E-06	1.73E-06	1.73E-06	2.84E-05	3.06E-04	5.31E-05	8.47E-06
F14	1.38E-03	7.69E-06	1.73E-06	1.73E-06	1.73E-06	1.92E-06	2.88E-06	4.29E-06	1.02E-05
F15	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F16	3.52E-06	7.69E-06	1.36E-05	1.73E-06	1.73E-06	1.24E-05	1.73E-06	1.73E-06	1.73E-06
F17	1.36E-05	4.73E-06	1.73E-06	1.73E-06	1.73E-06	1.57E-02	1.15E-04	6.64E-04	3.18E-06
F18	4.29E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	2.11E-03	4.17E-01	6.29E-01
F19	8.94E-04	1.64E-05	2.88E-06	1.73E-06	2.35E-06	8.19E-05	3.00E-02	1.99E-01	1.59E-03
F20	8.94E-04	4.29E-06	4.29E-06	5.29E-04	6.89E-05	1.92E-01	6.14E-01	1.83E-03	4.90E-04
F21	7.69E-06	3.52E-06	1.73E-06	1.73E-06	1.92E-06	6.14E-01	4.28E-02	7.86E-02	8.19E-05
F22	3.50E-02	4.11E-03	5.75E-06	5.22E-06	6.34E-06	3.61E-03	2.35E-06	8.47E-06	1.49E-05
F23	5.61E-06	1.73E-06	2.50E-01	1.00E + 00	1.00E + 00	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F24	1.73E-06	2.85E-06	4.07E-02	6.96E-01	2.46E-03	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F25	1.96E-04	3.13E-02	1.00E + 00	1.00E + 00	1.00E + 00	3.79E-06	1.73E-06	1.73E-06	1.73E-06
F26	1.91E-04	1.22E-03	7.88E-03	2.06E-01	2.06E-01	3.16E-02	1.06E-01	5.45E-02	8.22E-02
F27	1.73E-06	1.73E-06	1.73E-06	4.01E-05	1.00E + 00	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F28	2.56E-06	8.30E-06	2.44E-04	1.73E-06	1.00E + 00	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F29	3.51E-06	2.88E-06	1.11E-05	1.73E-06	3.96E-01	1.92E-06	1.73E-06	1.73E-06	3.88E-06
F30	1.73E-06	1.73E-06	1.92E-06	1.73E-06	1.83E-04	1.73E-06	1.92E-06	1.92E-06	1.73E-06

References

Abd Elaziz, M., Heidari, A. A., Fujita, H., & Moayedi, H. (2020). A competitive chain-based Harris Hawks Optimizer for global optimization and multi-level image thresholding problems. *Applied Soft Computing, 106347*.

Adarsh, B. R., Raghunathan, T., Jayabarathi, T., & Yang, X.-S. (2016). Economic dispatch using chaotic bat algorithm (Vol. 96).

Awad, N. H., Ali, M. Z., & Suganthan, P. N. (2017). Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In *2017 IEEE Congress on Evolutionary Computation (CEC)* (pp. 372–379).

Ba, A. F., Huang, H., Wang, M., Ye, X., Gu, Z., Chen, H., & Cai, X. (2020). Levy-based antlion-inspired optimizers with orthogonal learning scheme. *Engineering with Computers*. https://doi.org/10.1007/s00366-00020-01042-00367

Betley, J. N., Xu, S., Cao, Z. F. H., Gong, R., Magnus, C. J., Yu, Y., & Sternson, S. M. (2015). Neurons for hunger and thirst transmit a negative-valence teaching signal. *Nature, 521*, 180–185.

Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation, 10*, 646–657.

Bai, B., Guo, Z., Zhou, C., Zhang, W., & Zhang, J. (2021). Application of adaptive reliability importance sampling-based extended domain PSO on single mode failure in reliability engineering. *Information Sciences, 546*, 42–59.

Burnett, C. J., Li, C., Webber, E., Tsaousidou, E., Xue, S. Y., Brüning, J. C., & Krashes, M. J. (2016). Hunger-driven motivational state competition. *Neuron, 92*, 187–201.

Cannon, W. B., & Washburn, A. (1912). An explanation of hunger. *American Journal of Physiology-Legacy Content, 29*, 441–454.

Cao, B., Fan, S., Zhao, J., Yang, P., Muhammad, K., & Tanveer, M. (2020). Quantum-enhanced multiobjective large-scale optimization via parallelism. *Swarm and Evolutionary Computation, 57*, Article 100697.

Cao, B., Wang, X., Zhang, W., Song, H., & Lv, Z. (2020). A many-objective optimization model of industrial internet of things based on private blockchain. *IEEE Network, 34*, 78–83.

Cao, B., Zhao, J., Yang, P., Gu, Y., Muhammad, K., Rodrigues, J. J., & de Albuquerque, V. H. C. (2019). Multiobjective 3-D topology optimization of next-generation wireless data center network. *IEEE Transactions on Industrial Informatics, 16*, 3597–3605.

Chen, H., Yang, C., Heidari, A. A., & Zhao, X. (2019). An Efficient Double Adaptive Random Spare Reinforced Whale Optimization Algorithm. Expert Systems with Applications, 113018.

Chen, H., Wang, M., & Zhao, X. (2020). A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems. Applied Mathematics and Computation, 369, 124872 (https://doi.org/124810.121016/j.amc.122019.124872).

Chen, Y., He, L., Guan, Y., Lu, H., & Li, J. (2017). Life cycle assessment of greenhouse gas emissions and water-energy optimization for shale gas supply chain planning based on multi-level approach: Case study in Barnett, Marcellus, Fayetteville, and Haynesville shales. *Energy Conversion and Management, 134*, 382–398.

Chen, H., Heidari, A. A., Chen, H., Wang, M., Pan, Z., & Gandomi, A. H. (2020). Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies. *Future Generation Computer Systems, 111*, 175–198.

Chen, H., Heidari, A. A., Zhao, X., Zhang, L., & Chen, H. (2020). Advanced orthogonal learning-driven multi-swarm sine cosine optimization: Framework and case studies. *Expert Systems with Applications, 144*, Article 113113.

Chen, H., Li, S., Heidari, A. A., Wang, P., Li, J., Yang, Y., … Huang, C. (2020). Efficient multi-population outpost fruit fly-driven optimizers: Framework and advances in support vector machines. *Expert Systems with Applications, 142*.

Chen, H. L., Wang, G., Ma, C., Cai, Z. N., Liu, W. B., & Wang, S. J. (2016). An efficient hybrid kernel extreme learning machine approach for early diagnosis of Parkinson's disease. *Neurocomputing, 184*, 131–144.

Chen, H., Xu, Y., Wang, M., & Zhao, X. (2019). A balanced whale optimization algorithm for constrained engineering design problems. *Applied Mathematical Modelling, 71*, 45–59.

Chen, H., Yang, B., Liu, D., Liu, W., Liu, Y., Zhang, X., & Hu, L. (2015). Using blood indexes to predict overweight statuses: An extreme learning machine-based approach. *PLoS One, 10*, Article e0143003.

Cheng, M. Y., & Prayogo, D. (2014). Symbiotic organisms search: A new metaheuristic optimization algorithm. *Computers and Structures, 139*, 98–112.

Clutton-Brock, T. (2009). Cooperation between non-kin in animal societies. *Nature, 462*, 51–57.
Deng, Y., Zhang, T., Sharma, B. K., & Nie, H. (2019). Optimization and mechanism studies on cell disruption and phosphorus recovery from microalgae with magnesium modified hydrochar in assisted hydrothermal system. *Science of The Total Environment, 646*, 1140–1154.
Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonprametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithm (Vol. 1).
Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey (Vol. 344).
Elsayed abd el aziz, M., & Oliva, D. (2018). Parameter estimation of solar cells diode models by an improved opposition-based whale optimization algorithm (Vol. 171).
Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm - A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers and Structures, 110–111*, 151–166.
Fan, Y., Wang, P., Heidari, A. A., Wang, M., Zhao, X., Chen, H., & Li, C. (2020). Rationalized fruit fly optimization with sine cosine algorithm: A comprehensive analysis. *Expert Systems with Applications, 113486*.
Fei, X., Wang, J., Ying, S., Hu, Z., & Shi, J. (2020). Projective parameter transfer based sparse multiple empirical kernel learning Machine for diagnosis of brain disease. *Neurocomputing, 413*, 271–283.
Friedman, I. M., & Stricker, E. (1976). The physiological psychology of hunger: A physiological perspective. *Psychological Review, 83*, 409–431.
Friedman, M. I., Ulrich, P., & Mattes, R. (1999). A figurative measure of subjective hunger sensations. *Appetite, 32*, 395–404.
Fu, X., Fortino, G., Li, W., Pace, P., & Yang, Y. (2019). WSNs-assisted opportunistic network for low-latency message forwarding in sparse settings. *Future Generation Computer Systems, 91*, 223–237.
Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2013). Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Engineering with Computers, 29*, 17–35.
García, S., Fernández, A., Luengo, J., & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power (Vol. 180).
Gotceitas, V., & Godin, J.-G. (1991). Foraging under the risk of predation in juvenile Atlantic salmon (Salmo salar L.): Effects of social status and hunger. *Behavioral Ecology and Sociobiology, 29*, 255–261.
Guo, S., Tsai, J. S., Yang, C., & Hsu, P. (2015). A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set. In *2015 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1003–1010).
Gupta, S., & Deep, K. (2018). A hybrid self-adaptive sine cosine algorithm with opposition based learning (Vol. 119).
Hansen, N., Müller, S. D., & Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation, 11*, 1–18.
Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications (Vol. 97).
Holland, J. H. (1992). Genetic algorithms. *Scientific American, 267*, 66–72.
Hu, J., Chen, H., Heidari, A. A., Wang, M., Zhang, X., Chen, Y., & Pan, Z. (2021). Orthogonal learning covariance matrix for defects of grey wolf optimizer: Insights, balance, diversity, and feature selection. *Knowledge-Based Systems, 213*, 106684.
Hu, L., Hong, G., Ma, J., Wang, X., & Chen, H. (2015). An efficient machine learning approach for diagnosis of paraquat-poisoned patients. *Computers in Biology and Medicine, 59*, 116–124.
Hu, L., Li, H., Cai, Z., Lin, F., Hong, G., Chen, H., & Lu, Z. (2017). A new machine-learning method to prognosticate paraquat poisoned patients by combining coagulation, liver, and kidney indices. *PLoS One, 12*, Article e0186427.
Huang, H., Feng, X. a., Zhou, S., Jiang, J., Chen, H., Li, Y., & Li, C. (2019). A new fruit fly optimization algorithm enhanced support vector machine for diagnosis of breast cancer based on high-level features. BMC Bioinformatics, 20.
Huang, H., Heidari, A. A., Xu, Y., Wang, M., Liang, G., Chen, H., & Cai, X. (2020). Rationalized sine cosine optimization with efficient searching patterns. *IEEE access, 8*, 61471–61490.
Jarvandi, S., Booth, D., & Thibault, L. (2007). Hyper-homeostatic learning of anticipatory hunger in rats. *Physiology & Behavior, 92*, 541–547.
Jiang, Q., Shao, F., Lin, W., Gu, K., Jiang, G., & Sun, H. (2017). Optimizing multistage discriminative dictionaries for blind image quality assessment. *IEEE Transactions on Multimedia, 20*(8), 2035–2048.
Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm (Vol. 39).
Kaveh, A., & Mahdavi, V. R. (2014). Colliding bodies optimization: A novel meta-heuristic method. *Computers and Structures, 139*, 18–27.
Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In IEEE International Conference on Neural Networks - Conference Proceedings (Vol. 4, pp. 1942-1948).
Khashan, N., El-Hosseini, M., Y. Haikal, A., & Badawy, M. (2018). Biped Robot Stability Based on an A-C parametric Whale Optimization Algorithm (Vol. 31).
Koza, J. R., & Rice, J. P. (1992). Automatic programming of robots using genetic programming. In *Proceedings Tenth National Conference on Artificial Intelligence* (pp. 194–201).
Kumar, N., Hussain, I., Singh, B., & Panigrahi, B. (2017). Single Sensor-Based MPPT of Partially Shaded PV System for Battery Charging by Using Cauchy and Gaussian Sine Cosine Optimization (Vol. PP).
Lee, K. S., & Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering, 194*, 3902–3933.
Li, J., Chen, C., Chen, H., & Tong, C. (2017). Towards context-aware social recommendation via individual trust. *Knowledge-Based Systems, 127*, 58–66.
Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems, 111*, 300–323.
Li, Y., Cui, W.-G., Huang, H., Guo, Y.-Z., Li, K., & Tan, T. (2019). Epileptic seizure detection in EEG signals using sparse multiscale radial basis function networks and the Fisher vector approach. *Knowledge-Based Systems, 164*, 96–106.
Li, C., Hou, L., Sharma, B. Y., Li, H., Chen, C., Li, Y., … Chen, H. (2018). Developing a new intelligent system for the diagnosis of tuberculous pleural effusion. *Computer Methods and Programs in Biomedicine, 153*, 211–225.
Li, X., Huang, H., Zhao, H., Wang, Y., & Hu, M. J. T. V. C. (2020). Learning a convolutional neural network for propagation-based stereo image segmentation. *The Visual Computer, 36*, 39–52.
Li, J., & Lin, J. (2020). A probability distribution detection based hybrid ensemble QoS prediction approach. *Information Sciences, 519*, 289–305.
Li, Y., Liu, Y., Cui, W., Guo, Y., Huang, H., & Hu, Z. (2020). Epileptic seizure detection in EEG signals using a unified temporal-spectral squeeze-and-excitation network. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 28*, 782–794.
Li, Y., Liu, X., Zhang, S., & Ye, X. J. N. (2016). Human articulated body recognition method in high-resolution monitoring images. *Neurocomputing, 181*, 116–121.
Li, Y., Zhang, S., & Zhang, L. J. M. S. (2016). Mining location-aware discriminative blocklets for recognizing landmark architectures. *Multimedia System, 22*, 455–464.
Li, X., Zhao, H., Huang, H., Hu, Z., Xiao, L. J. M. T., & Applications. (2016). Interactive image recoloring by combining global and local optimization. *Multimedia Tools and Applications, 75*, 6431–6443.
Li, J., Zheng, X.-L., Chen, S.-T., Song, W.-W., & Chen, D.-R. (2014). An efficient and reliable approach for quality-of-service-aware service composition. *Information Sciences, 269*, 238–254.
Li, X., Zhu, Y., & Wang, J. (2019). Highly efficient privacy preserving location-based services with enhanced one-round blind filter. *IEEE Transactions on Emerging Topics in Computing, 1*.
Liang, X., Cai, Z., Wang, M., Zhao, X., Chen, H., & Li, C. (2020). Chaotic oppositional sine–cosine method for solving global optimization problems. *Engineering with Computers*. https://doi.org/10.1007/s00366-00020-01083-y
Liang, H., Liu, Y., Shen, Y., Li, F., & Man, Y. (2018). A Hybrid Bat Algorithm for Economic Dispatch With Random Wind Power (Vol. PP).
Liu, Y., Chong, G., Heidari, A. A., Chen, H., Liang, G., Ye, X., … Wang, M. (2020). Horizontal and vertical crossover of Harris hawk optimizer with Nelder-Mead simplex for parameter estimation of photovoltaic models. Energy Conversion and Management, 223, 113211 (https://doi.org/113210.111016/j.enconman.112020.113211).
Liu, E., Guo, B., Lv, L., Qiao, W., & Azimi, M. (2020). Numerical simulation and simplified calculation method for heat exchange performance of dry air cooler in natural gas pipeline compressor station. *Energy Science & Engineering*. https://doi.org/10.1002/ese3.661
Liu, D., Wang, S., Huang, D., Deng, G., Zeng, F., & Chen, H. (2016). Medical image classification using spatial adjacent histogram based on adaptive local binary patterns. *Computers in Biology and Medicine, 72*, 185–200.
Liu, E., Wang, X., Zhao, W., Su, Z., & Chen, Q. (2020). Analysis and Research on Pipeline Vibration of a Natural Gas Compressor Station and Vibration Reduction Measures. *Energy & Fuels*. https://doi.org/10.1021/acs.energyfuels.0c03663
Long, Q., Wu, C., & Wang, X. (2015). A system of nonsmooth equations solver based upon subgradient method. *Applied Mathematics and Computation, 251*, 284–299.
Luo, J., Chen, H., Heidari, A. A., Xu, Y., Zhang, Q., & Li, C. (2019). Multi-strategy boosted mutative whale-inspired optimization approaches. *Applied Mathematical Modelling, 73*, 109–123.
Luo, J., Chen, H., Zhang, Q., Xu, Y., Huang, H., & Zhao, X. (2018). An improved grasshopper optimization algorithm with application to financial stress prediction. *Applied Mathematical Modelling, 64*, 654–668.
Miller, N. E., Bailey, C. J., & Stevenson, J. A. F. (1950). Decreased "Hunger" but increased food intake resulting from hypothalamic lesions. *Science (New York N.Y.), 112*, 256–259.
Mirjalili, S. (2015c). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems, 89*, 228–249.
Mirjalili, S., Mirjalili, S., & Hatamlou, A. (2015). Multi-Verse Optimizer: a nature-inspired algorithm for global optimization (Vol. 27).
Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163–191.
Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67.
Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61.
Mirjalili, S. (2015a). The Ant Lion Optimizer (Vol. 83).
Mirjalili, S. (2015b). Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems.
Mirjalili, S. (2016). SCA: A Sine Cosine Algorithm for Solving Optimization Problems (Vol. 96).
Ma, X., Zhang, K., Zhang, L., Yao, C., Yao, J., … Wang, H. J. (2021). Data-Driven Niching Differential Evolution with Adaptive Parameters Control for History Matching and Uncertainty Quantification. *SPE J*. https://doi.org/10.2118/205014-PA
Molina, D., Poyatos, J., Del Ser, J., García, S., Hussain, A., & Herrera, F. (2020). Comprehensive Taxonomies of Nature-and Bio-inspired Optimization: Inspiration Versus Algorithmic Behavior, Critical Analysis Recommendations. *Cognitive Computation, 12*(5), 897–939.

Nenavath, H., & Jatoth, R. K. (2018). Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. *Applied Soft Computing Journal, 62*, 1019–1043.

Ni, T., Chang, H., Song, T., Xu, Q., Huang, Z., Liang, H., … Wen, X. (2020). Non-intrusive online distributed pulse shrinking-based interconnect testing in 2.5D IC. *IEEE Transactions on Circuits and Systems II: Express Briefs, 67*, 2657–2661.

O'brien, W. J., Browman, H. I., & Evans, B. I. (1990). Search strategies of foraging animals. *American Scientist, 78*, 152–160.

Pang, J., Zhou, H., Tsai, Y.-C., & Chou, F.-D. (2018). A scatter simulated annealing algorithm for the bi-objective scheduling problem for the wet station of semiconductor manufacturing. *Computers & Industrial Engineering, 123*, 54–66.

Peng, S., Chen, Q., Zheng, C., & Liu, E. (2020). Analysis of particle deposition in a new-type rectifying plate system during shale gas extraction. *Energy Science & Engineering, 8*(3), 702–717. https://doi.org/10.1002/ese3.543

Peng, S., Zhang, Z., Liu, E., Liu, W., & Qiao, W. (2020). A new hybrid algorithm model for prediction of internal corrosion rate of multiphase pipeline. *Journal of Natural Gas Science and Engineering, 103716*. https://doi.org/10.1016/j.jngse.2020.103716

Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation, 13*, 398–417.

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *CAD Computer Aided Design, 43*, 303–315.

Real, L. A. (1991). Animal choice behavior and the evolution of cognitive architecture. *Science, 253*, 980–986.

Reppucci, C. J., & Veenema, A. H. (2020). The social versus food preference test: A behavioral paradigm for studying competing motivated behaviors in rodents. *MethodsX, 7*, Article 101119.

Ridha, H. M., Gomes, C., Hizam, H., Ahmadipour, M., Heidari, A. A., & Chen, H. Multi-objective optimization and multi-criteria decision-making methods for optimal design of standalone photovoltaic system: A comprehensive review. Renewable and Sustainable Energy Reviews, 135, 110202.

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper Optimisation Algorithm: Theory and application (Vol. 105).

Savsani, P., & Savsani, V. (2016). Passing vehicle search (PVS): A novel metaheuristic algorithm. *Applied Mathematical Modelling, 40*, 3951–3978.

Shen, L., Chen, H., Yu, Z., Kang, W., Zhang, B., Li, H., … Liu, D. (2016). Evolving support vector machines using fruit fly optimization for medical data classification. *Knowledge-Based Systems, 96*, 61–75.

Shi, K., Tang, Y., Zhong, S., Yin, C., Huang, X., & Wang, W. (2018). Nonfragile asynchronous control for uncertain chaotic Lurie network systems with Bernoulli stochastic process. *International Journal of Robust and Nonlinear Control, 28*, 1693–1714.

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation, 12*, 702–713.

Song, S., Wang, P., Heidari, A. A., Wang, M., Zhao, X., Chen, H., … Xu, S. (2020). Dimension decided Harris hawks optimization with Gaussian mutation: Balance analysis and diversity patterns. *Knowledge-Based Systems, 106425*. https://doi.org/10.1016/j.knosys.102020.106425

Song, J., Zhong, Q., Wang, W., Su, C., Tan, Z., & Liu, Y. (2020). FPDP: Flexible privacy-preserving data publishing scheme for smart agriculture. *IEEE Sensors Journal*.

Storn, R., & Price, K. (1997). Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*, 341–359.

Sun, X., Liu, Y., Wei, D., Xu, M., Chen, H., & Han, J. (2013). Selection of interdependent genes via dynamic relevance analysis for cancer diagnosis. *Journal of Biomedical Informatics, 46*, 252–258.

Sun, G., Li, C., & Deng, L. (2021). An adaptive regeneration framework based on search space adjustment for differential evolution. *Neural Comput & Applic*. https://doi.org/10.1007/s00521-021-05708-1

Sutton, A. K., & Krashes, M. J. (2020). Integrating hunger with rival motivations. *Trends in Endocrinology & Metabolism, 31*, 495–507.

Tanabe, R., & Fukunaga, A. (2014). Improving the Search Performance of SHADE Using Linear Population Size Reduction.

Tu, J., Chen, H., Liu, J., Heidari, A. A., Zhang, X., Wang, M., … Pham, Q.-V. (2020). Evolutionary biogeography-based Whale optimization methods with communication structure: Towards measuring the balance. *Knowledge-Based Systems, 106642*. https://doi.org/10.1016/j.knosys.102020.106642

Tubishat, M., Abushariah, M., Idris, N., & Aljarah, I. (2018). Improved whale optimization algorithm for feature selection in Arabic sentiment analysis.

Wang, G. G. (2003). Adaptive response surface method using inherited Latin hypercube design points. *Journal of Mechanical Design, Transactions of the ASME, 125*, 210–220.

Wang, M., & Chen, H. (2020). Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis. Applied Soft Computing, 88, 105946 (https://doi.org/105910.101016/j.asoc.102019.105946).

Wang, M., Chen, H., Li, H., Cai, Z., Zhao, X., Tong, C., … Xu, X. (2017). Grey wolf optimization evolving kernel extreme learning machine: Application to bankruptcy prediction. *Engineering Applications of Artificial Intelligence, 63*, 54–68.

Wang, S. J., Chen, H. L., Yan, W. J., Chen, Y. H., & Fu, X. (2014). Face recognition and micro-expression recognition based on discriminant tensor subspace analysis plus extreme learning machine. *Neural Processing Letters, 39*, 25–43.

Wang, M., Chen, H., Yang, B., Zhao, X., Hu, L., Cai, Z., … Tong, C. (2017). Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses. *Neurocomputing, 267*, 69–84.

Wang, G.-G., Deb, S., & Cui, Z. (2019). Monarch butterfly optimization. *Neural Computing and Applications, 31*, 1995–2014.

Wang, X., Chen, H., Heidari, A. A., Zhang, X., Xu, J., Xu, Y., & Huang, H. (2020). Multi-population following behavior-driven fruit fly optimization: A Markov chain convergence proof and comprehensive analysis. Knowledge-Based Systems, 210, 106437 (https://doi.org/106410.101016/j.knosys.102020.106437).

Wang, Y., See, J., Oh, Y.-H., Phan, R. C. W., Rahulamathavan, Y., Ling, H.-C., … Li, X. (2017). Effective recognition of facial micro-expressions with video motion magnification. *Multimedia Tools and Applications, 76*, 21665–21690.

Wang, P., Xu, S., Li, Y., Wang, L., & Song, Q. (2018). Feature-based analysis of cell nuclei structure for classification of histopathological images. *Digital Signal Processing, 78*, 152–162.

Wang, T., Zhang, X., Jiang, R., Zhao, L., Chen, H., Luo, W. J. C. V., & Understanding, I. (2020). Video deblurring via spatiotemporal pyramid network and adversarial gradient prior. *Computer Vision and Image Understanding, 203*, Article 103135.

Wei, Y., Lv, H., Chen, M., Wang, M., Heidari, A. A., Chen, H., & Li, C. (2020). Predicting entrepreneurial intention of students: An extreme learning machine with Gaussian barebone Harris hawks optimizer. *IEEE Access, 8*, 76841–76855.

Wen, F. H., Yang, X., Gong, X., & Lai, K. K. (2017). Multi-scale volatility feature analysis and prediction of gold price. *International Journal of Information Technology & Decision Making, 16*, 205–223.

Wolpert, D., & Macready, W. (1997). No free lunch theorems for optimization (Vol. 1).

Wu, G., Mallipeddi, R., Suganthan, P., Wang, R., & Chen, H. (2015). Differential evolution with multi-population based ensemble of mutation strategies. *Information Sciences*. https://doi.org/10.1016/j.ins.2015.1009.1009

Wu, C., Wu, P., Wang, J., Jiang, R., Chen, M., & Wang, X. (2020). Critical review of data-driven decision-making in bridge operation and maintenance. *Structure and Infrastructure Engineering*, 1–24.

Wu, C., Yang, Z., Xu, Y., Zhao, Y., & Liu, Y. (2014). Human mobility enhances global positioning accuracy for mobile phone localization. *IEEE Transactions on Parallel and Distributed Systems, 26*, 131–141.

Xia, J., Chen, H., Li, Q., Zhou, M., Chen, L., Cai, Z., … Zhou, H. (2017). Ultrasound-based differentiation of malignant and benign thyroid Nodules: An extreme learning machine approach. *Computer Methods and Programs in Biomedicine, 147*, 37–49.

Xiong, Z., Xiao, N., Xu, F., Zhang, X., Xu, Q., Zhang, K., & Ye, C. (2020). An equivalent exchange based data forwarding incentive scheme for socially aware networks. *Journal of Signal Processing Systems*, 1–15.

Xu, Y., Shi, X. J. A., & Networks, S. W. (2015). ABAL: Aerial beacon assisted localization of wireless sensor networks with RSS maxima. *Adhoc & Sensor Wireless Networks, 29*, 93–112.

Xue, X., Zhang, K., Tan, K. C., Feng, L., Wang, J., Chen, G., … Yao, J. (2020). Affine Transformation-Enhanced Multifactorial Optimization for Heterogeneous Problems. *IEEE Transactions on Cybernetics*. https://doi.org/10.1109/TCYB.2020.3036393

Yang, X.-S., & Deb, S. (2009). Cuckoo search via Lévy flights. In 2009 World congress on nature & biologically inspired computing (NaBIC) (pp. 210-214): IEEE.

Yang, X.-S., Karamanoglu, M., & Xingshi, H. (2014). Flower pollination algorithm: A novel approach for multiobjective optimization (Vol. 46).

Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 5792 LNCS, pp. 169-178).

Yang, X.-S. (2010). A New Metaheuristic Bat-Inspired Algorithm (Vol. 284).

Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation, 3*, 82–102.

Yong, J., He, F., Li, H., & Zhou, W. (2018). A Novel Bat Algorithm based on Collaborative and Dynamic Learning of Opposite Population.

Yu, C., Chen, M., Cheng, K., Zhao, X., Ma, C., Kuang, F., & Chen, H. (2021). SGOA: Annealing-behaved grasshopper optimizer for global tasks. *Engineering with Computers*. https://doi.org/10.1007/s00366-00020-01234-00361

Yu, C., Heidari, A. A., & Chen, H. (2020). A quantum-behaved simulated annealing enhanced moth-flame optimization method. *Applied Mathematical Modelling*.

Yu, H., Zhao, N., Wang, P., Chen, H., & Li, C. (2020). Chaos-enhanced synchronized bat optimizer. *Applied Mathematical Modelling, 77*, 1201–1215.

Zhang, X., Jiang, R., Wang, T., Huang, P., & Zhao, L. J. N. (2020). Attention-based interpolation network for video deblurring. *Neurocomputing*. https://doi.org/10.1016/j.neucom.2020.1004.1147

Zhang, S., Li, P., Zhong, Y., & Xiang, J. (2014). Structural topology optimization based on the level set method using COMSOL. *CMES-Computer Modeling in Engineering & Sciences, 101*, 17–31.

Zhang, Y., Liu, R., Asghar Heidari, A., Wang, X., Chen, Y., Wang, M., & Chen, H. (2020). Towards augmented kernel extreme learning models for bankruptcy prediction: Algorithmic behavior and comprehensive analysis. *Neurocomputing*. https://doi.org/10.1016/j.neucom.2020.1010.1038

Zhang, S., Wang, J., Ghoshal, T., Wilkins, D., Mo, Y.-Y., Chen, Y., & Zhou, Y. (2018). lncRNA gene signatures for prediction of breast cancer intrinsic subtypes and prognosis. *Genes, 9*.

Zhang, X., Wang, D., Zhou, Z., & Ma, Y. J. I. (2019). Robust low-rank tensor recovery with rectification and alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*, 238–255.

Zhang, X., Xu, Y., Yu, C., Heidari, A. A., Li, S., Chen, H., & Li, C. (2020). Gaussian mutational chaotic fruit fly-built optimization and feature selection. *Expert Systems with Applications, 141*, Article 112976.

Zhao, H., Gao, D., Wang, M., Pan, Z. J. C., & Graphics. (2016). Real-time edge-aware weighted median filtering on the GPU. *Computers & Graphics, 61*, 11–18.

Zhao, H., Jiang, L., Jin, X., Du, H., & Li, X. J. T. V. C. (2018). Constant time texture filtering. *The Visual Computer, 34*, 83–92.

Zhao, D., Liu, L., Yu, F., Heidari, A. A., Wang, M., Liang, G., … Chen, H. (2020). Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D

Kapur entropy. *Knowledge-Based Systems, 106510*. https://doi.org/10.1016/j.knosys.102020.106510
Zhao, D., Liu, L., Yu, F., Heidari, A. A., Wang, M., Oliva, D., … Chen, H. (2020). Ant colony optimization with horizontal and vertical crossover search: Fundamental visions for multi-threshold image segmentation. *Expert Systems with Applications, 114122*.
Zhao, X., Zhang, X., Cai, Z., Tian, X., Wang, X., Huang, Y., … Hu, L. (2019). Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients. *Computational Biology and Chemistry, 78*, 481–490.
Zhou, H., Pang, J., Chen, P.-K., & Chou, F.-D. (2018). A modified particle swarm optimization algorithm for a batch-processing machine scheduling problem with arbitrary release times and non-identical job sizes. *Computers & Industrial Engineering, 123*, 67–81.
Zhou, Y., Tian, L., Zhu, C., Jin, X., & Sun, Y. (2019). Video coding optimization for virtual reality 360-degree source. *IEEE Journal of Selected Topics in Signal Processing, 14*(1), 118–129.
Zhou, Y., Yang, B., Hou, H., Zhang, L., Wang, T., & Hu, M. (2019). Continuous leakage-resilient identity-based encryption with tight security. *The Computer Journal, 62*, 1092–1105.
Zhu, W., Ma, C., Zhao, X., Wang, M., Heidari, A. A., Chen, H., & Li, C. (2020). Evaluation of sino foreign cooperative education project using orthogonal sine cosine optimized kernel extreme learning machine. *IEEE Access, 8*, 61107–61123.