

# **IEICE** **TRANSACTIONS**

## **on Fundamentals of Electronics, Communications and Computer Sciences**

DOI:10.1587/transfun.2023EAP1156

Publicized:2024/08/05

This advance publication article will be replaced by  
the finalized version after proofreading.



A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

## PAPER

# Memetic Gravitational Search Algorithm with Hierarchical Population Structure

Shibo DONG<sup>†</sup>, Haotian LI<sup>†</sup>, Yifei YANG<sup>††</sup>, Jiatianyi YU<sup>†</sup>, Zhenyu LEI<sup>†</sup>, *Nonmembers*,  
and Shangce GAO<sup>†a)</sup>, *Member*

**SUMMARY** The multiple chaos embedded gravitational search algorithm (CGSA-M) is an optimization algorithm that utilizes chaotic graphs and local search methods to find optimal solutions. Despite the enhancements introduced in the CGSA-M algorithm compared to the original GSA, it exhibits a pronounced vulnerability to local optima, impeding its capacity to converge to a globally optimal solution. To alleviate the susceptibility of the algorithm to local optima and achieve a more balanced integration of local and global search strategies, we introduce a novel algorithm derived from CGSA-M, denoted as CGSA-H. The algorithm alters the original population structure by introducing a multi-level information exchange mechanism. This modification aims to mitigate the algorithm's sensitivity to local optima, consequently enhancing the overall stability of the algorithm. The effectiveness of the proposed CGSA-H algorithm is validated using the IEEE CEC2017 benchmark test set, consisting of 29 functions. The results demonstrate that CGSA-H outperforms other algorithms in terms of its capability to search for global optimal solutions.

**key words:** *hierarchical, population structure, memetic algorithms, meta-heuristic algorithms, gravitational search algorithm*

## 1. Introduction

Inspired by natural and physical phenomena, evolution algorithms are experiencing rapid development. These algorithms prove to be highly effective during the search phase of problem optimization, significantly enhancing their capacity to explore global optimal solutions. They emulate biological behaviors and physical phenomena, which can be translated into efficient search mechanisms. Through multiple generations of continuous iteration and refinement, these algorithms exhibit exceptional search processes, greatly improving the performance in obtaining the best solutions. Evolutionary algorithms have been successfully used in a wide range of applications to solve various optimisation problems [1–3].

Currently, the field of optimization has been dominated by two main approaches: traditional mathematical optimization methods and meta-heuristic algorithms [4, 5]. The latter category includes popular algorithms such as the genetic algorithm [6], ladder sphere evolution search algorithm [7], and ant colony optimization [8]. Recently, the focus of many research labs has been on finding efficient solutions to complex optimization problems found in real-world scenarios, such as the dynamic location routing problem [9], wave en-

ergy converter position optimization problem [10], and artificial neural model training problem [11]. To tackle these challenging problems, researchers have looked to nature for inspiration, leading to the development of various evolution algorithms [12, 13]. These algorithms draw on the principles and behaviors observed in natural phenomena to inform their approach to solving optimization problems.

Most evolution algorithms typically use a panmictic population structure, where each individual has an equal chance to interact with others [14]. However, in recent years, a distributed population structure has become increasingly popular in parallel computing [15]. This approach employs multiple sub-populations to handle all individuals, with a panmictic approach usually performed within each sub-population. In contrast, a cellular structure only allows individuals to communicate with those in their predefined neighborhood [16, 17]. The hierarchical structure can facilitate information exchange among different sub-populations more frequently, thus improving the overall search efficiency of the algorithm [18–21]. Unlike homogeneous population structures, the small-world [22–24] and scale-free [25] structures consider the population as non-homogeneous networks, where some individuals have a higher probability of interacting with others. These structures aim to accelerate the convergence speed of the search algorithm.

The integration of population structure with meta-heuristic optimization algorithms, incorporating mechanisms such as stratification, distribution, collaboration, and adaptive adjustment, enhances algorithmic performance by achieving a better balance between exploration and exploitation, fostering global search, and maintaining diversity. This approach has demonstrated impressive success when combined with various metaheuristic optimization algorithms, such as DWSA (Distributional Western Chicken Swarm Algorithm) [26] and MLSGSA (Multi-Level Gravitational Search Algorithm) [21]. The results indicate that the population structure mitigates issues related to local optima, significantly improving search capabilities.

The Gravitational Search Algorithm (GSA) [27] is an exceptionally efficient optimization algorithm that has garnered considerable attention in the field [28–34]. In order to enhance its performance and bolster stability, several new iterations of GSA have been proposed, such as [35, 36]. CGSA-M represents a successful approach that integrates multiple chaotic mappings into the GSA framework. The decision to employ diverse chaotic mappings for updates

<sup>†</sup>Faculty of Engineering, University of Toyama, Toyama 930-8555, Japan.

<sup>††</sup>Faculty of Science and Technology, Hirosaki University, Hirosaki-shi, 036-8560, Japan.

a) E-mail: gaosc@eng.u-toyama.ac.jp

is determined through an adaptive approach, aimed at augmenting the search capability across different phases. However, despite these improvements, it still encounters challenges in avoiding local optima, resulting in suboptimal solutions.

However, this strategy of using chaotic mappings to enhance the local search capability of the algorithm, while increasing the search power, also raises the possibility of the algorithm getting trapped in local optima. In order to overcome the limitations of CGSA-M, we propose a novel memetic gravitational search algorithm with a hierarchical population structure called CGSA-H. Our algorithm introduces, for the first time, the concept of a multi-level population structure. By employing this novel design, we achieve a balance between exploration and exploitation during the search process. The larger subpopulations are well-suited for exploring the global solution space, while the smaller ones excel in the precise search of local solution spaces. This progressive optimization approach, facilitated by the hierarchical design, helps to prevent premature convergence to local optima. Through the utilization of the multi-level population structure, we are able to comprehensively explore the solution space of the problem, thereby enhancing the algorithm's performance and efficiency when dealing with complex problems. The proposed hierarchical structure allows different sub-populations to exchange information more frequently, which improves the overall search efficiency of CGSA-H.

We test CGSA-H on the IEEE CEC2017 benchmark optimization functions, and our experimental results demonstrate that it significantly outperforms its counterparts in terms of solution quality and convergence speed. Additionally, the search trajectories of CGSA-H on unimodal, multimodal, and mixed search landscapes show well-maintained development and exploration capabilities. This study provides evidence that the incorporation of a multi-level population structure can effectively improve the performance of CGSA-M in solving complex optimization problems.

The main contributions of this study are as follows:

(1) To enhance the overall search efficiency of CGSA-M, we introduce a four-level hierarchical population structure. This design is intended to boost the frequency of information exchange among distinct subgroups, ultimately improving the algorithm's performance.

(2) CGSA-H achieves improved accuracy without the need for parameter adjustments, showcasing its inherent capability to enhance algorithm performance.

## 2. A succinct overview of the conventional Gravitational Search Algorithm (GSA)

The Gravitational Search Algorithm (GSA) is a population-based metaheuristic algorithm inspired by the gravitational law among objects. Within the GSA population, each individual is regarded as an object and is evaluated based on its mass as a measure of performance. The position of an individual corresponds to a solution of the optimization problem

under consideration. Altering the position of an individual has the potential to lead to an enhancement in the quality of the solution.

In a formal context, each entity denoted as  $X_i = (x_i^1, \dots, x_i^d, \dots, x_i^D)$ , ( $i = 1, 2, 3, \dots, N$ ) within the system exerts gravitational forces upon other such entities in a  $D$ -dimensional exploration domain. Here, the variable  $x_i^d$  signifies the positional coordinates of the  $i$ -th entity along the  $d$ -th dimension. The velocity associated with entity  $X_i$  is denoted as  $V_i = (V_i^1, \dots, V_i^d, \dots, V_i^D)$ . During iteration  $t$ , the mass of every entity, represented as  $M_i(t)$ , is computed using a fitness-based mapping procedure outlined as follows:

$$M_i(t) = \frac{fit(X_i(t)) - worst(t)}{best(t) - worst(t)} \quad (1)$$

In this context, the term denoted as  $fit(X_i(t))$  represents the fitness evaluation of agent  $X_i$ , which is determined by the computation of the objective function. For a problem aiming at minimization, we establish the definitions of  $best(t)$  and  $worst(t)$  as provided below.

$$\begin{aligned} best(t) &= \min_{j=1,2,3,\dots,N} fit(X_j(t)) \\ worst(t) &= \max_{j=1,2,3,\dots,N} fit(X_j(t)) \end{aligned} \quad (2)$$

The influence on the  $i$ -th agent by the  $j$ -th agent is described as:

$$F_{ij}^d = G(t) \frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)) \quad (3)$$

where  $R_{ij}(t)$  represents the Euclidean distance between the positions of agents  $x_i$  and  $x_j$  at time  $t$ , computed as  $R_{ij}(t) = \|x_i(t) - x_j(t)\|_2$ . The parameter  $\varepsilon$  is a small constant introduced to avoid division by zero in the denominator of Eq. (3). Furthermore, the term  $G(t)$  denotes the gravitational constant at time  $t$ , and it is defined by:

$$G(t) = G_0 \cdot e^{-\alpha \frac{t}{t_{max}}} \quad (4)$$

The initial value of the gravitational constant is represented by  $G_0$ , while  $\alpha$  serves as a shrinking constant, and  $t_{max}$  signifies the maximum number of iterations. Regarding the  $i$ -th agent, the collective force applied to it results from a summation of forces exerted by neighboring agents, with random weights.

$$F_d^i(t) = \sum_{j \in K_{best}, j \neq i} rand_j F_{ij}^d(t) \quad (5)$$

where  $K_{best}$  denotes the subset containing the initial  $K$  agents with the highest fitness and greatest mass, and  $rand_j$  signifies a randomly generated number following a uniform distribution within the range  $[0, 1]$ . Additionally,

$$K = \left\lfloor \left( \beta + (1 - \frac{t}{t_{max}}) (1 - \beta) \right) N \right\rfloor \quad (6)$$

where the initial value of  $K$  is set to  $N$  and is progressively reduced in a linear manner, under the influence of a constant parameter represented by  $\beta$ . The symbol  $\lfloor \cdot \rfloor$  signifies

the floor function. Adhering to the principles of motion, the acceleration of the  $i$ -th agent is determined through the following equation:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \quad (7)$$

Subsequently, the subsequent velocity of an agent is determined by adding a portion of its current velocity to the calculated acceleration. Consequently, updates to its position and velocity can be performed as outlined below:

$$v_i^d(t+1) = rand_i v_i^d(t) + a_i^d(t) \quad (8)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (9)$$

Here, the variable  $rand_i$  represents a random value sampled from the interval  $[0, 1]$ . It's essential to emphasize that both  $rand_i$  and  $rand_j$  are generated as uniformly distributed random numbers, and they typically exhibit distinct values. Indeed, they serve as a means to introduce randomized traits into the search process, contributing to its exploration capabilities.

### 3. Multiple Chaos Embedded Gravitational Search Algorithm

Chaos characterizes non-linear dynamic systems, displaying bounded dynamic instability, pseudo-randomness, thorough exploration, and aperiodic behavior tied to initial conditions and control parameters [37]. Chaotic systems exhibit random changes, yet over time, they encompass all possible states. This trait is useful for creating a search mechanism to optimize objective functions. However, chaotic optimization excels in smaller search spaces but becomes inefficient for larger ones [38], resulting in lengthy optimization periods. Hence, chaotic search is often integrated into other global optimizers like evolutionary algorithms to enhance their search efficiency [39–49]. Chaotic local search, unlike substituting random values with chaotic sequences for GSA's control parameters, significantly improves GSA performance [50]. Notably, studies frequently employ chaotic local search for this purpose [40–49]. Consequently, CGSA-M adopts the approach of chaotic local search.

The definition of parallel chaotic local search involving multiple chaotic elements is as follows.

$$X_g^j(t) = X_g(t) + r(t)(U - L)(z^j(t) - 0.5) \quad (10)$$

Here,  $X_g^j$ , for  $j = 1, 2, \dots, 12$  represents a provisional candidate solution generated through the utilization of parallel chaotic local search, signifying the simultaneous creation of twelve candidate solutions through distinct chaotic maps. Subsequently, the most optimal solution from the twelve candidates is selected for comparison with the current global best solution, denoted as  $X_g(t)$ . If an enhancement in fitness is observed, the original solution is substituted; otherwise, it remains unchanged. This updating process can be precisely articulated as follows:

$$X_g(t+1) = \begin{cases} X_g^{j_{min}}, & \text{if } fit(X_g^{j_{min}}) \leq fit(X_g(t)) \\ X_g(t), & \text{otherwise} \end{cases} \quad (11)$$

$$j_{min} = j \in \{1, 2, \dots, 12\} \text{ s.t. } \min_{j=1,2,\dots,12} fit(X_g^j(t)) \quad (12)$$

### 4. Memetic Gravitational Search Algorithm with Hierarchical Population Structure

Due to the presence of multiple chaos factors, CGSA-M tends to excessively emphasize exploration, often leading to the identification of suboptimal solutions in numerous scenarios. To achieve a more effective balance between algorithmic development and exploration, a novel four-layer hierarchical population structure, denoted as CGSA-H, has been introduced upon the foundation of the original CGSA-M algorithm. The incorporation of the most valuable individual layer aims to enhance the algorithm's convergence speed. Furthermore, the introduction of a historical information storage layer serves to mitigate the potential impact of local optimality resulting from improved developmental capabilities. The descriptions of these layers are provided below, and the main process of the proposed CGSA-H algorithm is outlined in Algorithm 1.

Within the population, the individual possessing the most valuable information is selected and referred to as the pivotal individual. This pivotal individual exerts influence on the individuals within the Dielectric layer, directing them to explore in the vicinity of the pivotal individual. This accelerates the convergence of the algorithm, contributing to the overall performance enhancement. In the most valuable individual layer, we conducted an effective perturbation search based on the results obtained from the Dielectric layer, which significantly enhanced the performance of the algorithm. In the Most Valuable Individuals layer, we utilize  $y_s$  to generate  $y_{s'}$  based on its optimal characteristics. Here,  $y_s$  represents the best individual in the population. The formulation of this process is expressed as follows:

$$y_{s'}(t) = Y_s(t) + p \cdot (Z_{r2}(t) - Z_{r1}(t)) \cdot rand(0, 1) \quad (13)$$

where  $p$  is a constant value. In this study,  $p$  represents a constant value, specifically set to 1. Two individuals, denoted as  $Z_{r1}$  and  $Z_{r2}$  are randomly selected from the original information layer. We illustrate that the algorithm maintains robust performance even when the search step size remains unchanged. In many algorithms, parameter adjustment is a crucial process that can impact the performance and accuracy of the algorithm. However, parameter adjustment can often be a challenging task, particularly for complex algorithms. Therefore, an algorithm that performs well without requiring parameter adjustment is highly desirable. This means that researchers can utilize the algorithm without the need to invest additional time and effort into parameter tuning are randomly chosen from the historical information layer, enabling them to focus on other essential tasks.

**Historical Information Layer:** While the inclusion of  $y_s$  expedites perturbation search in the proximity of the Most

**Algorithm 1:** The pseudo-code of CGSA-H.

---

**Input:** Objective Function  $f_x$ , search space  $[L, U]$ , and maximum number of function evaluations  $FES$ ;  
**Output:** Final optimal entity;  
**Initialize:** Randomly generate GSA population  $\{X_1, X_2, \dots, X_N\}$ ,  $nFES = 0$ ;  
**while**  $nFES \leq FES$  **do**  
  **while**  $i \leq N$  **do**  
    Compute overall force  $F_i^d(t)$  according to Eqs. (1) - (6);  
    Compute acceleration  $a_i^d(t)$  according to Eq. (7);  
    Update velocity according to Eq. (8);  
    Update position according to Eq. (9);  
    Evaluate the fitness of the current entity;  
  **end**  
  Perform multiple parallel chaotic local searches according to Eqs. (10) and (11);  
  Implement the chaotic local search approach;  
  Decrease the chaotic search radius;  
  Select the current optimal entity  $Y_s(t)$  based on fitness;  
  Generate a new entity  $y_{s'}$  According to Eq. (13);  
  Update the optimal entity  $Y_s(t)$  according to Eq. (14);  
  Update the historical information layer according to Eq. (15);  
**end**  
RETURN the optimal level of fitness.

---

Valuable Individuals, facilitating faster algorithm convergence, it also introduces the inherent risk of premature convergence to a local optimum. To mitigate this risk, we introduce a historical information layer capable of information exchange with the Most Valuable Individuals layer. This establishment aims to strike a balance between exploitation and exploration within the most valuable individual segment, thus averting premature convergence to a local optimum. Here,  $y_s$  represents the best individual in the population, and  $y'_s$  denotes a transient individual.  $f$  represents the fitness function. In the event that the fitness value of  $y'_s$  surpasses that of  $y_s$ ,  $y'_s$  supersedes  $y_s$ ; otherwise,  $y_s$  is retained and persists into subsequent iterations. The formulation is expressed as follows:

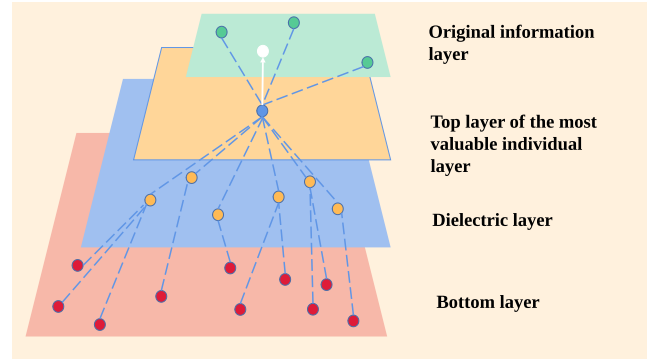
$$y_s(t) = \begin{cases} y_{s'}(t), & \text{if } f(y_{s'}(t)) \leq f(y_s(t)) \\ y_s(t), & \text{otherwise} \end{cases} \quad (14)$$

Within the population set  $Z$ , assuming the individual with the optimal fitness is denoted as  $Z_{max}$ . If the fitness of the individual  $y'_s$  surpasses that of  $Z_{max}$ , then  $Z_{max}$  is replaced with  $y'_s$ , thereby recording historical information. The formula expressing this process is as follows:

$$Z_{max}(t) = \begin{cases} y'_s(t) & \text{if } f(y'_s(t)) \leq f(Z_{max}(t)), \\ Z_{max}(t) & \text{otherwise} \end{cases} \quad (15)$$

CGSA-H is an enhanced version of CGSA-M designed to address the issues of low search accuracy and susceptibility to falling into local optima by incorporating a multi-level population structure.

As depicted in Fig. 1, CGSA-H constitutes a hierarchical structure comprising four layers. The initial CGSA-M is positioned in the first and second layers, while the top layer



**Fig. 1** Illustrative concept of hierarchical multiple chaos embedded gravitational search algorithm.

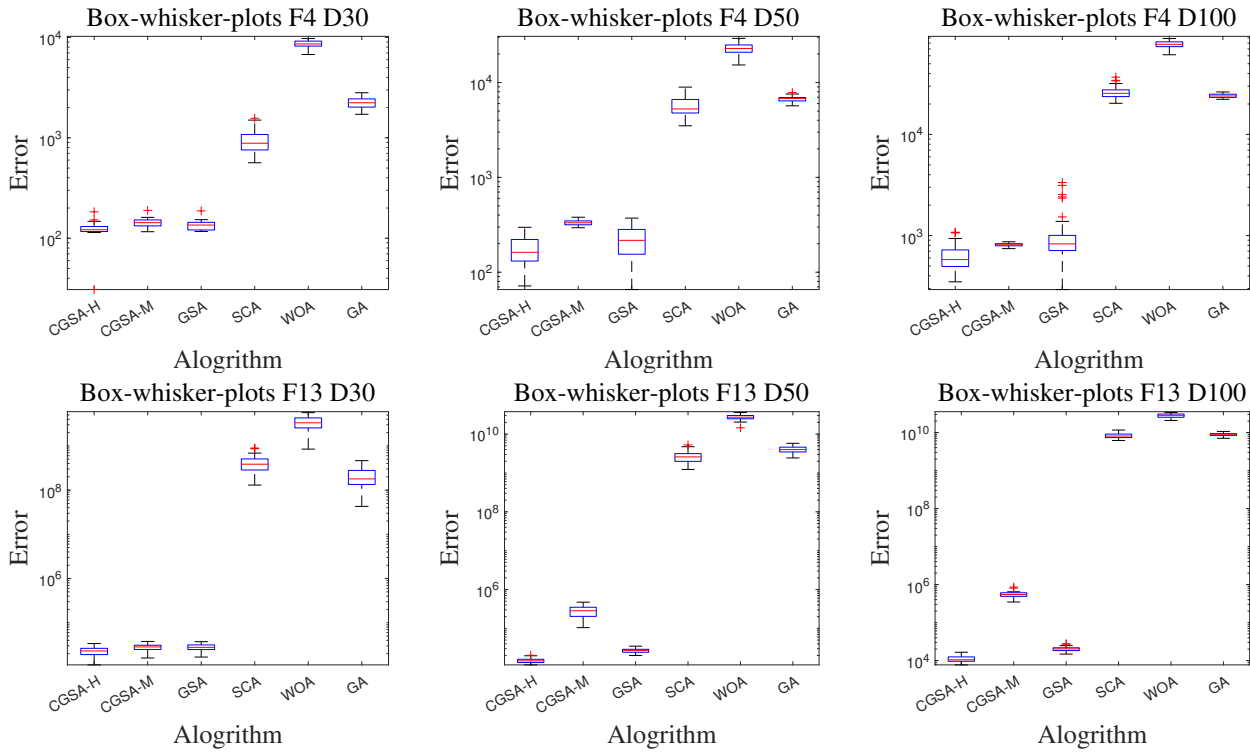
of the most valuable individual layer and the historical information layer are situated in the third and fourth layers, respectively. It is evident that there exists a bidirectional information flow between the historical information layer and the top layer of the most valuable individual layer. It is crucial to emphasize that, between the historical information layer and the top layer of the most valuable individual layer, white arrows signify the impact of optimal individual components on the information exchange components. Specifically, if the fitness of  $y'_s$  exceeds that of  $Z_{max}$ ,  $y'_s$  is employed to replace  $Z_{max}$ . Blue arrows indicate the update of the best individual component in the historical information storage layer.

In CGSA-H, we adjust the population structure based on the existing CGSA-M by adding two layers. The third layer, known as the most valuable individual layer, performs a perturbation search around the optimal solutions found in the second layer, thereby enhancing the performance of the algorithm. The fourth layer, the original information layer, is introduced due to the performance improvement from the most valuable individual layer, which inevitably increases the risk of the algorithm prematurely converging to local optima. To mitigate this risk, we introduce an original information layer capable of exchanging information with the most valuable individual layer.

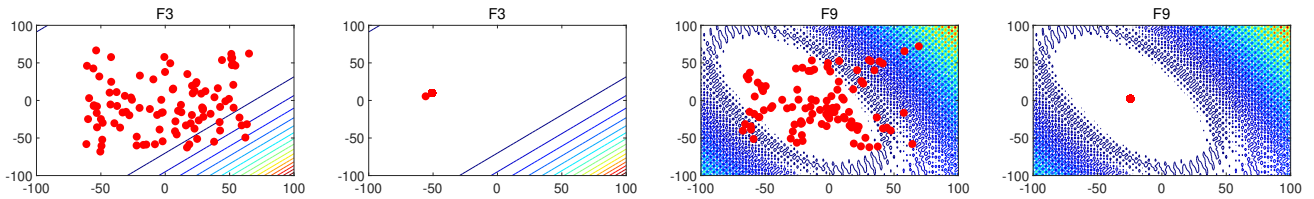
The multi-level population structure in CGSA-H achieves a superior balance between exploitation and exploration by enabling information exchange between the most valuable individual layer and the original information layer. By facilitating more frequent information exchange among different subpopulations, the algorithm can attain higher search accuracy and avoid premature convergence. Algorithm 1 presents the pseudo-code of CGSA-H.

## 5. Experiment Results

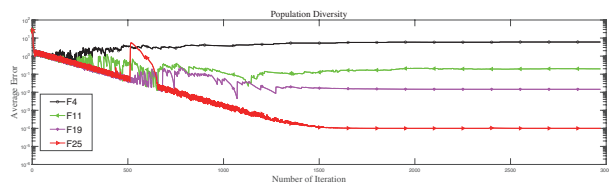
We test the proposed CGSA-H on a set of benchmark problems from IEEE CEC2017 to validate its performance. The IEEE CEC2017 problem set consists of 29 test problems, including 24 simplex optimization problems and 5 nonlinear optimization problems. Nonlinear optimization problems are those in which the objective function is nonlinear,



**Fig. 2** Box-whisker plots of algorithm performance metrics for functions F4 and F13 across multiple trials on IEEE CEC2017 test suite.



**Fig. 3** Search history of CGSA-H on F3 and F9 with two dimensions.



**Fig. 4** The population diversity of CGSA-H along with the iteration.

meaning it does not have a single peak. These test problems are designed to evaluate the performance of optimization algorithms on different types of optimization problems with different characteristics. It is worth noting that we exclude F2 from testing due to its instability, particularly in high-dimensional problems, and the significant differences in the performance of the same algorithm implemented in MATLAB.

The performance of CGSA-H is compared with several optimization algorithms, including the classical sine cosine algorithm (SCA), multiple chaos embedded gravita-

tional search algorithm (CGSA-M), whale optimization algorithm (WOA), and GSA. We conduct the experiments using a problem dimension of  $D = 30/50/100$ , a population size of 100, and a maximum number of function computations of  $D * 10^4$  to ensure fair comparisons. The algorithms are run independently 51 times for each benchmark problem. The experiments are conducted on a computer with a 3.00 GHz Intel (R) Core i7-9700 processor, 8.00 GB of memory, and a 64-bit operating system.

In this study, we employ the Wilcoxon rank-sum test, a non-parametric statistical method, to compare the median differences between two independent samples. This test is particularly suitable for sample data that do not follow a normal distribution. We use this test to compare the performance of our proposed algorithm with existing methods on the CEC2017 problem set and 13 practical problems. To quantitatively assess the performance of the algorithms, we record the number of wins/ties/losses (W/T/L) on the specified problem set. A “win” indicates the number of problems where our proposed algorithm statistically outperforms

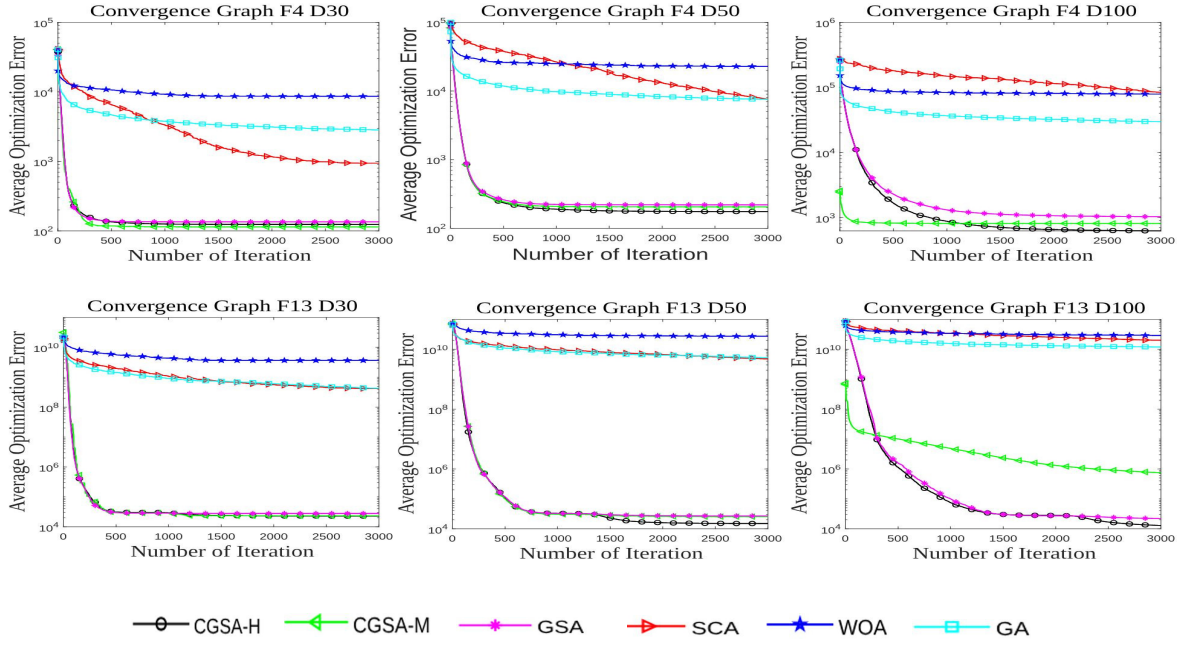


Fig. 5 Convergence graphs for functions F4 and F13 on IEEE CEC2017 test suite.

Table 1 Experimental results of CEC2017 on 30 dimensions.

	CGSA-H		CGSA-M		GSA		WOA		SCA		GA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
F1	1.934E+03	8.465E+02	<b>1.790E+03</b>	8.070E+02	2.037E+03	1.169E+03	3.669E+10	3.432E+09	1.234E+10	1.886E+09	1.083E+10	1.314E+09
F3	4.718E+04	1.095E+04	8.330E+04	5.770E+03	8.330E+04	5.395E+03	8.097E+04	4.197E+03	3.524E+04	6.480E+03	<b>3.334E+04</b>	3.576E+03
F4	<b>1.273E+02</b>	1.124E+01	1.420E+02	1.470E+01	1.351E+02	1.346E+01	8.553E+03	7.919E+02	9.429E+02	2.408E+02	2.230E+03	2.661E+02
F5	2.230E+02	1.735E+01	2.240E+02	1.930E+01	2.324E+02	1.613E+01	3.992E+02	1.649E+01	2.781E+02	2.057E+01	<b>2.070E+02</b>	2.160E+01
F6	<b>4.948E+01</b>	3.018E+00	5.010E+01	3.090E+00	5.038E+01	3.395E+00	7.865E+01	3.498E+00	4.956E+01	5.783E+00	5.681E+01	5.067E+00
F7	8.906E+01	1.017E+01	8.790E+01	1.160E+01	<b>8.725E+01</b>	1.123E+01	5.901E+02	2.891E+01	4.235E+02	3.849E+01	3.609E+02	5.452E+01
F8	1.529E+02	1.389E+01	1.520E+02	1.460E+01	1.540E+02	1.377E+01	3.229E+02	1.307E+01	2.501E+02	1.856E+01	<b>1.420E+02</b>	2.280E+01
F9	2.048E+03	2.696E+02	2.110E+03	4.140E+02	<b>1.986E+03</b>	3.064E+02	9.602E+03	8.083E+02	4.322E+03	9.179E+02	3.416E+03	4.639E+02
F10	3.739E+03	4.558E+02	<b>3.720E+03</b>	4.350E+02	3.918E+03	3.664E+02	7.134E+03	3.124E+02	7.216E+03	2.893E+02	5.140E+03	1.032E+03
F11	<b>1.769E+02</b>	5.566E+01	3.440E+02	9.780E+01	3.482E+02	8.194E+01	7.343E+03	1.436E+03	1.034E+03	4.362E+02	7.041E+02	7.847E+01
F12	<b>3.762E+06</b>	6.269E+06	1.530E+07	2.870E+07	1.149E+03	1.149E+06	2.256E+07	7.402E+09	1.394E+09	1.040E+09	1.593E+09	2.770E+08
F13	<b>2.094E+04</b>	5.095E+03	2.790E+04	5.450E+03	2.781E+04	4.890E+03	3.414E+09	1.225E+09	4.171E+08	1.851E+08	2.085E+08	1.136E+08
F14	<b>1.360E+03</b>	2.440E+03	4.340E+05	1.370E+05	4.758E+05	1.116E+05	3.438E+06	1.515E+06	1.374E+05	7.293E+04	1.620E+05	7.924E+04
F15	<b>3.785E+03</b>	1.513E+03	1.070E+04	2.010E+03	1.022E+04	1.873E+03	1.182E+08	5.077E+07	1.202E+07	9.820E+06	1.372E+04	5.093E+03
F16	<b>1.432E+03</b>	3.097E+02	1.560E+03	3.040E+02	1.610E+03	2.960E+02	3.847E+03	3.075E+02	2.001E+03	2.166E+02	2.376E+03	3.144E+02
F17	1.009E+03	2.781E+02	1.100E+03	2.230E+02	1.149E+03	1.885E+02	1.807E+03	2.202E+02	<b>6.989E+02</b>	1.654E+02	7.327E+02	1.907E+02
F18	<b>3.912E+04</b>	1.297E+04	3.290E+05	1.740E+05	3.346E+05	1.757E+05	1.965E+07	9.463E+06	3.332E+06	1.539E+06	5.081E+05	2.608E+05
F19	<b>4.019E+03</b>	1.977E+03	1.180E+04	5.100E+03	1.187E+04	4.158E+03	1.835E+08	5.326E+07	2.467E+07	1.334E+07	1.401E+06	8.631E+05
F20	8.776E+02	2.417E+02	1.030E+03	1.720E+02	1.005E+03	1.579E+02	9.791E+02	6.324E+02	9.947E+01	6.324E+02	<b>5.104E+02</b>	1.273E+02
F21	<b>4.492E+02</b>	2.216E+01	4.520E+02	2.130E+01	4.536E+02	1.685E+01	6.290E+02	2.466E+01	4.563E+02	1.681E+01	4.496E+02	2.848E+01
F22	3.880E+03	1.968E+03	3.680E+03	2.130E+03	3.922E+03	1.864E+03	5.607E+03	5.037E+02	5.876E+03	2.511E+03	<b>2.656E+03</b>	6.246E+02
F23	9.310E+02	1.727E+02	1.160E+03	1.360E+02	1.295E+03	1.224E+02	1.242E+03	6.722E+01	<b>6.845E+02</b>	2.393E+01	9.362E+02	6.880E+01
F24	8.380E+02	7.133E+01	8.600E+02	5.940E+01	8.676E+02	4.748E+01	1.311E+03	6.622E+01	<b>7.641E+02</b>	2.414E+01	1.068E+03	6.533E+01
F25	<b>4.253E+02</b>	1.248E+01	4.300E+02	1.360E+01	4.335E+02	1.024E+01	1.767E+03	1.340E+02	7.053E+02	7.408E+01	6.718E+02	1.879E+01
F26	3.712E+03	1.486E+03	<b>3.560E+03</b>	1.490E+03	4.076E+03	9.225E+02	7.079E+03	5.483E+02	4.333E+03	3.172E+02	5.238E+03	8.234E+02
F27	1.184E+03	3.236E+02	1.650E+03	3.950E+02	1.824E+03	2.493E+02	<b>5.000E+02</b>	7.668E-05	7.051E+02	4.123E+01	1.188E+03	1.040E+02
F28	<b>4.918E+02</b>	3.556E+01	5.220E+02	5.750E+01	5.312E+02	8.182E+01	5.000E+02	2.108E-02	1.019E+03	1.150E+02	1.228E+03	6.816E+01
F29	<b>1.731E+03</b>	1.933E+02	1.800E+03	2.160E+02	1.799E+03	2.336E+02	3.025E+03	3.176E+02	1.753E+03	2.522E+02	2.225E+03	3.241E+02
F30	<b>4.784E+04</b>	8.603E+03	2.000E+05	1.460E+05	1.600E+05	1.279E+05	5.286E+08	2.093E+08	6.672E+07	2.421E+07	1.393E+07	5.163E+06
W/T/L	-/-		16/13/0		15/14/0		26/2/1		19/4/6		20/3/6	

the comparison algorithm, a “tie” denotes the number of problems where there is no significant difference in performance between the two algorithms, and a “loss” refers to the number of problems where the proposed algorithm underperforms compared to the comparison algorithm. This method allow us to determine the relative efficacy of our proposed algorithm against competitive algorithms across multiple performance metrics. The results of the test reveal that our algorithm demonstrates superior performance on most of the problems, as reflected by the number of wins.

This statistical comparison provides a solid foundation for the performance evaluation of our research.

The results of the experiments are summarized in Tables 1, 2, and 3, which demonstrate that the proposed CGSA-H significantly outperforms its peers in terms of solution quality. It is also found to be more suitable for dealing with high-dimensional optimization problems. The convergence diagrams and box-whisker diagrams shown in Figs. 2 and 5 further demonstrate the algorithm’s fast convergence and search capabilities.



**Table 2** Experimental results of CEC2017 on 50 dimensions.

	CGSA-H		CGSA-M		GSA		WOA		SCA		GA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
F1	5.958E+02	9.928E+02	<b>5.386E+02</b>	9.972E+02	8.815E+02	1.162E+03	7.772E+10	4.570E+09	3.844E+10	5.663E+09	3.397E+10	2.001E+09
F3	1.120E+05	1.625E+04	1.701E+05	9.531E+03	1.682E+05	1.057E+04	1.847E+05	1.096E+04	1.006E+05	1.573E+04	<b>9.136E+04</b>	5.706E+03
F4	<b>1.874E+02</b>	6.095E+01	2.049E+02	7.095E+01	2.194E+02	7.399E+01	2.284E+04	2.788E+03	5.661E+03	1.354E+03	6.714E+03	4.266E+02
F5	<b>3.244E+02</b>	2.151E+01	3.311E+02	2.037E+01	3.259E+02	1.921E+01	6.599E+02	2.025E+01	5.511E+02	2.920E+01	3.462E+02	3.507E+01
F6	5.682E+01	2.655E+00	<b>5.611E+01</b>	2.452E+00	5.680E+01	2.968E+00	9.548E+01	2.817E+00	6.866E+01	4.962E+00	7.026E+01	5.403E+00
F7	2.275E+02	2.789E+01	<b>2.218E+02</b>	2.451E+01	2.225E+02	2.667E+01	1.098E+03	4.251E+01	9.066E+02	6.448E+01	7.761E+02	5.836E+01
F8	3.542E+02	2.009E+01	3.528E+02	2.332E+01	<b>3.497E+02</b>	2.121E+01	6.449E+02	1.958E+01	5.512E+02	3.043E+01	3.670E+02	3.806E+01
F9	8.350E+03	5.141E+02	<b>8.288E+03</b>	5.758E+02	8.331E+03	6.923E+02	3.471E+04	1.868E+03	2.079E+04	3.709E+03	1.294E+04	1.661E+03
F10	6.872E+03	5.844E+02	6.873E+03	5.905E+02	<b>6.823E+03</b>	5.875E+02	1.349E+04	3.817E+02	1.334E+04	4.102E+02	9.409E+03	1.527E+03
F11	<b>5.617E+02</b>	1.993E+02	1.246E+03	2.816E+02	1.238E+03	2.332E+02	1.865E+04	1.609E+03	4.802E+03	1.210E+03	4.666E+03	4.321E+02
F12	<b>1.082E+06</b>	3.274E+05	1.884E+06	4.695E+05	1.998E+06	5.520E+05	5.481E+10	6.137E+09	1.149E+10	2.630E+09	1.587E+10	2.002E+09
F13	<b>1.461E+04</b>	1.832E+03	2.551E+04	3.866E+03	2.673E+04	3.587E+03	2.747E+10	4.359E+09	2.661E+09	8.451E+08	4.505E+09	8.205E+08
F14	<b>3.076E+04</b>	4.463E+04	2.819E+05	9.584E+04	3.366E+05	2.260E+05	3.415E+07	1.241E+07	1.936E+06	9.174E+05	2.898E+06	1.210E+06
F15	<b>1.272E+04</b>	3.324E+03	1.578E+04	4.073E+03	1.527E+04	3.252E+03	3.081E+09	8.099E+08	3.400E+08	1.342E+08	2.446E+07	1.126E+07
F16	2.027E+03	2.768E+02	<b>1.980E+03</b>	3.725E+02	2.014E+03	3.121E+02	6.770E+03	6.058E+02	3.789E+03	3.437E+02	4.246E+03	5.540E+02
F17	1.857E+03	3.274E+02	1.896E+03	2.437E+02	<b>1.840E+03</b>	3.268E+02	4.597E+03	4.833E+02	2.575E+02	2.438E+02	1.993E+03	3.228E+02
F18	<b>3.104E+05</b>	2.523E+05	1.316E+06	7.176E+05	1.386E+06	9.503E+05	8.301E+07	3.040E+07	1.422E+07	6.718E+06	1.135E+07	1.804E+06
F19	<b>1.555E+04</b>	2.914E+03	2.439E+04	7.368E+03	2.491E+04	5.392E+03	1.161E+09	2.489E+08	2.199E+08	1.058E+08	4.261E+06	2.205E+06
F20	1.562E+03	2.510E+02	1.608E+03	2.763E+02	1.557E+03	2.900E+02	1.843E+03	1.427E+02	1.773E+03	1.885E+02	<b>1.135E+03</b>	2.394E+02
F21	6.332E+02	3.591E+01	<b>6.303E+02</b>	2.795E+01	6.362E+02	3.206E+01	1.116E+03	4.036E+01	7.620E+02	3.225E+01	7.918E+02	5.092E+01
F22	9.369E+03	5.288E+02	<b>9.167E+03</b>	5.935E+02	9.190E+03	5.647E+02	1.406E+04	4.111E+02	1.366E+04	3.714E+02	1.089E+04	1.156E+03
F23	1.755E+03	2.286E+02	2.039E+03	1.427E+02	2.049E+03	1.428E+02	2.054E+03	7.684E+01	<b>1.208E+03</b>	5.621E+01	1.731E+03	9.270E+01
F24	<b>1.170E+03</b>	1.422E+02	1.353E+03	5.719E+01	1.353E+03	6.518E+01	2.246E+03	8.761E+01	1.263E+03	5.272E+01	1.971E+03	1.243E+02
F25	<b>6.761E+02</b>	4.122E+01	7.208E+02	8.416E+01	7.376E+02	8.176E+01	8.008E+03	3.976E+02	3.426E+03	5.342E+02	3.246E+03	1.594E+02
F26	<b>2.693E+03</b>	2.541E+03	4.065E+03	2.455E+03	3.045E+03	2.129E+03	1.211E+04	3.100E+02	9.058E+03	6.189E+02	9.722E+03	7.148E+02
F27	2.893E+03	6.500E+02	3.704E+03	4.835E+02	3.611E+03	4.203E+02	<b>5.000E+02</b>	7.982E-05	1.672E+03	1.656E+02	3.035E+03	2.462E+02
F28	6.282E+02	5.471E+01	7.228E+02	1.121E+02	6.846E+02	9.777E+01	<b>5.000E+02</b>	4.997E-03	3.533E+03	4.530E+02	3.615E+03	1.526E+02
F29	<b>2.465E+03</b>	2.748E+02	2.511E+03	2.747E+02	2.651E+03	3.459E+02	1.960E+04	7.334E+03	4.261E+03	5.919E+02	5.893E+03	7.698E+02
F30	<b>1.487E+07</b>	3.292E+06	4.323E+07	5.363E+06	4.118E+07	5.548E+06	3.372E+09	7.337E+08	5.830E+08	1.837E+08	2.950E+08	2.790E+07
W/T/L	--/--		15/14/0		15/14/0		27/0/2		26/0/3		24/3/2	

**Table 3** Experimental results of CEC2017 on 100 dimensions.

	CGSA-H		CGSA-M		GSA		WOA		SCA		GA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
F1	4.117E+03	2.341E+03	3.747E+03	2.599E+03	<b>3.702E+03</b>	2.589E+03	2.148E+11	5.578E+09	1.531E+11	9.584E+09	1.163E+11	3.478E+09
F3	2.711E+05	1.930E+04	3.428E+05	1.233E+04	3.409E+05	1.108E+04	3.372E+05	1.233E+04	2.861E+05	1.995E+04	<b>2.326E+05</b>	7.073E+03
F4	<b>6.245E+02</b>	1.635E+02	8.699E+02	2.918E+02	1.024E+03	6.354E+02	7.761E+04	6.263E+03	2.588E+04	3.423E+03	2.407E+04	1.028E+03
F5	<b>8.027E+02</b>	3.135E+01	8.115E+02	2.977E+01	8.039E+02	3.103E+01	1.555E+03	2.970E+01	1.351E+03	5.763E+01	9.552E+02	7.774E+01
F6	<b>5.670E+01</b>	1.325E+00	5.762E+01	1.560E+00	5.723E+01	1.642E+00	1.061E+02	2.538E+00	8.885E+01	3.950E+00	7.724E+01	5.075E+00
F7	9.256E+02	6.055E+01	<b>8.917E+02</b>	7.581E+01	9.156E+02	7.493E+01	2.932E+03	9.210E+01	2.687E+03	1.203E+02	2.289E+03	1.309E+02
F8	9.572E+02	3.471E+01	<b>9.492E+02</b>	3.998E+01	9.555E+02	4.313E+01	1.656E+03	3.318E+01	1.405E+03	5.557E+01	1.102E+03	6.846E+01
F9	1.919E+04	6.342E+02	<b>1.910E+04</b>	8.600E+02	1.911E+04	6.847E+02	7.229E+04	3.918E+03	6.669E+04	6.906E+03	3.097E+04	3.627E+03
F10	1.463E+04	9.354E+02	<b>1.435E+04</b>	7.362E+02	1.455E+04	9.180E+02	3.013E+04	4.681E+02	3.023E+04	4.870E+02	2.375E+04	3.163E+03
F11	<b>1.806E+04</b>	4.682E+03	3.815E+04	5.249E+03	3.742E+04	9.452E+03	2.463E+05	2.718E+04	6.980E+04	1.102E+04	6.625E+04	4.341E+03
F12	<b>1.174E+06</b>	1.201E+06	1.896E+07	5.244E+07	1.541E+07	4.111E+07	1.480E+11	8.740E+09	5.331E+10	7.136E+09	5.835E+10	3.939E+09
F13	<b>1.103E+04</b>	2.158E+03	2.038E+04	2.772E+03	2.021E+04	2.757E+04	2.765E+10	3.391E+09	8.224E+09	1.303E+09	8.830E+09	7.847E+08
F14	<b>1.802E+05</b>	1.021E+05	5.519E+05	8.836E+04	5.617E+05	9.870E+04	4.181E+07	8.942E+06	1.801E+07	6.980E+06	4.464E+06	3.213E+05
F15	<b>1.951E+03</b>	8.257E+02	6.905E+03	1.461E+03	6.743E+03	1.245E+03	1.180E+10	1.860E+09	2.566E+09	6.868E+08	2.055E+09	4.190E+08
F16	5.395E+03	4.778E+02	5.379E+03	5.745E+02	<b>5.292E+03</b>	5.537E+02	1.794E+04	1.035E+03	1.081E+04	6.022E+02	1.269E+04	7.150E+02
F17	3.443E+03	5.040E+02	3.471E+03	4.910E+02	<b>3.376E+03</b>	4.835E+02	1.643E+06	6.736E+05	9.135E+03	1.205E+03	1.441E+04	2.547E+03
F18	<b>2.229E+05</b>	6.615E+04	4.871E+05	1.006E+05	4.823E+05	7.493E+04	8.863E+07	2.761E+07	3.340E+07	1.146E+07	2.928E+06	2.910E+05
F19	<b>1.833E+03</b>	1.253E+03	5.270E+03	1.419E+03	4.843E+03	1.245E+03	1.272E+10	1.601E+09	2.111E+09	6.412E+08	2.197E+09	3.446E+08
F20	<b>4.176E+03</b>	4.900E+02	4.246E+03	4.954E+02	4.255E+03	4.756E+02	5.557E+03	2.399E+02	5.116E+03	2.624E+02	3.442E+03	5.150E+02
F21	<b>1.346E+03</b>	6.686E+01	1.381E+03	7.776E+01	1.384E+03	7.623E+01	2.725E+03	9.157E+01	1.778E+03	6.635E+01	2.164E+03	1.107E+02
F22	<b>1.846E+04</b>	7.223E+02	1.849E+04	9.369E+02	1.859E+04	9.135E+02	3.173E+04	5.699E+02	3.130E+04	6.217E+03	2.457E+04	2.771E+03
F23	3.479E+03	6.192E+02	4.375E+03	2.859E+02	4.326E+03	2.674E+02	3.979E+03	1.224E+02	<b>2.468E+03</b>	8.346E+01	6.327E+03	1.794E+02
F24	<b>2.541E+03</b>	2.128E+02	2.812E+03	2.652E+02	2.767E+03	1.742E+02	6.801E+03	2.334E+02	3.899E+03	1.741E+02	6.903E+03	3.172E+02
F25	<b>1.317E+03</b>	8.443E+01	1.629E+03	1.896E+02	1.675E+03	1.919E+02	1.829E+04	9.336E+02	1.145E+04	1.529E+03	7.812E+03	2.205E+02
F26	<b>1.591E+04</b>	5.838E+03	1.610E+04	4.916E+03	1.694E+04	4.268E+03	3.959E+04	1.064E+03	2.920E+04	1.471E+03	3.267E+04	1.025E+03
F27	5.481E+03	1.108E+03	7.145E+03	7.720E+02	6.765E+03	9.450E+02	<b>5.000E+02</b>	9.353E-05	4.022E+03	3.771E+02	7.116E+03	5.249E+02
F28	1.057E+03	1.583E+02	1.428E+03	3.439E+02	1.508E+03	3.635E+02	<b>5.000E+02</b>	1.377E-04	1.547E+04	1.586E+03	1.348E+04	4.216E+02
F29	<b>5.434E+03</b>	4.395E+02	5.777E+03	4.239E+02	5.749E+03	4.401E+02	1.441E+05	5.045E+04	1.258E+04	1.250E+03	1.555E+04	1.349E+03
F30	<b>1.302E+05</b>	2.547E+05	2.624E+06	9.390E+06	3.513E+06	1.555E+07	2.444E+10	3.621E+09	5.490E+09	1.001E+09	8.937E+09	9.451E+08
W/T/L	--/--		17/11/0		18/11/0		27/0/2		27/0/2		27/0/2	

In Table 4, we detail the specific content of thirteen real-world problems. Table 5 presents the comparative results of CGSA-H against other algorithms across these thirteen real-world scenarios. The results clearly demonstrate that CGSA-H exhibits excellent performance in practical applications.

Fig. 3 illustrates the search trajectory graph of CGSA-H on F3 and F9. The individual trajectories on the function graph as the number of iterations increases, using the IEEE CEC2017 as the test function set, demonstrate CGSA-H's ability to overcome local optima. Images are recorded for 2 and 200 iterations in F3 and F9, respectively. We observe

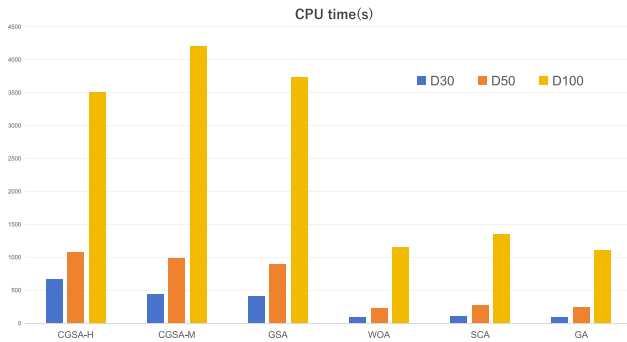
that the populations narrowed the search range and eventually converged to the minimum range in both F3 and F9, indicating CGSA-H's strong exploitation potential.

Fig. 4 shows that CGSA-H displays significant population diversity during the initial stages of the search process, aiding the algorithm in breaking free from local optima and averting premature convergence. From F11 and F19, it can be seen that from the beginning of the iteration, the population diversity decreases rapidly, and in the middle of the iteration, the population diversity maintains a stable value, thus proving that the algorithm has the ability to explore along with the ability to exploit. From F25, it can be seen that the



**Table 4** Descriptions of 13 real-world problems.

Function	Description
$\mathcal{P}_1$	A parameter estimation for frequency adjustment sound waves
$\mathcal{P}_2$	A Lennard-Jones potential energy problem
$\mathcal{P}_3 - \mathcal{P}_4$	Two optimization problems for the Tersoff potential function
$\mathcal{P}_5$	A spread spectrum radar polly phase code design problem
$\mathcal{P}_6$	A cost optimization problem for transmission network expansion planning
$\mathcal{P}_7$	A circular shaped antenna array design problem
$\mathcal{P}_8 - \mathcal{P}_{12}$	Five static system economic dispatch problems
$\mathcal{P}_{13}$	A hydrothermal term scheduling problems

**Fig. 6** The bar graph illustrates the CPU run times for a single execution in 30, 50, and 100 dimensions of all tested algorithms on the IEEE CEC2017 functions.

population diversity has been high from the beginning to the end of the iteration, thus proving the algorithm's strong exploration capability.

Fig. 6 illustrates the computational time required for running each algorithm once on 29 problems in the IEEE CEC2017 benchmark. Overall, CGSA-H exhibits a 6.50% reduction in computation time and a 55.17% increase in performance compared to CGSA-M. This validates the effectiveness of our algorithmic enhancements, particularly in higher dimensions, where CGSA-H not only performs better but also requires less computational time in comparison with other GSA variants.

## 6. Conclusions

To address the issue of CGSA-M falling into local optima, we propose a hierarchical multi-chaotic embedded gravitational search algorithm (CGSA-H). Two additional hierarchical components are added to the original CGSA-M, resulting in a four-layer hierarchical population structure. The most valuable individual layer improves the population's interaction during the search process and records the optimal value of the main population. Experimental results demonstrate that the proposed hierarchical population structure significantly enhances the accuracy of CGSA-M, and the improved CGSA-H outperforms comparable algorithms in terms of solution quality.

To evaluate the effectiveness of CGSA-H, we compare it with other well-known heuristics. The population diver-

sity plot of CGSA-H indicates that the algorithm possesses exploration ability while maintaining strong exploitation ability, demonstrating the efficacy of our modified scheme. In conclusion, CGSA-H is an algorithm with robust performance improvements. Future work may focus on the following important studies: 1) further improving the performance of the CGSA-H algorithm, 2) applying the population structure scheme to additional MHAs, and 3) conducting performance studies on practical applications, such as training neural networks and solving new energy optimization problems.

## Acknowledgment

This research was partially supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP22H03643.

## References

- [1] A. Slowik and H. Kwasnicka, "Evolutionary algorithms and their applications to engineering problems," *Neural Computing and Applications*, vol.32, pp.12363–12379, 2020.
- [2] P.A. Vikhar, "Evolutionary algorithms: A critical review and its future prospects," 2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC), pp.261–265, IEEE, 2016.
- [3] M.G. Castillo Tapia and C.A. Coello Coello, "Applications of multi-objective evolutionary algorithms in economics and finance: A survey," 2007 IEEE Congress on Evolutionary Computation, pp.532–539, 2007.
- [4] J. Tang, G. Liu, and Q. Pan, "A Review on Representative Swarm Intelligence Algorithms for Solving Optimization Problems: Applications and Trends," *IEEE/CAA Journal of Automatica Sinica*, vol.8, no.10, pp.1627–1643, 2021.
- [5] Z.H. Zhan, L. Shi, K.C. Tan, and J. Zhang, "A survey on evolutionary computation for complex continuous optimization," *Artificial Intelligence Review*, vol.55, no.1, pp.59–110, 2022.
- [6] S. Forrest, "Genetic Algorithms," *ACM Computing Surveys (CSUR)*, vol.28, no.1, pp.77–80, 1996.
- [7] H. Yang, S. Gao, R.L. Wang, and Y. Todo, "A Ladder Spherical Evolution Search Algorithm," *IEICE TRANSACTIONS on Information and Systems*, vol.104, no.3, pp.461–464, 2021.
- [8] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol.1, no.4, pp.28–39, 2006.
- [9] S. Gao, Y. Wang, J. Cheng, Y. Inazumi, and Z. Tang, "Ant colony optimization with clustering for solving the dynamic location routing problem," *Applied Mathematics and Computation*, vol.285, pp.149–173, 2016.
- [10] H. Yang, Y. Yu, J. Cheng, Z. Lei, Z. Cai, Z. Zhang, and S. Gao, "An intelligent metaphor-free spatial information sampling algorithm for balancing exploitation and exploration," *Knowledge-Based Systems*, vol.250, p.109081, 2022.
- [11] Z. Wang, S. Gao, J. Wang, H. Yang, and Y. Todo, "A Dendritic Neuron Model with Adaptive Synapses Trained by Differential Evolution Algorithm," *Computational Intelligence and Neuroscience*, vol.2020, 2020.
- [12] A. Tzaniotis and G. Dounias, "Nature inspired optimization algorithms or simply variations of metaheuristics?," *Artificial Intelligence Review*, vol.54, no.3, pp.1841–1862, 2021.
- [13] Y. Wang, Y. Yu, S. Cao, X. Zhang, and S. Gao, "A review of applications of artificial intelligent algorithms in wind farms," *Artificial Intelligence Review*, vol.53, no.5, pp.3447–3500, 2020.

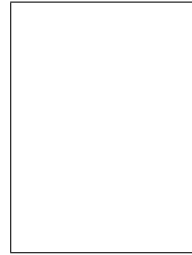
**Table 5** Experimental results on thirteen real-world problems.

	CGSA-H		CGSA-M		GSA		WOA		SCA		GA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
$P_1$	2.290E+01	2.950E+00	2.525E+01	1.323E+00	2.583E+01	1.556E+00	1.946E+01	4.747E+00	<b>1.617E+01</b>	4.751E+00	2.451E+01	1.498E+00
$P_2$	-1.847E+01	3.465E+00	-1.838E+01	3.772E+00	-1.855E+01	3.930E+00	-1.952E+01	4.859E+00	-1.192E+01	9.773E-01	<b>-2.138E+01</b>	2.263E+00
$P_3$	-3.012E+01	3.524E+00	-2.995E+01	3.382E+00	<b>-3.028E+01</b>	3.806E+00	-2.771E+01	4.041E+00	-2.084E+01	1.067E+00	-1.889E+01	1.469E+00
$P_4$	-2.039E+01	2.597E+00	-1.846E+01	2.765E+00	-1.845E+01	2.951E+00	<b>-2.060E+01</b>	3.137E+00	-1.545E+01	1.346E+00	-9.451E+00	1.724E+00
$P_5$	8.968E-01	2.014E-01	9.808E-01	1.814E-01	9.193E-01	2.275E-01	1.893E+00	2.000E-01	<b>1.693E+00</b>	1.157E-01	1.701E+00	1.353E-01
$P_6$	<b>2.287E+02</b>	1.194E+01	2.772E+02	4.405E+01	2.830E+02	7.052E+01	2.826E+02	2.922E+01	2.355E+02	1.288E+01	2.353E+02	2.510E+01
$P_7$	-1.211E+01	8.098E-01	-1.129E+01	4.344E-01	-1.126E+01	5.182E-01	-1.101E+01	1.233E+00	<b>-1.242E+01</b>	1.259E+00	-8.161E+00	5.365E-01
$P_8$	1.775E+04	5.502E+03	9.092E+04	4.944E+04	1.203E+05	5.726E+04	<b>1.554E+04</b>	6.768E+01	1.602E+04	4.527E+02	3.018E+04	1.985E+04
$P_9$	1.922E+04	1.145E+02	1.917E+04	1.049E+02	1.920E+04	1.150E+02	1.930E+04	2.237E+02	1.959E+04	2.928E+02	<b>1.905E+04</b>	2.033E+02
$P_{10}$	1.444E+05	8.159E+04	3.265E+05	4.379E+04	3.324E+05	3.840E+04	<b>4.172E+04</b>	2.510E+04	1.617E+05	4.172E+04	3.697E+05	5.667E+04
$P_{11}$	<b>1.453E+05</b>	2.760E+03	1.458E+05	1.936E+03	1.464E+05	2.182E+03	1.486E+05	6.900E+03	3.789E+06	3.442E+06	7.272E+07	1.642E+07
$P_{12}$	2.976E+06	7.247E+05	2.743E+06	5.348E+05	<b>2.720E+06</b>	6.020E+05	1.079E+10	1.951E+09	5.777E+09	8.298E+08	1.565E+10	2.969E+09
$P_{13}$	9.547E+05	3.911E+04	<b>9.418E+05</b>	1.250E+03	9.421E+05	1.353E+03	2.777E+06	2.211E+06	7.618E+07	5.910E+06	1.600E+06	3.798E+05
W/T/L	-	-	8/5/0	-	7/5/1	-	8/2/3	-	9/3/1	-	9/2/2	-

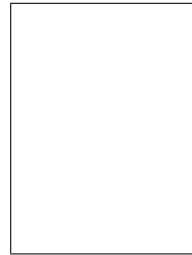
- [14] Z. Wu, Q. Xu, G. Ni, and G. Yu, "The study of genetic information flux network properties in genetic algorithms," *International Journal of Modern Physics C*, vol.26, no.07, p.1550076, 2015.
- [15] Y.J. Gong, W.N. Chen, Z.H. Zhan, J. Zhang, Y. Li, Q. Zhang, and J.J. Li, "Distributed evolutionary algorithms and their models: A survey of the state-of-the-art," *Applied Soft Computing*, vol.34, pp.286–300, 2015.
- [16] E. Alba and B. Dorronsoro, "The exploration/exploitation tradeoff in dynamic cellular genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol.9, no.2, pp.126–142, 2005.
- [17] Y. Shi, H. Liu, L. Gao, and G. Zhang, "Cellular particle swarm optimization," *Information Sciences*, vol.181, no.20, pp.4460–4493, 2011.
- [18] C.C. Lai and C.Y. Chang, "A hierarchical evolutionary algorithm for automatic medical image segmentation," *Expert Systems with Applications*, vol.36, no.1, pp.248–259, 2009.
- [19] G. Chen, Y. Li, K. Zhang, X. Xue, J. Wang, Q. Luo, C. Yao, and J. Yao, "Efficient hierarchical surrogate-assisted differential evolution for high-dimensional expensive optimization," *Information Sciences*, vol.542, pp.228–246, 2021.
- [20] Y. Wang, Y. Yu, S. Gao, H. Pan, and G. Yang, "A hierarchical gravitational search algorithm with an effective gravitational constant," *Swarm and Evolutionary Computation*, vol.46, pp.118–139, 2019.
- [21] Y. Wang, S. Gao, M. Zhou, and Y. Yu, "A multi-layered gravitational search algorithm for function optimization and real-world problems," *IEEE/CAA Journal of Automatica Sinica*, vol.8, no.1, pp.94–109, 2021.
- [22] H. Du, X. Wu, and J. Zhuang, "Small-World Optimization Algorithm for Function Optimization," *Advances in Natural Computation: Second International Conference, ICNC 2006, Xi'an, China, September 24–28, 2006. Proceedings, Part II 2*, pp.264–273, Springer, 2006.
- [23] M. Vora and T. Mirmalinee, "Dynamic small world particle swarm optimizer for function optimization," *Natural Computing*, vol.17, pp.901–917, 2018.
- [24] H. Dai, S. Gao, Y. Yang, and Z. Tang, "Effects of "rich-gets-richer" rule on small-world networks," *Neurocomputing*, vol.73, no.10–12, pp.2286–2289, 2010.
- [25] C. Zhang and Z. Yi, "Scale-free fully informed particle swarm optimization algorithm," *Information Sciences*, vol.181, no.20, pp.4550–4568, 2011.
- [26] H. Li, H. Yang, B. Zhang, H. Zhang, and S. Gao, "Swarm Exploration Mechanism-Based Distributed Water Wave Optimization," *International Journal of Computational Intelligence Systems*, vol.16, no.1, pp.1–26, 2023.
- [27] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Information Sciences*, vol.179, no.13, pp.2232–2248, 2009.
- [28] W.S. Tan, M.Y. Hassan, H.A. Rahman, M.P. Abdullah, and F. Hussin, "Multi-distributed generation planning using hybrid particle swarm optimisation-gravitational search algorithm including voltage rise issue," *IET Generation, Transmission & Distribution*, vol.7, no.9, pp.929–942, 2013.
- [29] Z.k. Feng, S. Liu, W.j. Niu, S.s. Li, H.j. Wu, and J.y. Wang, "Ecological operation of cascade hydropower reservoirs by elite-guide gravitational search algorithm with lévy flight local search and mutation," *Journal of Hydrology*, vol.581, p.124425, 2020.
- [30] E. Rashedi, E. Rashedi, and H. Nezamabadi-pour, "A comprehensive survey on gravitational search algorithm," *Swarm and Evolutionary Computation*, vol.41, pp.141–158, 2018.
- [31] G. Sahoo et al., "A Review on Gravitational Search Algorithm and its Applications to Data Clustering & Classification," *IJ Intelligent Systems and Applications*, vol.6, pp.79–93, 2014.
- [32] N. Siddiquea and H. Adelib, "Applications of gravitational search algorithm in engineering," 2016.
- [33] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Filter modeling using gravitational search algorithm," *Engineering Applications of Artificial Intelligence*, vol.24, no.1, pp.117–122, 2011.
- [34] A. Hatamlou, S. Abdullah, and H. Nezamabadi-Pour, "Application of Gravitational Search Algorithm on Data Clustering," *Rough Sets and Knowledge Technology: 6th International Conference, RSKT 2011, Banff, Canada, October 9–12, 2011. Proceedings 6*, pp.337–346, Springer, 2011.
- [35] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "BGSA: binary gravitational search algorithm," *Natural Computing*, vol.9, pp.727–745, 2010.
- [36] Y. Wang, S. Gao, Y. Yu, Z. Cai, and Z. Wang, "A gravitational search algorithm with hierarchy and distributed framework," *Knowledge-Based Systems*, vol.218, p.106877, 2021.
- [37] J. Gleick, *Chaos: Making a New Science*, Penguin, 2008.
- [38] M.S. Tavazoei and M. Haeri, "Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms," *Applied Mathematics and Computation*, vol.187, no.2, pp.1076–1085, 2007.
- [39] T. Xiang, X. Liao, and K. Wong, "An improved particle swarm optimization algorithm combined with piecewise linear chaotic map," *Applied Mathematics and Computation*, vol.190, no.2, pp.1637–1645, 2007.
- [40] S. Saremi, S. Mirjalili, and A. Lewis, "Biogeography-based optimisation with chaos," *Neural Computing and Applications*, vol.25, pp.1077–1097, 2014.
- [41] A.A. Heidari, R. Ali Abbaspour, and A. Rezaee Jordehi, "An efficient chaotic water cycle algorithm for optimization tasks," *Neural Computing and Applications*, vol.28, pp.57–85, 2017.
- [42] M. Mitić, N. Vuković, M. Petrović, and Z. Miljković, "Chaotic fruit fly optimization algorithm," *Knowledge-Based Systems*, vol.89, pp.446–458, 2015.
- [43] G.G. Wang, L. Guo, A.H. Gandomi, G.S. Hao, and H. Wang, "Chaotic Krill Herd algorithm," *Information Sciences*, vol.274, pp.17–34, 2014.
- [44] A.H. Gandomi and X.S. Yang, "Chaotic bat algorithm," *Journal of Computational Science*, vol.5, no.2, pp.224–232, 2014.
- [45] D. Jia, G. Zheng, and M. Khurram Khan, "An effective memetic differential evolution algorithm based on chaotic local search," *In-*

formation Sciences, vol.181, no.15, pp.3175–3187, 2011.

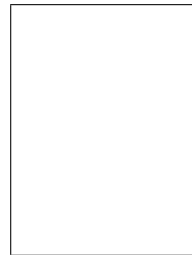
- [46] R. Arul, G. Ravi, and S. Velusami, "Chaotic self-adaptive differential harmony search algorithm based dynamic economic dispatch," *International Journal of Electrical Power Energy Systems*, vol.50, pp.85–96, 2013.
- [47] A. Kazem, E. Sharifi, F.K. Hussain, M. Saberi, and O.K. Hussain, "Support vector regression with chaos-based firefly algorithm for stock market price forecasting," *Applied Soft Computing*, vol.13, no.2, pp.947–958, 2013.
- [48] Y. Li, Q. Wen, L. Li, and H. Peng, "Hybrid chaotic ant swarm optimization," *Chaos, Solitons Fractals*, vol.42, no.2, pp.880–889, 2009.
- [49] S. Talatahari, B. Farahmand Azar, R. Sheikholeslami, and A. Gandomi, "Imperialist competitive algorithm combined with chaos for global optimization," *Communications in Nonlinear Science and Numerical Simulation*, vol.17, no.3, pp.1312–1319, 2012.
- [50] S. Gao, C. Vairappan, Y. Wang, Q. Cao, and Z. Tang, "Gravitational search algorithm combined with chaos for unconstrained numerical optimization," *Applied Mathematics and Computation*, vol.231, pp.48–62, 2014.



**Ji Tianyi YU** received his B.E. degree in Computer Information Engineering from Changzhou Institute of Technology, Changzhou, China in 2021. He is currently pursuing the M.E. degree at University of Toyama, Toyama, Japan. His current research interests are computational intelligence.



**Zhenyu LEI** received the Ph.D. degree in Science and Engineering from the University of Toyama, Toyama, Japan, in 2023. He is currently an Assistant Professor with the Faculty of Engineering, University of Toyama, Japan. His current research interests include evolutionary computation, machine learning, and neural network for real-world applications and optimization problems.



**Shangce GAO** received his Ph.D. degree in Innovative Life Science from University of Toyama, Toyama, Japan in 2011. He is currently a Professor with the Faculty of Engineering, University of Toyama, Japan. His current research interests include nature-inspired technologies. His research has led to over 100 publications in top venues. He serves as an Associate Editor for many international journals such as *IEEE Transactions on Neural Networks and Learning Systems*, and *IEEE/CAA Journal of*



**Shibo DONG** received his B.E. degree from Institute of Technologists, Saitama, Japan in 2022. He is currently pursuing his M.E. degree at the University of Toyama, Toyama, Japan. His current research interests are in computational intelligence.

*Automatica Sinica.*



**Haotian LI** received his M.E. degree in Intellectual Information Engineering from University of Toyama, Japan in 2023. He is currently pursuing the Ph.D. degree at University of Toyama, Toyama, Japan. His current research interests are computational intelligence.



**Yifei YANG** received his Ph.D. degree in Intellectual Information Engineering from the University of Toyama, Toyama, Japan, in 2024. He is currently working as an assistant professor at Graduate School of Science and Technology in Hirosaki University. His current research interests include complex system, meta-heuristic algorithms, large-scale optimization, new energy optimization problems, and artificial neural networks.