IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 21, 2021, accepted May 20, 2021, date of publication May 24, 2021, date of current version June 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3083220

Comparative Study on Single and Multiple Chaotic Maps Incorporated Grey Wolf Optimization Algorithms

ZHE XU[1], HAICHUAN YANG[2], JIAYI LI[2], XINGYI ZHANG[3], BO LU[4], AND SHANGCE GAO[2], (Senior Member, IEEE)

[1]School of Computer Information and Engineering, Changzhou Institute of Technology, Changzhou 213032, China
[2]Faculty of Engineering, University of Toyama, Toyama 930-8555, Japan
[3]Shanghai General Hospital affiliated to Shanghai Jiaotong University, Shanghai 200080, China
[4]Faculty of Engineering, Shanghai Normal University Tianhua College, Shanghai 201815, China

Corresponding authors: Xingyi Zhang (zhangxingyi@msn.com), Bo Lu (lb2364@sthu.edu.cn), and Shangce Gao (gaosc@eng.u-toyama.ac.jp)

This work was supported in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 19KJB520003, in part by the Natural Science Foundation of Shanghai under Grant 19ZR1402000, in part by the Jiangsu Province 333 Project under Grant BRA2020152, in part by the Heilongjiang Provincial Natural Science Foundation of China under Grant QC2018080, in part by the Humanities and Social Science Fund of Ministry of Education of China under Grant 19YJCZH120, in part by the National Natural Science Foundation (NNSF) of China under Grant 61901063, and in part by the Science and Technology Plan Project of Changzhou under Grant CE20205042.

ABSTRACT As a meta-heuristic algorithm that simulates the intelligence of gray wolves, grey wolf optimizer (GWO) has a wide range of applications in practical problems. As a kind of local search, chaotic local search (CLS) has a strong ability to get rid of the local optimum due to its integration of chaotic maps. To enhance GWO, CLS is always incorporated into GWO to increase its population diversity and accelerate algorithm's convergence. However, it is still unclear that how may chaotic maps should be used in CLS and how to embed them into GWO. To address these challenging issues, this paper studies both single and multiple chaotic maps incorporated GWOs. Extensive comparative experiments are conducted based on IEEE Congress on Evolutionary Computation (CEC) benchmark test suit. The results show that CLS incorporated GWOs generally perform better than the original GWO, suggesting the effectiveness of such hybridization. Moreover, a remarkable finding of this work is that the piecewise linear chaotic map (PWLCM) and Gaussian map have the most potential to improve the search performance of GWO. Additionally, CLS incorporated GWOs also perform significantly better than some other state-of-the-art meta-heuristic algorithms. This study not only gives more insights into the mechanism of how CLS makes influence on GWO, but also finds that the most suitable choice of chaotic map for it.

INDEX TERMS Computational intelligence, soft computing, chaotic local search, optimization algorithms, grey wolf optimizer, meta-heuristics.

I. INTRODUCTION

Meta-heuristic algorithms (MHAs) have received great interests during the past several decades [1], and dozens of meta-heuristics have been proposed in the literature [2]. Typically, a meta-heuristic algorithm denotes a generalized formulation of heuristic methods that aim to solve a variety of optimization problems. Based on the perspective of metaphors by which these meta-heuristics are motivated, MHA can be classified into bio-inspired, physics-inspired, sociology-inspired, and other algorithms [3]. Representative bio-inspired algorithms include genetic algorithms [4], evolutionary strategies [5], differential evolution (DE) [6]–[8], spherical evolution [9], artificial immune algorithms [10], particle swarm optimization (PSO) [11], ant colony optimization [12], etc. Physics-inspired algorithms consist of simulated annealing [13], gravitational search algorithm [14], and quantum computing [15], while sociology-inspired ones usually denote imperialist competitive algorithm [16], brain storm optimization [17], culture algorithm [18], memetic algorithms [19], and so on. More importantly, these MHAs have been widely applied on various practical problems,

The associate editor coordinating the review of this manuscript and approving it for publication was Baker Mohammad.

from engineering [20], [21] to bio-informatics [22], [23], and achieved great successes in comparison with traditional mathematical analysis methods as they can obtain an acceptable solution with reasonable computational burden [24]–[28].

Despite some criticisms that new MHAs based on more metaphors but without essential differences between existing ones are no longer considered as significant contributions to the community [29], the improvements of MHAs are still of great importance as they indeed provide more accuracy and fruitful solutions for real-world applications [30], [31]. Recently, the developments of MHAs are usually realized from the following aspects: 1) self-adaption of hyper-parameters, 2) population structure evolution, 3) balance of exploitation and exploration, 4) theoretical analysis of the search dynamics, and 5) memetic computing manner.

As most of MHAs have some hyper-parameters needed to be adjusted [32], tremendous efforts have been done to make these parameters self-adaptive [33]–[35]. For a special meta-heuristic algorithm, e.g., differential evolution [36], parameter (self-adaptive) control can significant enhance the search performance of the algorithm, from population size [37], scale factor [38], crossover rate [39] and etc. MHAs usually possess a population and the organization of individuals is formed via a population structure [40], which is formally panmictic, cellular, distributed, hierarchical, scale-free, etc [41], [42]. MHAs based on these specific population structures have shown great improvement in terms of optimization performance [43]–[46]. The balance of exploitation and exploration is always considered as a key scientific issue for MHAs [47], and various attempts have already been made to achieve such balance [48]. In addition to convergence analysis for MHAs, theoretical analysis of search dynamics has also received great interests recently [49], [50]. Additionally, hybridization or ensemble strategies to combine several different MHAs have been considered as a promising method to improve their performance [51], among which an MHA incorporated with a local search operator is termed as the memetic computing [52]. It is especially flexible to use problem-inherent information or knowledge to design local search operators for discrete optimization problems, e.g., traveling salesman problem [53], job-shop scheduling problem [54], location routing problem [55], etc. For continuous optimization, local search operators, including random walk [56], Levy flight [57], Cauchy and Gaussian mutations [58], and chaotic local search (CLS) [59], usually perform an excellent local exploitation in the search space.

The chaotic local search (CLS), which fully utilizes the characteristics of chaotic maps, has been regarded as one of the most promising strategies to improve the performance of MHAs [60]–[62]. By incorporating the ergodicity and non-repetitious nature of chaos [63], CLS not only urges an MHA to exploit the local neighborhood of a solution, but also enables it to get rid of the local minima once it is trapped [64]. Initially, chaotic maps are used to generated chaotic sequences to replace the random numbers in MHAs. It has been demonstrated in [65] that chaotic sequences generally performs better than random ones in evolutionary algorithms. Later, many MHAs have used chaotic sequences, substituting random numbers, to generate parameters' values to maintain the randomness of the search. In [66], the Logistic chaotic map is used to generate values for inertia weights in PSO, while the acceleration parameters are also generated by chaotic sequences in PSO [67]. Twelve different chaotic maps are tested by combining with an accelerated PSO, and the results suggested that the sinusoidal map performs the best [68]. A Logistic chaotic sequence is used to generate population for grey wolf optimization (GWO) [69]. Compared with substituting random number by chaotic sequences, CLS is more promising as it can search for better solutions directly. In [70], the piecewise liner chaotic map (PWLCM) is implemented to perform a CLS and then incorporated into PSO. CLS has already been widely employed in gravitational search algorithm [71]–[73], artificial bee colony optimization [74], brain storm optimization [75], differential evolution [76], salp swarm algorithm [77], and many others [62], [78]–[82]. It is worth pointing out that these previous algorithms only use a single chaotic map to perform CLS, while most recently several searches have noticed that multiple chaotic maps might perform better as they can simultaneously use different search dynamics. In [83], several chaotic maps are selectively used to perform CLS in a differential evolution algorithm based on the accumulated success information. Similarly, multiple chaotic maps are used parallelly to implement CLS in gravitational search algorithm [84], cuckoo search algorithm [85], and harmony search algorithm [86]. All these results suggest that multiple chaotic maps incorporated MHAs might perform better in comparison with a single one. Nevertheless, it is unclear that which type of chaotic map should be used for a specific MHA and the number of chaotic maps used in CLS is still problem-dependent.

Based on the above research motivations, in this work, we for the first time perform a comparative study on chaos embedded grey wolf optimization algorithms. The grey wolf optimization (GWO) algorithm [87], which mimics the hunting mechanisms of grey wolves and their hierarchical leadership, has received much interest and achieved great success in many applications, such as control [88], Internet of things [89], engineering design [90], etc. However, GWO still suffers from premature convergence and low capacity of jumping out of local optimum once it is trapped [91]. Although there are also some criticisms regarding the novelty of GWO [92] that it is a reiteration of ideas arisen from PSO and DE, it is still meaningful and challenging to improve the performance of GWO, not only for the diversity of research, but also for the practical applications. To further improve the search performance of GWO, this paper proposes a number of local chaotic search-based GWOs, by means of single chaotic map and multiple chaotic maps incorporation strategies, respectively. In single chaotic map incorporated GWOs (CGWOs), only a single chaotic map is implemented to perform CLS, while in the multiple chaotic maps incorporated GWO (MCGWO), all available chaotic maps are simultaneously implemented. In each iteration of the implementation of MCGWO, a chaotic map from a set of multiple chaotic maps is selected based on a probability, which is generated by an accumulated success based mechanism. It is expected that the most effective chaotic map has the highest probability to survive into the next iteration to perform the CLS. The objective of this study is to find out whether CLS can take effect on GWO and which is the most effective chaotic map for GWO. To realize this, we conduct extensive experiments based on IEEE Congress on Evolutionary Computation (CEC) benchmark optimization functions. Comprehensive comparative results are obtained, from which our valuable findings are summarized as follows:

1) Experimental results show that both single and multiple chaotic maps incorporated GWOs, i.e., CGWOs and MCGWO, can generally perform better than the original GWO. Furthermore, search dynamic analysis suggests that chaotic maps can enable GWO to have a better exploration ability, especially in the earlier search phase, which is potentially benefit for improving the algorithm's ability of jumping out the local optimum. All these indicate that CLS is an effective method for GWO to alleviate its premature convergence and further improve its search performance.

2) Comparative study also finds out that the PWLCM is the most suitable choice for CGWOs when optimizing problems with low dimensions, while Gaussian map seems to perform the best when handling high dimensional problems.

3) Although some previous researches [83], [84] suggested that the simultaneous utilization of multiple chaotic maps can perform better than a single one for gravitational search and differential evolution algorithms, it is not true for GWO, for which the single chaotic map performs better. This result indicates that the incorporation scheme of CLS for meta-heuristics is algorithm-oriented. It is worth studying the number of chaotic maps and their incorporation scheme for each meta-heuristic algorithm.

The contribution of this work to the literature can be summarized as follows: First of all, to our best knowledge, this is the first work that comprehensively analyzes the effects of single and multiple maps embedded CLS on GWO. From the analysis results, the choice of both the embedding type of CLS and the number of chaotic maps are discovered. Second, as a strategy to further improve algorithms' search performance, CLS is again verified to perform very well on algorithms by means of a memetic manner. Last but not least, we also provide powerful and effective optimization methods, i.e., CGWOs and MCGWO, for practical problems.

The remainder of this work is organized as follows: In Section 2, we introduced twelve different types of chaotic maps. In Section 3, we describe the original GWO. In Section 4, we illustrate chaotic local search and chaos incorporated grey wolf optimization algorithms. In Section 5, we present the results of comparative experiments. Finally, we present a summary and future research directions.

II. CHAOTIC MAPS

Different chaotic maps have been widely used in meta-heuristic algorithms. In this study, we took twelve the most widely used chaotic maps for analysis. These chaotic maps' determination equations are summarized as follows:

(1) Logistic map:

$$z_{t+1} = \mu z_k(1 - z_t) \tag{1}$$

where z_t is the tth chaotic number, $z_t \in (0, 1)$, the initial $z_0 \in (0, 1)$ and $z_0 \notin \{0, 0.25, 0.5, 0.75, 1.0\}$. We set μ=4 and z_0=0.152.

(2) PWLCM:

$$z_{t+1} = \begin{cases} z_t/p, & z_t \in (0, p) \\ (1 - z_t)(1 - p), & z_t \in [p, 1) \end{cases} \tag{2}$$

p is set to 0.7, z_0=0.002.

(3) Singer map:

$$z_{t+1} = \mu(7.86 z_t - 23.31 z_t^2 + 28.75 z_t^3 - 13.302875 z_t^4) \tag{3}$$

When μ is set between 0.9 and 1.08, singer map exhibits chaotic behaviors, and we set μ=1.073 and z_0=0.152.

(4) Sine map:

$$z_{t+1} = \frac{a}{4} sin(\pi z_t) \tag{4}$$

where $a \in (0, 4]$, and $z \in (0, 1)$. We set $a = 4$ and $z_0 = 0.152$.

(5) Gaussian map:

$$z_{t+1} = \begin{cases} 0, & z_t = 0 \\ (\mu/z_t) mod(1), & z_t \neq 0 \end{cases} \tag{5}$$

we set $\mu = 1$ and $z_0 = 0.152$.

(6) Tent map:

$$z_{t+1} = \begin{cases} z_t/\beta, & 0 < z_t \leq \beta \\ (1 - z_t)/(1 - \beta), & \beta < z_t \leq 1 \end{cases} \tag{6}$$

where $\beta = 0.4$ and $z_0 = 0.152$.

(7) Bernoulli map:

$$z_{t+1} = \begin{cases} z_t/(1 - \lambda), & 0 < z_t \leq 1 - \lambda \\ (z_t - 1 + \lambda)/\lambda, & 1 - \lambda < z_t < 1 \end{cases} \tag{7}$$

where $\lambda = 0.4$ and $z_0 = 0.152$.

(8) Chebyshev map:

$$z_{t+1} = \cos(\phi \cos^{-1} z_t) \tag{8}$$

where $\phi = 5$ and $z_0 = 0.152$.

(9) Circle map:

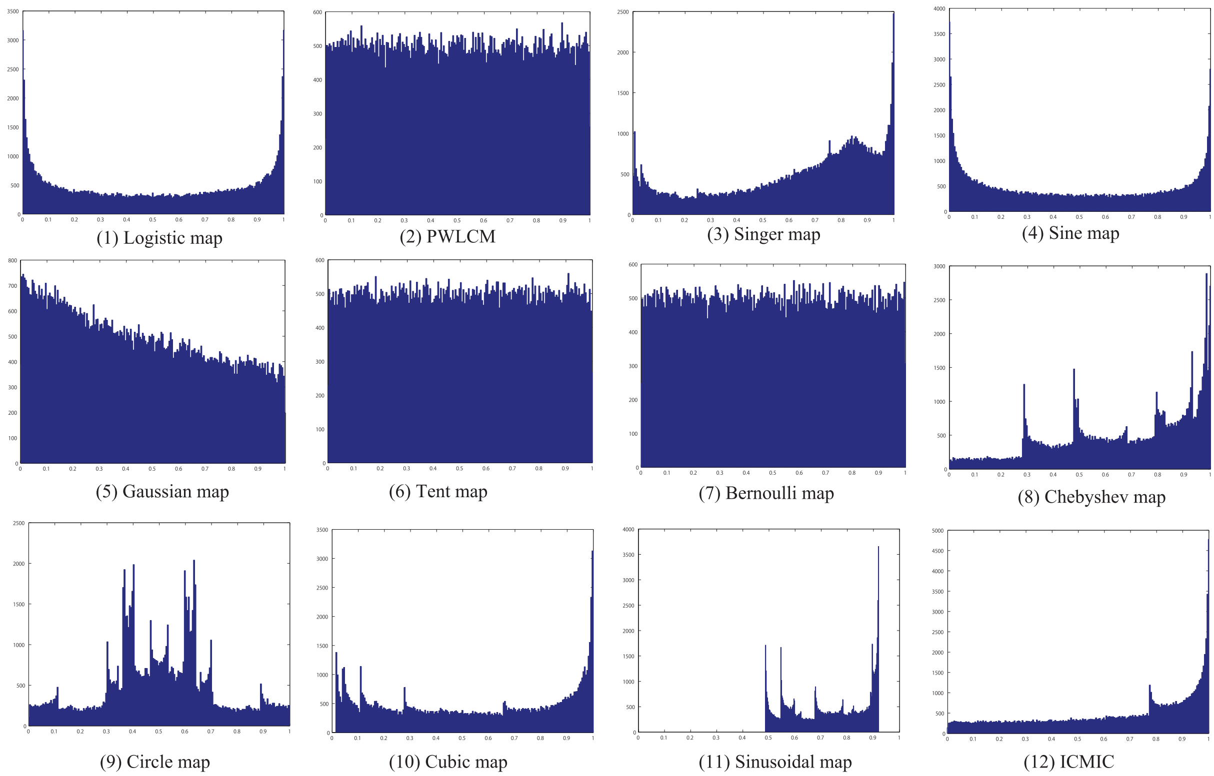
$$z_{t+1} = z_t + a - \frac{b}{2\pi} \sin(2\pi z_t) mod(1) \tag{9}$$

When $a = 0.5$ and $b = 2.2$, circle map shows chaotic features. We set $z_0 = 0.152$ in the experiment.

(10) Cubic map:

$$z_{t+1} = \rho z_t(1 - z_t^2) \tag{10}$$

where $\rho = 2.59$ and $z_0 = 0.242$.



(1) Logistic map (2) PWLCM (3) Singer map (4) Sine map
(5) Gaussian map (6) Tent map (7) Bernoulli map (8) Chebyshev map
(9) Circle map (10) Cubic map (11) Sinusoidal map (12) ICMIC

FIGURE 1. Histogram graphs of twelve chaotic maps considered in this study.

(11) Sinusoidal map:

$$z_{t+1} = a z_t^2 \sin(\pi z_t) \tag{11}$$

where $a = 2.3$ and $z_0 = 0.74$.

(12) Iterative chaotic map with infinite collapses (ICMIC):

$$z_{t+1} = \sin(a/z_t) \tag{12}$$

where $a \in (0, \infty)$, and we set $a = 70$ in experiment. ICMIC generates sequence in $(-1, 0) \cup (0, 1)$. Therefore, if the value is negative, its absolute value is used.

The histogram graphs of all considered chaotic maps with twenty thousand generated points are depicted in Fig. 1. From it, we can find that different chaotic maps possess different points distribution, which might have significant influence on the search length in CLS. Thus, it is valuable to find out which chaotic map can perform the best for a chaotic meta-heuristic algorithm.

III. GREY WOLF OPTIMIZATION (GWO)

GWO is one of the most widely recognized meta-heuristic algorithms [87]. It is inspired by the prey-predation activities of gray wolves and developed as an effective optimization method. The GWO's optimization process includes gray wolf social hierarchy and hunting behavior. The specific steps of GWO can be expressed via Eqs. (13)-(16).

$$\vec{x}\,(t+1) = \frac{\vec{x_1} + \vec{x_2} + \vec{x_3}}{3} \tag{13}$$

where t is the number of current iteration. $\vec{x_1}$, $\vec{x_2}$ and $\vec{x_3}$ can be calculated as:

$$\begin{aligned} \vec{x_1} &= \vec{x_\alpha} - \vec{A_1} \cdot \left(\vec{R_\alpha}\right) \\ \vec{x_2} &= \vec{x_\beta} - \vec{A_2} \cdot \left(\vec{R_\beta}\right) \\ \vec{x_3} &= \vec{x_\delta} - \vec{A_3} \cdot \left(\vec{R_\delta}\right) \end{aligned} \tag{14}$$

where $\vec{x_\alpha}$, $\vec{x_\beta}$ and $\vec{x_\delta}$ are the positions of individuals α, β and δ, respectively. α is the leader of grey wolves with the best fitness value. β presents the second level with the second best fitness value, and δ denotes the third level with the third one. $\vec{R}$ represents the distance between the current candidate gray wolf and the best three wolves, and it can be formulated as:

$$\begin{aligned} \vec{R_\alpha} &= \left|\vec{B_1} \cdot \vec{x_\alpha} - \vec{x}\right| \\ \vec{R_\beta} &= \left|\vec{B_2} \cdot \vec{x_\beta} - \vec{x}\right| \\ \vec{R_\delta} &= \left|\vec{B_3} \cdot \vec{x_\delta} - \vec{x}\right| \end{aligned} \tag{15}$$

where $\vec{A}$ and $\vec{B}$ are synergy coefficient vectors, and they can be calculated as:

$$\begin{aligned} \vec{A} &= 2a \cdot \vec{r_1} - a \\ \vec{B} &= 2 \cdot \vec{r_2} \end{aligned} \tag{16}$$

where r_1, r_2 are random vectors generated between 0 and 1. A decrease for the value of a will cause the value of $\vec{A}$ to fluctuate. In other words, $\vec{A}$ is a random vector in the interval $[-a, a]$, where a decreases linearly along with the iterations.

As summarized in [87], when $|A| > 1$, gray wolves are scattered in various areas as far as possible and search for prey. When $|A| < 1$, the gray wolf will focus on hunting prey in a certain area. Another parameter in the GWO algorithm is $\vec{B}$, which is a vector of random values in the interval [0, 2].

The Pseudo-code of GWO is shown in Algorithm 1. Through the previous analysis of GWO [87], [93], [94], we can find that GWO has the characteristics of strong exploitation ability, but it usually suffers from premature convergence because the top three individuals in the population greatly take effect on other individuals, thus making the search move toward these three individuals too much.

Algorithm 1 Procedures of GWO

Initialization
repeat
  for $i = 1$ to n **do**
    Update the fitness of X_α, X_β and X_δ
    if $f(x_i) < f(X_\alpha)$ **then**
      $f(X_\alpha) = f(x_i)$, $X_\alpha = x_i$
    end if
    if $f(x_i) > f(X_\alpha)$ and $f(x_i) < f(X_\beta)$ **then**
      $f(X_\beta) = f(x_i)$, $X_\beta = x_i$
    end if
    if $f(x_i) > f(X_\alpha)$ and $f(x_i) > f(X_\beta)$ and $f(x_i) < f(X_\delta)$ **then**
      $f(X_\delta) = f(x_i)$, $X_\delta = x_i$
    end if
  end for
  for $i = 1$ to n **do**
    Calculate $\vec{A}$ and $\vec{C}$ by Eq. (16)
    Calculate $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ by Eq. (14)
    Calculate $\vec{X}$ by Eq. (13)
  end for
  Evaluate $f(x_i')$
until Iteration number

IV. CHAOTIC GREY WOLF OPTIMIZATION ALGORITHMS

In this section, the chaotic grey wolf optimization algorithms will be elaborated. There are generally two methods to incorporate chaotic maps into a heuristic algorithm, i.e., using chaotic sequences to substitute the random numbers in the algorithm, or using CLS to perform a local search. Recently, it has been widely accepted that CLS generally perform better than the sequence substitution method, because the latter is only implemented as a parameter control manner [35] while the former can directly search in the landscape [73]. Thus, in this study we only discuss the CLS scheme to be incorporated into GWO.

A. CHAOTIC LOCAL SEARCH

The CGWO adds a CLS operator to the GWO to maintain a high population diversity as well as a strong ability to get rid of the local optimum. In this process, random numbers generated by J chaotic maps are selected as parameters to adjust the radius of the local search. This operator acts on the best individual α generated by the GWO. The unified implementation manner of CLS can be expressed by:

$$x_{g'} = x_g + v^j \cdot (x_{r2} - x_{r1}), \tag{17}$$

where x_g represents the best individual in current population. v^j is the parameter generated by the chaotic map j ($j = 1, 2, \ldots, J$), and its range is (0, 1). $x_{g'}$ is a temporary individual generated by the local search operator. If the fitness value of $x_{g'}$ is better than x_g, $x_{g'}$ replaces x_g, otherwise keep x_g to be survived into the next iteration. Besides, x_{r1} and x_{r2} are two individuals randomly selected from the population of individuals U. In U, we assume the individual with the greatest fitness as U_{max}. If the fitness of $x_{g'}$ is better than that of U_{max}, replace U_{max} with $x_{g'}$. The conceptual sketch of CLS is illustrated in Fig. 2.

Algorithm 2 Chaotic Local Search

Randomly pick up two individuals x_{r1} and x_{r2}
Pick out the best individuals x_g in the population
Using chaotic map j to generate a random value v^j
$x_g' \leftarrow x_g + v^j \cdot (x_{r2} - x_{r1})$

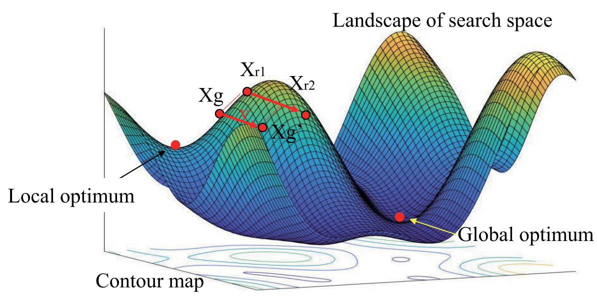


FIGURE 2. Conceptual sketch of CLS: a two-dimensional multi-model landscape with multiple local optima is illustrated, where CLS is desired to enable the algorithm to jump out the local optimum and find the global optimum eventually.

For CLS, some remarks are given as:

1) The local search acts on the current optimal solution x_g to make a trade-off between search performance and computational complexity. In this study, x_g is the $\vec{x_\alpha}$ in each iteration of GWO;
2) Once the generated $x_{g'}$ exceeds the search boundary, it will be reset within the search range by a random feasible value.

Algorithm 2 shows the pseudo-code of CLS.

B. SINGLE CHAOS EMBEDDED CGWO

The CLS that utilizes only single chaotic map is shown as:

$$x_{g'}(t) = x_g(t) + v(t) \cdot r \cdot (x_{r2}(t) - x_{r1}(t)) \qquad t = 1, 2, \ldots, T \tag{18}$$

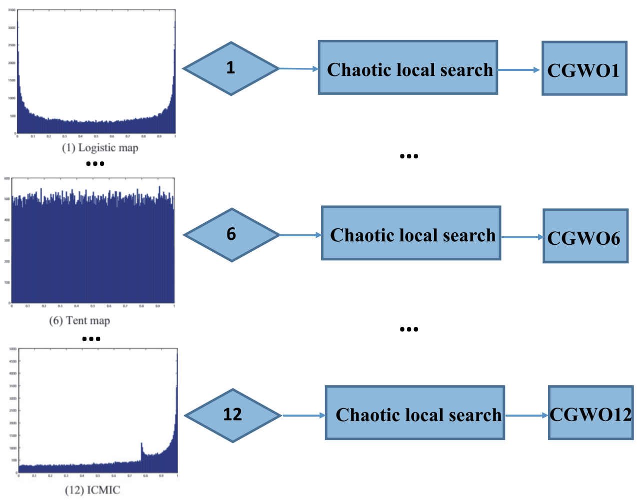


FIGURE 3. Incorporation scheme of CLS in CGWO.

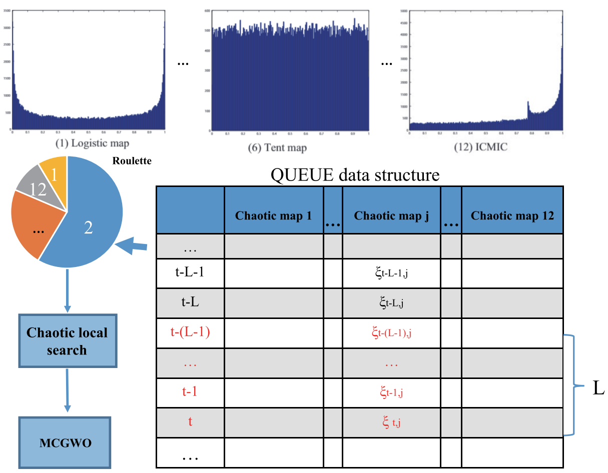


FIGURE 4. Incorporation scheme of CLS in MCGWO.

where t is the iteration number. r is a scaling parameter of CLS. Due to the premature convergence of GWO, we assign r equal to 5 to enlarge the scope of CLS in CGWO and thereafter maintain the diversity of the population.

In this study, the optimization problems are minimization ones. After performing CLS in Eq. (18), an update process is implemented as:

$$x_g(t) = \begin{cases} x_{g'}(t) & \text{If } \ f(x_{g'}(t)) \leq f(x_g(t)), \\ x_g(t) & \text{Otherwise} \end{cases} \tag{19}$$

where f denotes the fitness function of the optimization problem. If the fitness is improved, the offspring individual $x_{g'}$ will replace the current best individual, while other individuals enter into the next iteration. The different types of single chaos embedded CGWO using the chaotic map in Eqs. (1) $\sim$ (12) are termed as CGWO1 $\sim$ CGWO12, respectively. The schematic diagram of CGWO is shown in Fig. 3.

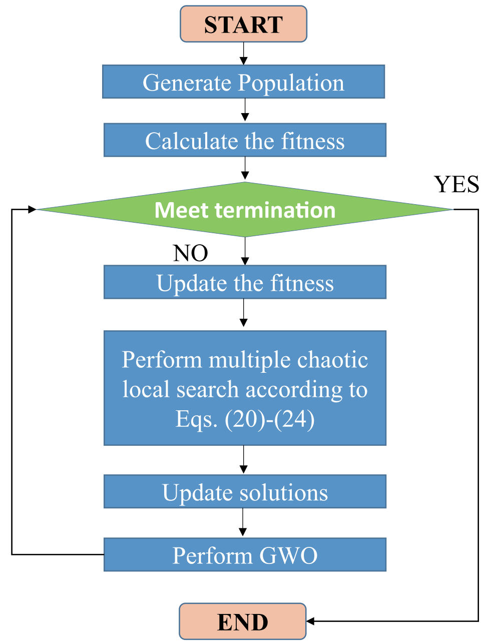


FIGURE 5. Flowchart of MCGWO.

Algorithm 3 Procedures of CGWOs and MCGWO

Initialization
repeat
  for $i = 1$ to n **do**
    Update the fitness of X_α, X_β and X_δ
    if $f(x_i) < f(X_\alpha)$ **then**
      $f(X_\alpha) = f(x_i)$, $X_\alpha = x_i$
    end if
    if $f(x_i) > f(X_\alpha)$ and $f(x_i) < f(X_\beta)$ **then**
      $f(X_\beta) = f(x_i)$, $X_\beta = x_i$
    end if
    if $f(x_i) > f(X_\alpha)$ and $f(x_i) > f(X_\beta)$ and $f(x_i) < f(X_\delta)$ **then**
      $f(X_\delta) = f(x_i)$, $X_\delta = x_i$
    end if
  end for
  CGWOs: implement Eqs. (18)(19)
  MCGWO: implement Eqs. (20)$\sim$ (24) and Eq. (19)
  if $f(X'_\alpha) < f(X_\alpha)$ **then**
    $X_\alpha \leftarrow X'_\alpha$;
  end if
  for $i = 1$ to n **do**
    Calculate $\vec{A}$ and $\vec{C}$ by Eq. (16)
    Calculate $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ by Eq. (14)
    Calculate $\vec{X}$ by Eq. (13)
  end for
  Evaluate $f(x'_i)$
until Iteration number

C. MULTIPLE CHAOS EMBEDDED CGWO

The core implementation framework of multiple chaos embedded GWO (MCGWO) is illustrated in Fig. 4, and its

TABLE 1. IEEE CEC2017 benchmark functions' definition.

Name	Functions	Property	Minimization value
F1	Shifted and Rotated Bent Cigar Function	Unimodal Functions	100
F3	Shifted and Rotated Zakharov Function		300
F4	Shifted and Rotated Rosenbrock's Function	Simple Multimodal Functions	400
F5	Shifted and Rotated Rastrigin's Function		500
F6	Shifted and Rotated Expanded Scaffer's F6 Function		600
F7	Shifted and Rotated Lunacek Bi_Rastrigin Function		700
F8	Shifted and Rotated Non-Continuous Rastrigin's Function		800
F9	Shifted and Rotated Levy Function		900
F10	Shifted and Rotated Schwefel's Function		1000
F11	Hybrid Function 1	Hybrid Functions	1100
F12	Hybrid Function 2		1200
F13	Hybrid Function 3		1300
F14	Hybrid Function 4		1400
F15	Hybrid Function 5		1500
F16	Hybrid Function 6		1600
F17	Hybrid Function 7		1700
F18	Hybrid Function 8		1800
F19	Hybrid Function 9		1900
F20	Hybrid Function 10		2000
F21	Composition Function 1	Composition Functions	2100
F22	Composition Function 2		2200
F23	Composition Function 3		2300
F24	Composition Function 4		2400
F25	Composition Function 5		2500
F26	Composition Function 6		2600
F27	Composition Function 7		2700
F28	Composition Function 8		2800
F29	Composition Function 9		2900
F30	Composition Function 10		3000

TABLE 2. Parameter settings of the heuristic algorithms.

Algorithms	Parameters
CGWO	$N = 100$, α linearly decreases from 2 to 0, $r = 5$
MCGWO	$N = 100$, α linearly decreases from 2 to 0, $L = 24$, $r = 5$
GWO	$N = 100$, α linearly decreases from 2 to 0

TABLE 3. Experimental results obtained by CGWOs, MCGWO and GWO on CEC2017.

$D = 30$		$D = 50$		$D = 100$	
Algorithm	W/T/L	Algorithm	W/T/L	Algorithm	W/T/L
CGWO1	14/4/11	CGWO1	11/8/10	CGWO1	13/4/12
CGWO2	14/13/2	CGWO2	14/6/9	CGWO2	13/7/9
CGWO3	12/8/9	CGWO3	12/6/11	CGWO3	13/5/11
CGWO4	13/8/8	CGWO4	11/7/11	CGWO4	13/6/10
CGWO5	15/7/7	CGWO5	13/9/7	CGWO5	15/12/2
CGWO6	12/14/3	CGWO6	12/8/9	CGWO6	13/6/10
CGWO7	13/14/2	CGWO7	12/9/8	CGWO7	13/6/10
CGWO8	8/19/2	CGWO8	6/14/9	CGWO8	13/5/11
CGWO9	11/15/3	CGWO9	11/10/8	CGWO9	13/6/10
CGWO10	13/9/7	CGWO10	12/9/8	CGWO10	13/5/11
CGWO11	1/25/3	CGWO11	0/13/16	CGWO11	0/10/19
CGWO12	12/16/1	CGWO12	11/9/9	CGWO12	13/7/9
MCGWO	13/11/5	MCGWO	13/10/6	MCGWO	13/7/9
GWO	-/-/-	GWO	-/-/-	GWO	-/-/-

flowchart is given in Fig. 5. We use J ($J = 12$) chaotic maps in Eqs. (1) $\sim$ (12) to generate chaotic sequences $v^j(t)$ ($j = 1, 2, \ldots, J$, $t = 1, 2, \ldots$). After that, the following method is used to perform a multi-chaotic local search:

$$x^j_{g'}(t) = x_g(t) + v^j(t) \cdot r \cdot (x_{r1}(t) - x_{r2}(t)) \qquad j = 1, 2, \ldots, J; t = 1, 2, \ldots, T \tag{20}$$

In order to select the appropriate chaotic map j among all J chaotic maps, a roulette selection method based on the accumulated success information is used. As illustrated in Fig. 4, let $\xi_{t,j}$ (its initial value is 0) denotes the difference of a successful improvement for $x_g(t)$ at iteration t using the j-th chaotic map. Although several chaotic maps are used in the whole search iterations, CLS uses only one in each iteration, and its probability of being selected $p_{t,j}$ is calculated as follows:

$$p_{t,j} = \frac{S_{t,j}}{\sum_{m=1}^{J} S_{t,m}} \tag{21}$$

$$S_{t,j} = \frac{\xi_{t,j}}{\sum_{n=max(k-L+1,1)}^{L} \xi_{n,j}} + 1/J \tag{22}$$

$$\xi_{t,j} = \begin{cases} \xi_{t-1,j} + \Delta, & \text{if } f(x^j_{g'}(t)) < f(x_g(t)) \ \& \ t \le L \\ \xi_{t-1,j}, & \text{if } f(x^j_{g'}(t)) \ge f(x_g(t)) \ \& \ t \le L \\ \xi_{t-1,j} + \Delta - (\xi_{t-L,j} - \xi_{t-L-1,j}), & \text{if } f(x^j_{g'}(t)) < f(x_g(t)) \ \& \ t > L \\ \xi_{t-1,j} - (\xi_{t-L,j} - \xi_{t-L-1,j}), & \text{otherwise} \end{cases} \tag{23}$$

$$\Delta = f(x_g(t)) - f(x^j_{g'}(t)) \tag{24}$$

If $x^j_{g'}(t)$ outperforms $x_g(t)$, it means that the chaotic map in the current iteration succeeded in the local search, and

TABLE 4. Friedman test results on IEEE CEC2017 with $D = 30, 50, 100$.

		Friedman	Ranking	p-values	Friedman	Ranking	p-values	Friedman	Ranking	p-values
			$D = 30$			$D = 50$			$D = 100$	
CGWO1	Logistic map	8.6207	10	0.001523	8.9655	11	0.000146	8.4483	10	0.000272
CGWO2	PWLCM	**5.1379**	**1**	–	6.1379	5	0.220899	6.7586	6	0.035465
CGWO3	Singer map	8.4138	9	0.002865	7.8621	9	0.005213	8.931	11	0.000045
CGWO4	Sine map	9	12	0.000439	11.1379	13	0	10.5862	13	0
CGWO5	Gaussian map	5.3793	2	0.826091	**4.7931**	**1**	–	**4.4483**	**1**	–
CGWO6	Tent map	5.7931	4	0.550924	6.2414	6	0.187401	7	8	0.020194
CGWO7	Bernoulli map	5.6897	3	0.615519	5.1034	3	0.777565	6.7586	6	0.035465
CGWO8	Chebyshev map	8.8966	11	0.000623	9.5172	12	0.000017	9.069	12	0.000026
CGWO9	Circle map	7.7586	8	0.017056	8.5172	10	0.000699	6.1724	3	0.116552
CGWO10	Cubic map	6.8621	6	0.116552	4.8276	2	0.97496	6.4138	4	0.073594
CGWO11	Sinusoidal map	9.7241	14	0.00003	11.4138	14	0	11.6897	14	0
CGWO12	ICMIC	6.7586	5	0.140146	7.4828	7	0.014354	6.6897	5	0.041327
MCGWO	All 12 maps	7.4138	7	0.038301	5.4483	4	0.550924	4.8621	2	0.706427
GWO		9.5517	13	0.000059	7.5517	8	0.012037	7.1724	9	0.01315

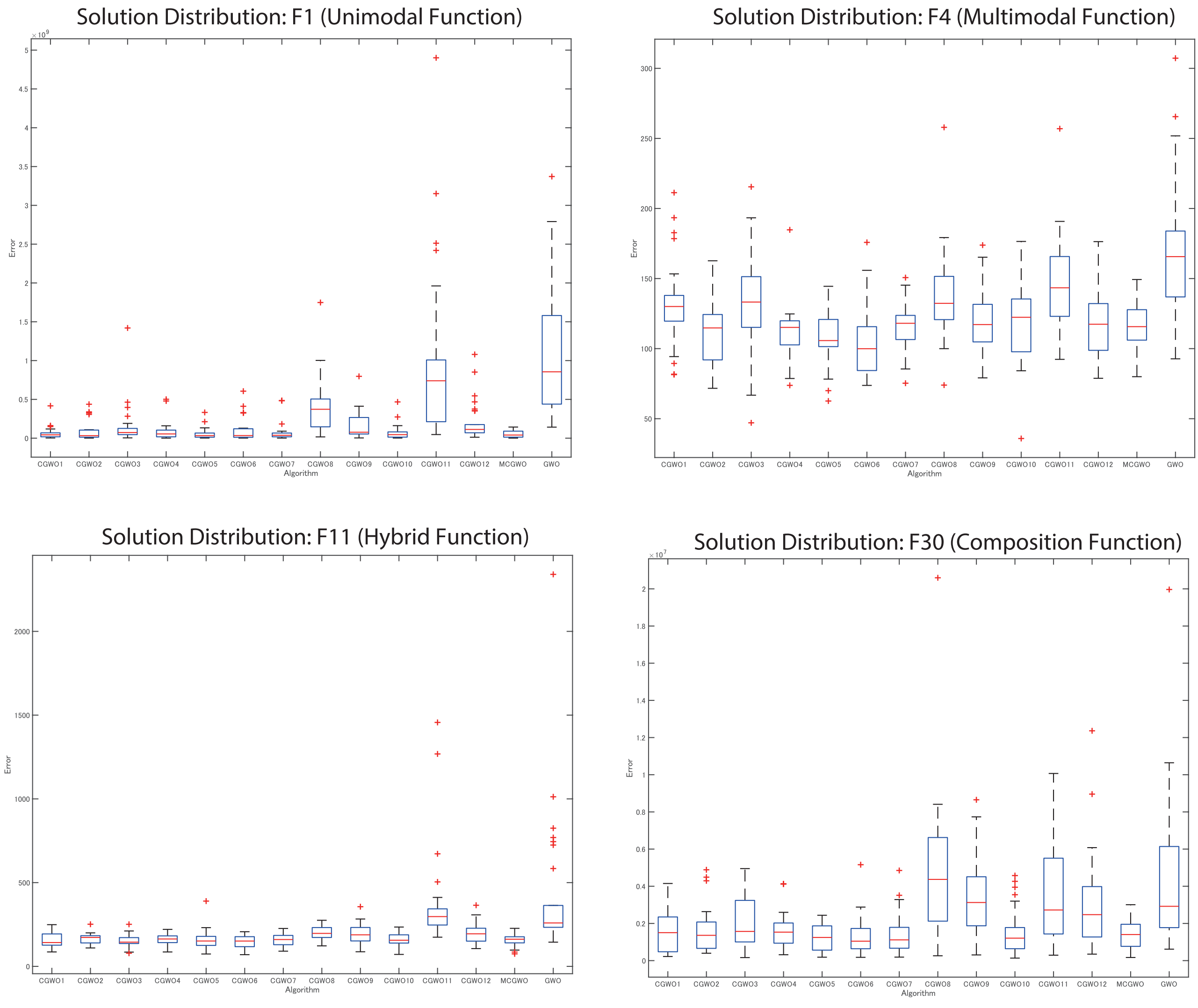


FIGURE 6. Solution distribution of GWO, CGWOs, and MCGWO algorithms on IEEE CEC2017 functions ($D = 30$).

we add Δ to $\xi_{t-1,j}$. f is the fitness function, and $L = 24$ is the maximum number of iterations for storing cumulative success information. The above operations make the chaotic map that is more advantageous in the current search environment more likely to be selected. Eq. (21) shows the normalized selection probability $p_{t,j}$ of the jth chaotic map

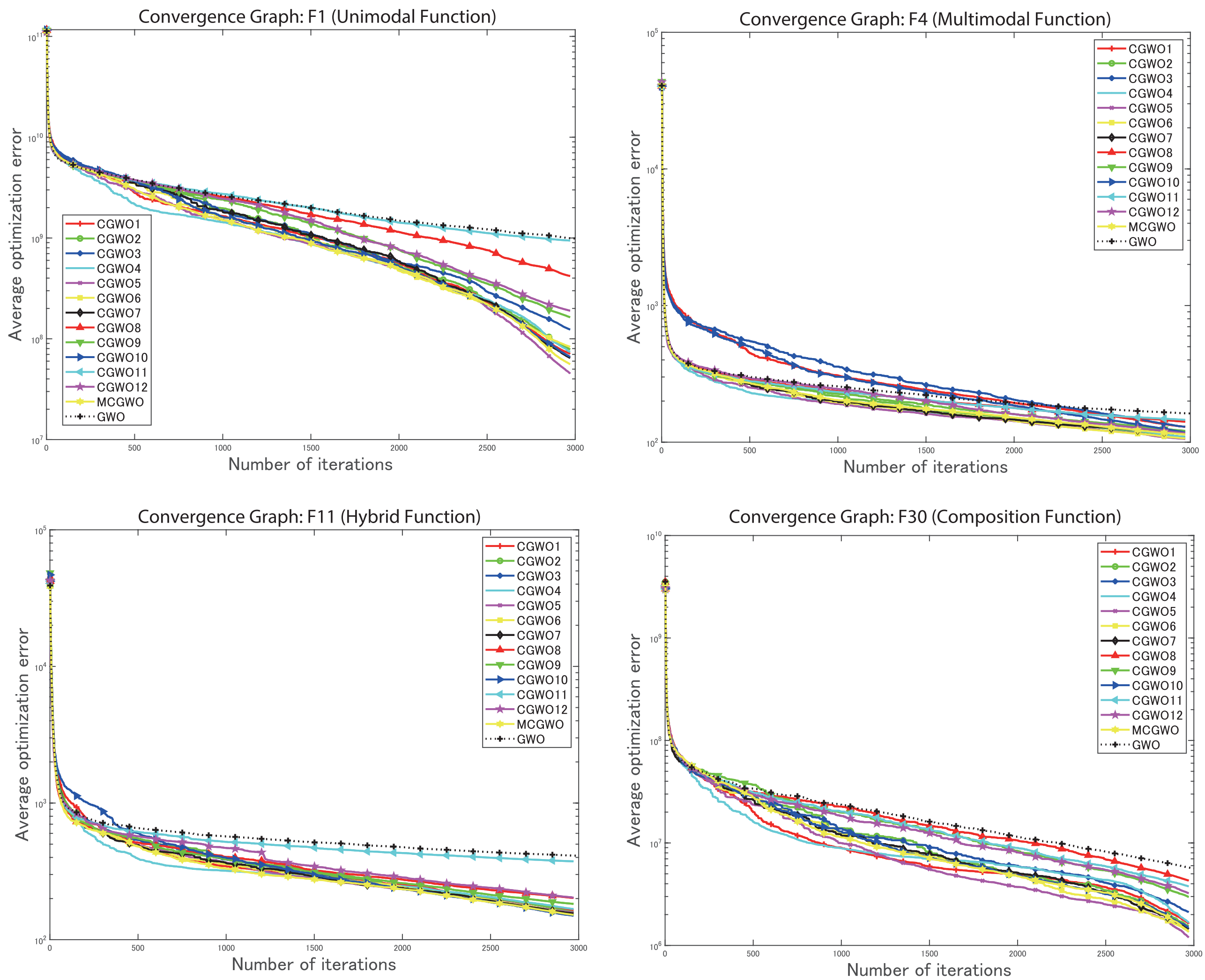


FIGURE 7. Convergence characteristics of GWO, CGWOs, and MCGWO algorithms on IEEE CEC2017 functions ($D = 30$).

at the tth iteration. When t is 0, all $p_{t,j}$ values are equal to $1/J$, which means that all chaotic maps have the same probability to be selected to perform CLS. Along with the evolution, the selection probabilities change with the feedback from the population, which is manipulated by Eqs. (22) and (23). Eq. (23) is used to calculate the accumulative success value ξ for each chaotic map. The basic idea behind this calculation is that the chaotic maps which are more potential to give better solutions are more easily to be selected in CLS.

Regarding the selection method, more detailed explanations are given as:

(1) When $t <= L$, if the selected chaotic map j has performed a success evolution, we add Δ to its previous success value $\xi_{t-1,j}$ (i.e., $1st$ conditional formula in Eq. (23));
(2) If no improvement of current selected chaotic map j has been found, the success value ξ keeps the same (i.e., $2nd$ conditional formula in Eq. (23));
(3) The longest storage of such success memory is L, once $t > L$, we use QUEUE data structure to calculate the accumulative success value, which means only the information of the past L iterations should be accumulated. Here, we give the illustrative QUEUE data structure with size of L (first-in-first-out) of the accumulative success values ξ, as shown in Fig. 4.

After executing CLS, $x_g(t)$ is updated according to Eq. (19). The Pseudo-code of CGWOs and MCGWO is summarized in Algorithm 3.

V. EXPERIMENTS

In this section, we make a comprehensive performance comparison based on the benchmark functions taken from IEEE Congress on Evolutionary Computation 2017 (CEC2017) (https://github.com/P-N-Suganthan/CEC2017-BoundContrained) to verify the performance of proposed methods. The test suit is summarized in Table 1. The objectives of such comparative study is to find out 1) whether

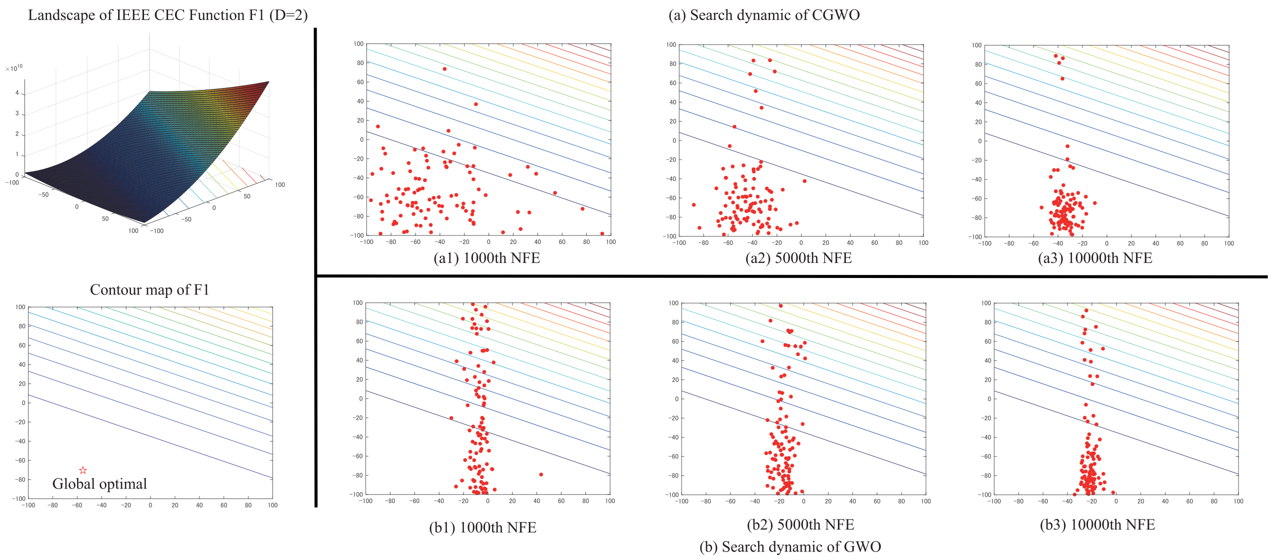


FIGURE 8. **Search dynamics of CGWOs and GWO on IEEE CEC2017 function F1.**

CLS can help GWO perform better and effective enough in comparison with other state-of-the-art meta-heuristic algorithms, 2) which kind of incorporation method should be used for GWO (single chaotic map embedded or multiple chaotic maps embedded), and 3) which chaotic map is the most suitable one to perform CLS for GWO.

A. EXPERIMENTAL SETUP

In all experiments, the common parameters of all algorithms are set in the same manner, which is specifically described as follows: the maximum number of evaluations is set to $10^4 * D$, where D is the dimension of each benchmark function. For each function, every algorithm runs 51 times repeatedly for statistical analysis. The experiments were conducted on a PC with 3.30 GHz Intel(R) Core(TM) i5 CPU and 8GB RAM using MATLAB R2018a.

B. PERFORMANCE EVALUATION CRITERIA

The following criteria were used for evaluating the performance of all compared algorithms.

(1) Non-parametric statistical test: "$W/T/L$" represents the result of the Wilcoxon rank-sum test. This is a non-parametric statistical test used to determine the level of significant difference between algorithms. If the obtained p-value is less than the significance level 0.05, the difference between the two algorithms can be recognized. When the control algorithm is significantly better than its counterpart, it is recorded as "+", otherwise it is recorded as "−". The sign "=" indicates that the compared algorithms are tied on the function.

(2) Convergence curve graph: The convergence curve illustrates the current optimal history at each iteration, thereby comparing the convergence speed of different algorithms. The x-axis represents the number of function evaluations. The y-axis represents the average error value after being logged.

(3) Box-and-whisker diagrams: The line above the blue box means the maximum value, and the line below the box means the minimum value. The upper and lower edges of the box represent the first and third quartiles, respectively. The red line represents the median. The red + indicates extreme values. The longer the distance between the maximum and minimum, the greater the fluctuation of the solution and the more unstable the performance of the algorithm. Furthermore, the lower the position of the graph in the coordinates means the better solution.

C. COMPARISON AMONG CGWOs, MCGWO AND GWO

In this section, we compared the results among CGWOs, MCGWO and GWO on CEC2017 benchmark functions. Table 2 shows the parameters' setting of each algorithm. Among them, N is the population size. The "$W/T/L$" among CGWOs, MCGWO and GWO on CEC2017 test functions with 30, 50 and 100 dimensions are summarized in Table 3, while the detailed experimental results are listed in Appendix.

From Table 3, we can find that most CGWOs and MCGWO can perform significantly better than the original GWO as the number of won cases (W) is more than that of lost cases (L), suggesting that CLS indeed enable GWO to perform better. An exception is CGWO11, which performs worse than GWO. This reveals that Sinusoidal map is not a good choice to perform CLS. The reason might be also observed from Fig. 1 that Sinusoidal map only generate random numbers in the interval of [0.5, 0.9], which substantially limits the search range of CLS.

We also used the Friedman test [95] to rank the compared algorithms. In the Friedman test, the NULL hypothesis

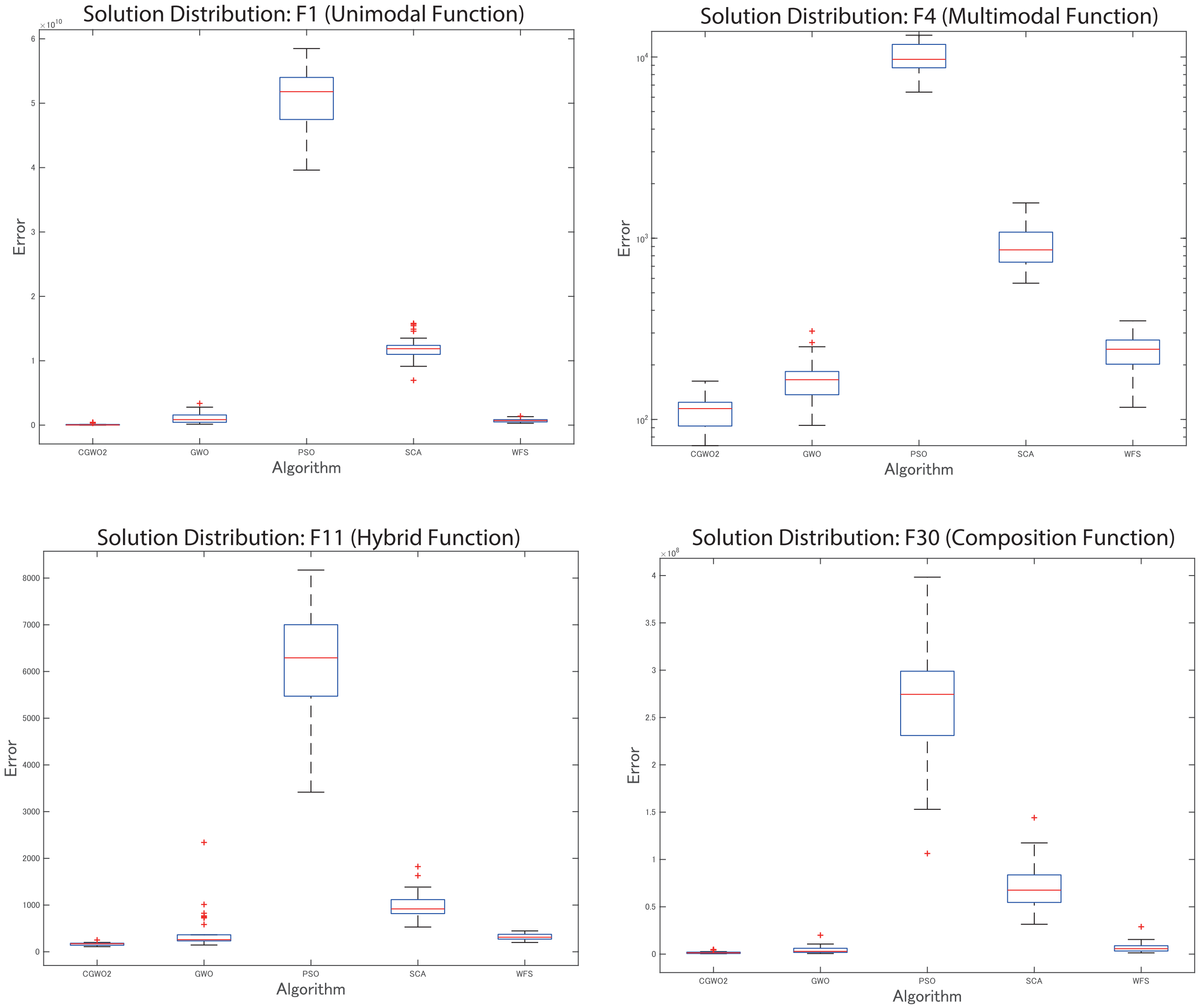


FIGURE 9. **Solution Distribution of CGWO2 and other competitor algorithms on IEEE CEC2017 functions ($D = 30$).**

assumes that mean values among all compared algorithms are the same, and it gives the average rank that each algorithm obtained to estimate its performance. The lower the obtained rank is, the better the algorithm performs. The Friedman test results are summarized in Table 4. From it, it is clear that CGWO2 performs the best for optimization functions with 30 dimensions, and CGWO5 performs the best for 50 and 100 dimensions. This result suggests that 1) the PWLCM is the most suitable candidate chaotic map to perform CLS for GWO when encountering lower dimensional problems, and 2) Gaussian map is the best choice for higher dimensional problems.

It is surprised that, in comparison with CGWOs, MCGWO only ranked 7th, 4th and 2nd among all compared algorithms for 30, 50 and 100 dimensional functions, respectively, although similar method has shown great potential for gravitational search algorithm [84] and differential evolution algorithm [80]. This valuable finding suggests that the multiple chaotic maps incorporated GWO can't significantly improve the performance of GWO, especially for lower dimensional problems. Thus, it motivates us to design more sophisticated incorporation scheme of multiple chaotic maps for CLS as future work, e.g., by using the maximum Lyapunov exponent of each chaotic map.

Additionally, Figs. 6 and 7 show box-and-whisker diagrams and convergence graphs of CGWOs, MCGWO and GWO, respectively. It can be seen from Fig. 6 that most of CGWOs have the lower altitude and shorter distant, which confirms its better performance and stronger robustness in comparison with GWO. We can also see from Fig. 7 that most of the CGWO variants converge much faster than GWO, and in the later stage of convergence, CGWOs and MCGWO still maintain a fairly fast convergence trend.

To conclude, the comparative results among CGWOs, MCGWO, and GWO show that CLS definitely enables GWO to perform better, and PWLCM and Gaussian map

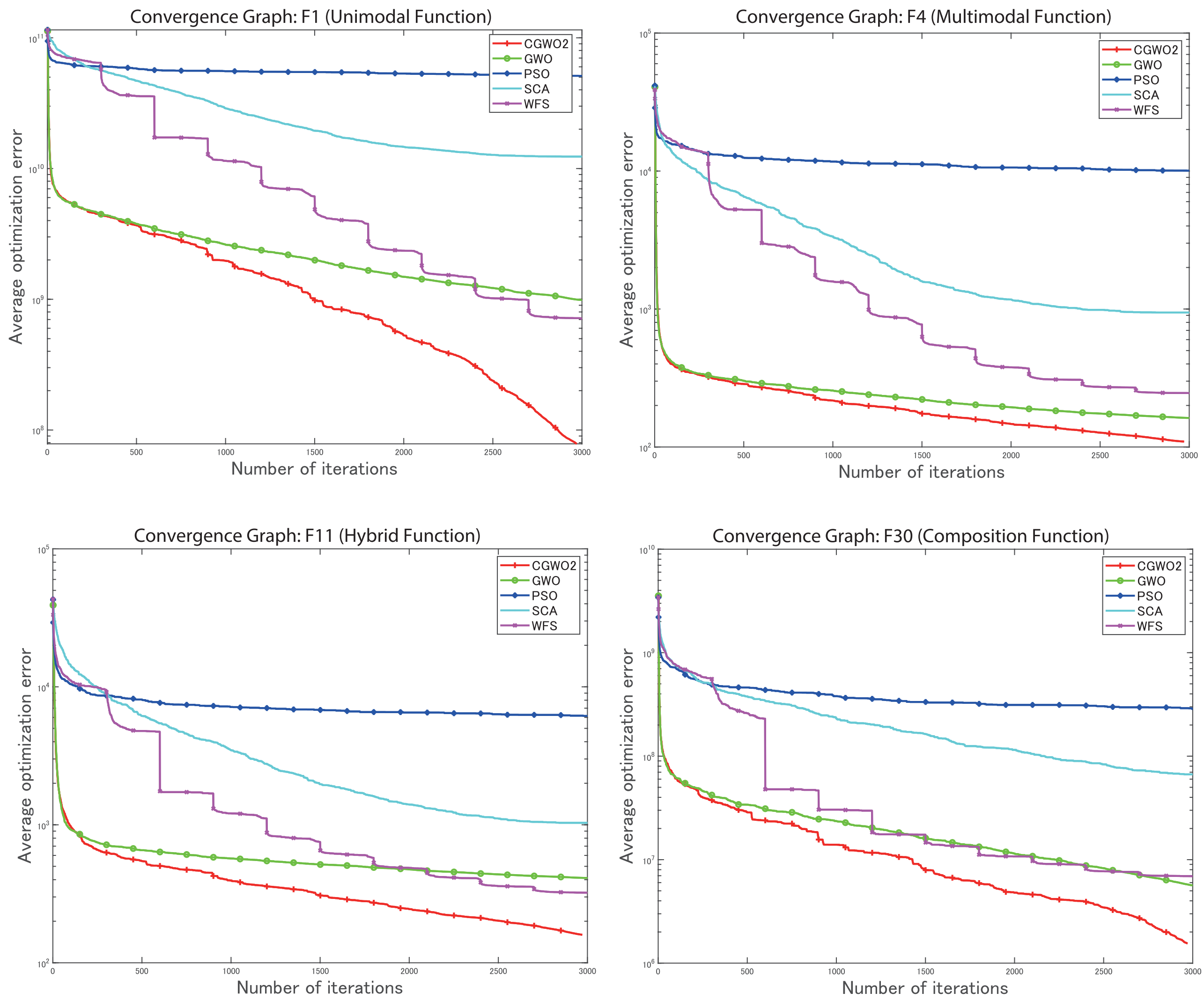


FIGURE 10. Convergence characteristics of CGWO2 and other competitor algorithms on IEEE CEC2017 functions ($D = 30$).

implemented as a single map embedding scheme are two most suitable chaotic maps for GWO.

D. COMPARISON OF CGWO WITH OTHER META-HEURISTIC ALGORITHMS

In this section, we compare CGWOs with other meta-heuristic algorithms, including PSO [96], sine cosine algorithm (SCA) [97], and wingsuit flying search algorithm (WFS) [98]. PSO and SCA are population-based meta-heuristic algorithms, among which PSO simulates bird flocking and animal social behaviors and SCA uses sine and cosine functions to optimize the problem. WFS proposes a new optimization method by simulating the decision-making process during the landing of the glider. The parameters of these algorithms are presented in Table 5.

On CEC2017 benchmark optimization functions, we compared CGWO with above meta-heuristic algorithms under the conditions of 30, 50, and 100 dimensions, respectively. The statistical results are shown in Table 6, while the detailed experimental data are summarized in Appendix. In Table 6, we only summarize the best-performing variants of CGWOs,

TABLE 5. Parameter settings of the heuristic algorithms.

Algorithms	Parameters
CGWO	$N = 100$, α linearly decreases from 2 to 0, $r = 5$
WFS	$N = 3E + 04$, $v > 0$
GWO	$N = 100$, α linearly decreases from 2 to 0
PSO	$N = 100$, $c_1 = 2.0$, $c_2 = 2.0$
SCA	$N = 100$, $\alpha = 2$

TABLE 6. Experimental results on IEEE CEC2017 with $D = 30$, 50, 100.

$D = 30$		$D = 50$		$D = 100$	
Algorithm	W/T/L	Algorithm	W/T/L	Algorithm	W/T/L
CGWO2	-/-/-	CGWO5	-/-/-	CGWO5	-/-/-
GWO	14/13/2	GWO	13/9/7	GWO	15/12/2
PSO	29/0/0	PSO	29/0/0	PSO	29/0/0
SCA	29/0/0	SCA	29/0/0	SCA	29/0/0
WFS	28/1/0	WFS	29/0/0	WFS	20/7/2

TABLE 7. Performance comparison among GWO, CGWOs, and MCGWO in terms of solution accuracy on IEEE CEC2017 with $D = 30$.

	GWO	GWO1		GWO2		GWO3		GWO4	
	mean ± std	mean ± std		mean ± std		mean ± std		mean ± std	
F1	9.89E+08 ± 7.80E+08	7.09E+07 ± 7.71E+07	+	7.84E+07 ± 9.66E+07	+	1.23E+08 ± 2.13E+08	+	7.41E+07 ± 9.82E+07	+
F3	2.99E+04 ± 1.05E+04	3.86E+03 ± 2.95E+03	+	2.45E+03 ± 1.76E+03	+	3.84E+03 ± 2.55E+03	+	5.34E+03 ± 3.22E+03	+
F4	1.62E+02 ± 4.26E+01	1.29E+02 ± 2.94E+01	+	1.10E+02 ± 2.12E+01	+	1.30E+02 ± 3.21E+01	+	1.09E+02 ± 2.00E+01	+
F5	8.67E+01 ± 3.32E+01	1.07E+02 ± 3.60E+01	-	8.58E+01 ± 2.27E+01	=	1.01E+02 ± 2.73E+01	-	9.89E+01 ± 3.76E+01	-
F6	**4.43E+00** ± 2.72E+00	7.16E+00 ± 3.10E+00	-	5.44E+00 ± 2.99E+00	-	7.68E+00 ± 3.80E+00	-	5.24E+00 ± 2.75E+00	=
F7	**1.32E+02** ± 3.03E+01	1.75E+02 ± 4.72E+01	-	1.54E+02 ± 4.44E+01	=	1.49E+02 ± 3.47E+01	=	1.67E+02 ± 5.08E+01	-
F8	7.65E+01 ± 2.34E+01	7.85E+01 ± 2.49E+01	=	**7.24E+01** ± 1.73E+01	=	8.37E+01 ± 2.20E+01	=	8.20E+01 ± 3.29E+01	=
F9	4.95E+02 ± 3.89E+02	7.62E+02 ± 5.49E+02	-	5.10E+02 ± 3.22E+02	=	5.63E+02 ± 3.72E+02	-	6.44E+02 ± 4.52E+02	-
F10	3.18E+03 ± 9.33E+02	3.03E+03 ± 6.42E+02	+	3.20E+03 ± 9.88E+02	+	3.15E+03 ± 5.72E+02	=	3.34E+03 ± 1.03E+03	=
F11	4.12E+02 ± 3.93E+02	1.64E+02 ± 4.25E+01	+	1.60E+02 ± 3.44E+01	+	1.52E+02 ± 3.63E+01	+	1.68E+02 ± 3.63E+01	+
F12	3.28E+07 ± 3.58E+07	1.00E+07 ± 9.95E+06	+	8.91E+06 ± 6.79E+06	+	1.05E+07 ± 7.44E+06	+	1.08E+07 ± 1.02E+07	+
F13	3.74E+06 ± 1.92E+07	**1.03E+05** ± 4.37E+04	-	1.90E+05 ± 1.14E+05	-	1.24E+05 ± 3.57E+04	-	1.59E+05 ± 1.08E+05	-
F14	1.38E+05 ± 2.83E+05	2.68E+03 ± 2.70E+03	+	5.37E+03 ± 8.57E+03	+	4.10E+03 ± 4.05E+03	+	2.63E+03 ± 4.25E+03	+
F15	1.99E+05 ± 4.71E+05	2.34E+04 ± 1.32E+04	+	3.49E+04 ± 2.40E+04	+	2.54E+04 ± 1.70E+04	+	3.92E+04 ± 2.68E+04	=
F16	7.02E+02 ± 2.69E+02	9.20E+02 ± 3.29E+02	-	**6.72E+02** ± 2.14E+02	=	8.49E+02 ± 2.67E+02	-	8.08E+02 ± 3.00E+02	-
F17	2.53E+02 ± 1.17E+02	2.37E+02 ± 1.33E+02	=	2.23E+02 ± 1.16E+02	=	2.28E+02 ± 1.17E+02	=	2.34E+02 ± 1.20E+02	=
F18	5.46E+05 ± 7.05E+05	8.26E+04 ± 7.36E+04	+	9.56E+04 ± 6.60E+04	+	9.97E+04 ± 7.81E+04	+	1.38E+05 ± 1.69E+05	+
F19	6.00E+05 ± 6.49E+05	1.16E+05 ± 1.51E+05	+	1.08E+05 ± 1.43E+05	+	2.98E+05 ± 3.55E+05	+	1.31E+05 ± 1.71E+05	+
F20	3.47E+02 ± 1.27E+02	3.35E+02 ± 1.17E+02	=	**2.98E+02** ± 1.23E+02	=	3.32E+02 ± 1.24E+02	=	3.85E+02 ± 1.28E+02	-
F21	2.76E+02 ± 2.02E+01	2.95E+02 ± 3.79E+01	-	2.76E+02 ± 1.97E+01	=	2.76E+02 ± 2.02E+01	=	2.90E+02 ± 3.82E+01	=
F22	1.45E+03 ± 1.45E+03	**3.32E+02** ± 9.21E+02	+	1.69E+03 ± 1.75E+03	=	4.95E+02 ± 9.99E+02	+	1.81E+03 ± 2.18E+03	+
F23	4.38E+02 ± 2.75E+01	5.11E+02 ± 6.03E+01	-	**4.34E+02** ± 3.31E+01	=	4.83E+02 ± 3.65E+01	-	4.66E+02 ± 4.43E+01	-
F24	5.16E+02 ± 5.09E+01	5.64E+02 ± 6.10E+01	-	5.19E+02 ± 4.22E+01	=	5.43E+02 ± 4.90E+01	-	5.57E+02 ± 5.16E+01	-
F25	4.52E+02 ± 2.27E+01	4.34E+02 ± 1.95E+01	+	4.18E+02 ± 1.83E+01	+	4.37E+02 ± 2.13E+01	=	4.16E+02 ± 1.68E+01	+
F26	1.82E+03 ± 3.45E+02	1.96E+03 ± 8.11E+02	-	1.78E+03 ± 4.67E+02	=	1.90E+03 ± 8.53E+02	-	1.88E+03 ± 3.98E+02	=
F27	5.38E+02 ± 1.81E+01	5.72E+02 ± 3.19E+01	-	5.28E+02 ± 1.47E+01	+	5.64E+02 ± 3.20E+01	-	5.32E+02 ± 1.73E+01	+
F28	5.59E+02 ± 5.61E+01	4.82E+02 ± 4.10E+01	+	4.73E+02 ± 3.30E+01	+	4.92E+02 ± 4.26E+01	+	4.77E+02 ± 3.62E+01	+
F29	7.82E+02 ± 1.53E+02	8.43E+02 ± 1.85E+02	=	7.80E+02 ± 1.17E+02	=	8.09E+02 ± 1.68E+02	=	8.38E+02 ± 1.26E+02	=
F30	5.65E+06 ± 5.05E+06	1.63E+06 ± 1.15E+06	+	1.55E+06 ± 1.16E+06	+	2.12E+06 ± 1.26E+06	+	1.67E+06 ± 1.09E+06	+
W/T/L	–	14/4/11		14/13/2		12/8/9		13/8/8	

TABLE 8. Performance comparison among GWO, CGWOs, and MCGWO in terms of solution accuracy on IEEE CEC2017 with $D = 30$ (Continued 1).

	CGWO5		CGWO6		CGWO7		CGWO8		CGWO9	
	mean ± std		mean ± std		mean ± std		mean ± std		mean ± std	
F1	**4.52E+07** ± 5.92E+07	+	8.33E+07 ± 1.60E+08	+	6.46E+07 ± 9.55E+07	+	4.20E+08 ± 4.32E+08	+	1.64E+08 ± 1.68E+08	+
F3	**2.32E+03** ± 1.79E+03	+	3.57E+03 ± 2.28E+03	+	2.77E+03 ± 2.03E+03	+	1.43E+04 ± 8.09E+03	+	6.05E+03 ± 3.33E+03	+
F4	**1.06E+02** ± 2.06E+01	+	1.07E+02 ± 2.46E+01	+	1.15E+02 ± 1.83E+01	+	1.41E+02 ± 3.40E+01	+	1.21E+02 ± 2.22E+01	+
F5	9.16E+01 ± 3.13E+01	=	8.63E+01 ± 2.25E+01	=	9.73E+01 ± 3.19E+01	=	8.74E+01 ± 1.81E+01	=	8.46E+01 ± 2.05E+01	=
F6	5.26E+00 ± 3.45E+00	-	4.86E+00 ± 3.13E+00	=	5.43E+00 ± 3.25E+00	=	4.72E+00 ± 2.37E+00	=	5.29E+00 ± 2.10E+00	-
F7	1.66E+02 ± 4.10E+01	-	1.54E+02 ± 3.62E+01	=	1.55E+02 ± 3.45E+01	-	1.39E+02 ± 3.88E+01	=	1.41E+02 ± 3.78E+01	=
F8	7.42E+01 ± 2.34E+01	=	8.01E+01 ± 2.54E+01	=	7.48E+01 ± 2.14E+01	=	8.25E+01 ± 2.25E+01	=	7.78E+01 ± 1.91E+01	=
F9	6.22E+02 ± 4.39E+02	-	4.98E+02 ± 3.59E+02	=	4.91E+02 ± 3.43E+02	=	5.77E+02 ± 4.32E+02	=	5.81E+02 ± 4.30E+02	-
F10	**2.89E+03** ± 7.52E+02	+	3.03E+03 ± 8.49E+02	=	3.06E+03 ± 7.60E+02	=	3.19E+03 ± 9.45E+02	=	3.10E+03 ± 8.89E+02	=
F11	1.58E+02 ± 4.78E+01	+	1.55E+02 ± 3.89E+01	+	1.56E+02 ± 3.61E+01	+	2.02E+02 ± 4.72E+01	+	1.83E+02 ± 5.07E+01	+
F12	8.73E+06 ± 6.10E+06	+	1.22E+07 ± 1.08E+07	+	1.20E+07 ± 1.01E+07	+	1.84E+07 ± 1.80E+07	=	1.49E+07 ± 1.32E+07	+
F13	1.47E+05 ± 8.01E+04	-	1.79E+05 ± 7.72E+04	-	2.37E+05 ± 2.89E+05	-	5.42E+05 ± 2.80E+06	-	2.25E+05 ± 3.04E+05	-
F14	3.57E+03 ± 6.92E+03	+	2.53E+03 ± 2.33E+03	+	4.53E+03 ± 7.93E+03	+	2.56E+04 ± 2.86E+04	+	1.37E+04 ± 1.47E+04	+
F15	3.15E+04 ± 2.36E+04	+	4.14E+04 ± 3.24E+04	=	4.25E+04 ± 3.24E+04	=	4.31E+04 ± 4.10E+04	=	4.03E+04 ± 2.31E+04	=
F16	7.50E+02 ± 2.29E+02	-	7.49E+02 ± 2.67E+02	=	7.43E+02 ± 2.19E+02	=	7.65E+02 ± 2.60E+02	=	7.16E+02 ± 3.10E+02	=
F17	2.35E+02 ± 1.38E+02	=	2.40E+02 ± 1.27E+02	=	2.34E+02 ± 1.39E+02	=	**2.21E+02** ± 1.18E+02	=	2.31E+02 ± 1.29E+02	=
F18	7.68E+04 ± 5.42E+04	+	1.01E+05 ± 1.03E+05	+	8.78E+04 ± 7.75E+04	+	3.07E+05 ± 3.35E+05	=	1.68E+05 ± 1.23E+05	+
F19	**8.54E+04** ± 9.10E+04	+	1.19E+05 ± 1.34E+05	+	1.29E+05 ± 1.28E+05	+	4.50E+05 ± 1.07E+06	+	2.88E+05 ± 3.74E+05	+
F20	3.39E+02 ± 1.25E+02	=	3.36E+02 ± 1.26E+02	=	3.20E+02 ± 1.05E+02	=	3.12E+02 ± 1.26E+02	=	3.47E+02 ± 1.34E+02	=
F21	2.80E+02 ± 2.62E+01	=	2.76E+02 ± 2.21E+01	=	2.75E+02 ± 1.91E+01	=	2.75E+02 ± 2.13E+01	=	**2.69E+02** ± 1.66E+01	=
F22	1.40E+03 ± 1.59E+03	+	1.34E+03 ± 1.54E+03	+	1.32E+03 ± 1.54E+03	+	1.49E+03 ± 1.56E+03	=	1.64E+03 ± 1.72E+03	+
F23	4.53E+02 ± 3.61E+01	-	4.55E+02 ± 4.08E+01	-	4.41E+02 ± 3.30E+01	=	4.39E+02 ± 3.30E+01	=	4.40E+02 ± 3.62E+01	=
F24	5.36E+02 ± 4.33E+01	=	5.31E+02 ± 3.69E+01	-	5.29E+02 ± 4.35E+01	=	5.10E+02 ± 4.14E+01	=	5.23E+02 ± 4.61E+01	=
F25	**4.13E+02** ± 1.69E+01	+	4.17E+02 ± 1.63E+01	+	4.19E+02 ± 2.29E+01	+	4.44E+02 ± 2.68E+01	=	4.28E+02 ± 1.84E+01	+
F26	1.93E+03 ± 4.34E+02	-	1.87E+03 ± 4.18E+02	=	**1.70E+03** ± 5.20E+02	=	1.88E+03 ± 3.81E+02	-	1.82E+03 ± 4.49E+02	=
F27	5.30E+02 ± 1.30E+01	+	5.28E+02 ± 1.29E+01	+	**5.26E+02** ± 1.30E+01	+	5.32E+02 ± 1.67E+01	+	5.31E+02 ± 1.48E+01	=
F28	**4.70E+02** ± 3.51E+01	+	4.80E+02 ± 3.37E+01	+	4.80E+02 ± 3.54E+01	+	5.27E+02 ± 5.37E+01	+	4.98E+02 ± 2.98E+01	+
F29	8.07E+02 ± 1.34E+02	=	**7.55E+02** ± 1.15E+02	=	7.63E+02 ± 1.35E+02	=	7.89E+02 ± 1.43E+02	=	7.89E+02 ± 1.84E+02	=
F30	**1.20E+06** ± 8.35E+05	+	1.36E+06 ± 1.09E+06	+	1.44E+06 ± 1.02E+06	+	4.29E+06 ± 3.31E+06	=	2.97E+06 ± 2.23E+06	=
W/T/L	15/7/7		12/14/3		13/14/2		8/19/2		11/15/3	

i.e., CGWO2, CGWO5 and CGWO5, on the 30, 50 and 100 dimensions, respectively. It can be seen from the table that CGWOs outperform PSO, SCA and WFS on most of the benchmark functions, which indicates that CGWOs are more competitive than other classic algorithms after being incorporated by CLS.

In addition, Figs. 9 and 10 show the box-and-whisker diagrams and convergence graphs of CGWOs and other meta-heuristic algorithms, respectively. According to Fig. 9, the CGWO has the lowest altitude and relatively short distant than other heuristic algorithms, suggesting that it has a stronger local optimal jumping ability and better search

TABLE 9. Performance comparison among GWO, CGWOs, and MCGWO in terms of solution accuracy on IEEE CEC2017 with $D = 30$ (Continued 2).

	CGWO10		CGWO11		CGWO12		MCGWO	
	mean ± std		mean ± std		mean ± std		mean ± std	
F1	6.62E+07 ± 9.79E+07	+	9.43E+08 ± 9.21E+08	=	1.90E+08 ± 2.52E+08	+	5.60E+07 ± 6.47E+07	+
F3	2.59E+03 ± 1.75E+03	+	2.99E+04 ± 9.79E+03	=	6.62E+03 ± 4.51E+03	+	4.12E+03 ± 2.58E+03	+
F4	1.18E+02 ± 2.45E+01	+	1.46E+02 ± 3.39E+01	=	1.17E+02 ± 2.08E+01	+	1.14E+02 ± 1.86E+01	+
F5	9.93E+01 ± 2.80E+01	=	**8.18E+01** ± 1.68E+01	=	8.25E+01 ± 1.85E+01	=	9.28E+01 ± 2.82E+01	=
F6	7.94E+00 ± 4.46E+00	-	5.32E+00 ± 2.88E+00	=	4.75E+00 ± 2.82E+00	=	5.33E+00 ± 2.75E+00	-
F7	1.51E+02 ± 3.89E+01	=	1.50E+02 ± 4.68E+01	=	1.42E+02 ± 4.62E+01	=	1.51E+02 ± 4.25E+01	=
F8	7.66E+01 ± 1.62E+01	=	8.53E+01 ± 2.44E+01	=	7.67E+01 ± 1.66E+01	=	7.77E+01 ± 2.37E+01	=
F9	5.11E+02 ± 3.16E+02	=	5.67E+02 ± 4.40E+02	-	**4.86E+02** ± 3.34E+02	=	5.41E+02 ± 3.63E+02	-
F10	3.10E+03 ± 5.77E+02	=	2.93E+03 ± 6.74E+02	=	3.04E+03 ± 7.00E+02	=	3.29E+03 ± 1.10E+03	=
F11	**1.50E+02** ± 3.83E+01	+	3.74E+02 ± 2.67E+02	=	2.03E+02 ± 5.88E+01	+	1.52E+02 ± 3.65E+01	+
F12	9.26E+06 ± 7.18E+06	+	**3.70E+07** ± 5.49E+07	=	1.44E+07 ± 1.94E+07	+	1.25E+07 ± 1.14E+07	+
F13	1.24E+05 ± 2.50E+05	-	3.35E+06 ± 1.92E+07	-	2.29E+05 ± 4.00E+05	-	1.62E+05 ± 7.23E+04	-
F14	**1.66E+03** ± 2.27E+03	+	6.02E+04 ± 1.30E+05	+	1.84E+04 ± 1.99E+04	+	3.58E+03 ± 6.10E+03	+
F15	**2.30E+04** ± 1.20E+04	+	1.46E+05 ± 3.80E+05	=	3.15E+04 ± 1.95E+04	+	3.71E+04 ± 3.05E+04	=
F16	7.64E+02 ± 2.48E+02	=	7.71E+02 ± 2.98E+02	=	7.06E+02 ± 2.57E+02	=	7.32E+02 ± 2.64E+02	=
F17	2.68E+02 ± 1.24E+02	=	2.46E+02 ± 1.30E+02	=	2.33E+02 ± 1.30E+02	=	2.57E+02 ± 1.18E+02	=
F18	**7.54E+04** ± 5.44E+04	+	7.70E+05 ± 9.13E+05	-	2.13E+05 ± 1.75E+05	+	1.04E+05 ± 8.51E+04	+
F19	1.23E+05 ± 1.19E+05	+	5.15E+05 ± 1.18E+06	=	3.36E+05 ± 3.65E+05	+	1.22E+05 ± 1.44E+05	+
F20	3.16E+02 ± 1.01E+02	=	3.36E+02 ± 1.41E+02	=	2.98E+02 ± 1.23E+02	=	3.49E+02 ± 1.51E+02	=
F21	2.85E+02 ± 2.60E+01	-	2.72E+02 ± 1.82E+01	=	2.75E+02 ± 1.85E+01	=	2.83E+02 ± 2.66E+01	=
F22	4.03E+02 ± 1.06E+03	+	1.64E+03 ± 1.48E+03	=	1.47E+03 ± 1.61E+03	+	1.41E+03 ± 1.62E+03	+
F23	4.83E+02 ± 4.93E+01	-	4.44E+02 ± 4.15E+01	=	4.46E+02 ± 3.80E+01	=	4.62E+02 ± 4.43E+01	-
F24	5.39E+02 ± 5.05E+01	-	**5.09E+02** ± 4.10E+01	=	5.12E+02 ± 4.70E+01	=	5.39E+02 ± 4.69E+01	-
F25	4.30E+02 ± 2.45E+01	+	4.62E+02 ± 3.39E+01	=	4.26E+02 ± 1.64E+01	+	4.14E+02 ± 2.04E+01	+
F26	1.99E+03 ± 7.35E+02	-	1.84E+03 ± 3.58E+02	=	1.80E+03 ± 2.86E+02	=	1.92E+03 ± 5.16E+02	=
F27	5.54E+02 ± 2.15E+01	-	5.36E+02 ± 1.63E+01	=	5.35E+02 ± 1.65E+01	=	5.31E+02 ± 1.57E+01	+
F28	4.85E+02 ± 3.52E+01	+	5.50E+02 ± 7.51E+01	=	5.07E+02 ± 3.17E+01	+	4.90E+02 ± 3.51E+01	+
F29	7.70E+02 ± 1.53E+02	=	7.67E+02 ± 1.33E+02	=	7.66E+02 ± 1.78E+02	=	7.78E+02 ± 1.37E+02	=
F30	1.49E+06 ± 1.17E+06	+	3.77E+06 ± 2.91E+06	=	3.23E+06 ± 2.41E+06	=	1.36E+06 ± 8.88E+05	+
W/T/L	13/9/7		1/25/3		12/16/1		13/11/5	

TABLE 10. Performance comparison among GWO, CGWOs, and MCGWO in terms of solution accuracy on IEEE CEC2017 with $D = 50$.

	GWO	GWO1		GWO2		GWO3		GWO4	
	mean ± std	mean ± std		mean ± std		mean ± std		mean ± std	
F1	4.99E+09 ± 2.61E+09	8.17E+08 ± 8.36E+08	+	6.12E+08 ± 6.10E+08	+	6.73E+08 ± 5.97E+08	+	9.62E+08 ± 9.99E+08	+
F3	7.19E+04 ± 1.58E+04	1.81E+04 ± 6.49E+03	+	**1.36E+04** ± 6.67E+03	+	1.79E+04 ± 6.89E+03	+	2.29E+04 ± 7.82E+03	+
F4	4.57E+02 ± 1.83E+02	2.69E+02 ± 7.89E+01	+	2.78E+02 ± 8.12E+01	+	2.99E+02 ± 8.11E+01	+	3.26E+02 ± 9.87E+01	+
F5	**1.90E+02** ± 3.75E+01	2.25E+02 ± 7.71E+01	-	1.93E+02 ± 3.33E+01	-	2.15E+02 ± 4.44E+01	-	2.11E+02 ± 7.04E+01	=
F6	**1.07E+01** ± 3.01E+00	1.72E+01 ± 4.20E+00	-	1.77E+01 ± 4.31E+00	-	1.74E+01 ± 4.86E+00	-	1.98E+01 ± 7.08E+00	-
F7	**3.01E+02** ± 6.11E+01	3.58E+02 ± 8.86E+01	-	3.35E+02 ± 6.35E+01	-	3.38E+02 ± 7.28E+01	-	3.61E+02 ± 1.05E+02	-
F8	**1.99E+02** ± 4.68E+01	2.24E+02 ± 6.44E+01	=	2.08E+02 ± 3.33E+01	=	2.20E+02 ± 5.26E+01	-	2.26E+02 ± 6.50E+01	-
F9	**4.19E+03** ± 1.87E+03	7.16E+03 ± 2.48E+03	-	6.06E+03 ± 2.46E+03	-	6.11E+03 ± 2.32E+03	-	7.54E+03 ± 2.64E+03	-
F10	5.79E+03 ± 8.45E+02	6.28E+03 ± 1.94E+03	=	5.68E+03 ± 9.02E+02	=	5.63E+03 ± 1.04E+03	=	6.54E+03 ± 2.13E+03	=
F11	1.79E+03 ± 1.15E+03	3.40E+02 ± 9.64E+01	+	3.50E+02 ± 1.08E+02	+	3.70E+02 ± 1.15E+02	+	3.63E+02 ± 1.15E+02	+
F12	3.85E+08 ± 5.98E+08	6.72E+07 ± 5.02E+07	+	7.77E+07 ± 6.58E+07	+	8.41E+07 ± 6.06E+07	+	9.35E+07 ± 5.78E+07	+
F13	9.06E+07 ± 1.31E+08	1.90E+06 ± 3.15E+06	+	1.41E+06 ± 1.69E+06	+	1.40E+06 ± 1.72E+06	+	2.91E+06 ± 3.58E+06	+
F14	4.25E+05 ± 4.26E+05	4.62E+04 ± 4.18E+04	+	2.34E+04 ± 2.40E+04	+	4.08E+04 ± 4.59E+04	+	4.98E+04 ± 5.11E+04	+
F15	6.73E+06 ± 1.31E+07	6.90E+04 ± 4.58E+04	=	5.04E+04 ± 2.81E+04	+	6.21E+04 ± 2.41E+04	=	7.78E+04 ± 5.51E+04	=
F16	1.33E+03 ± 4.16E+02	1.37E+03 ± 4.24E+02	=	**1.12E+03** ± 2.80E+02	+	1.29E+03 ± 3.19E+02	=	1.42E+03 ± 5.23E+02	=
F17	9.15E+02 ± 1.91E+02	9.99E+02 ± 2.75E+02	=	9.62E+02 ± 2.12E+02	=	1.02E+03 ± 2.38E+02	-	1.04E+03 ± 3.33E+02	=
F18	2.18E+06 ± 2.22E+06	4.35E+05 ± 3.46E+05	+	3.07E+05 ± 2.70E+05	+	4.03E+05 ± 2.80E+05	+	4.93E+05 ± 3.92E+05	+
F19	1.99E+06 ± 4.52E+06	7.92E+05 ± 5.21E+05	=	7.23E+05 ± 4.05E+05	+	6.71E+05 ± 4.05E+05	+	8.14E+05 ± 5.89E+05	=
F20	7.31E+02 ± 2.54E+02	7.49E+02 ± 2.60E+02	=	7.50E+02 ± 2.37E+02	=	7.26E+02 ± 2.54E+02	=	7.47E+02 ± 2.45E+02	=
F21	**3.86E+02** ± 3.00E+01	4.33E+02 ± 8.31E+01	-	4.03E+02 ± 5.05E+01	=	4.06E+02 ± 6.00E+01	=	4.36E+02 ± 8.40E+01	-
F22	5.69E+03 ± 1.53E+03	5.68E+03 ± 2.94E+03	=	5.64E+03 ± 3.07E+03	=	5.41E+03 ± 3.24E+03	=	6.34E+03 ± 3.01E+03	-
F23	**6.27E+02** ± 5.96E+01	7.62E+02 ± 9.84E+01	-	7.35E+02 ± 8.16E+01	-	7.57E+02 ± 1.06E+02	-	7.91E+02 ± 2.32E+02	-
F24	**7.13E+02** ± 1.00E+02	8.60E+02 ± 2.19E+02	-	8.20E+02 ± 9.00E+01	-	8.27E+02 ± 9.15E+01	-	8.90E+02 ± 2.13E+02	-
F25	8.80E+02 ± 1.88E+02	7.19E+02 ± 8.91E+01	+	7.23E+02 ± 1.03E+02	+	7.24E+02 ± 8.77E+01	+	7.26E+02 ± 1.19E+02	+
F26	**3.30E+03** ± 4.96E+02	4.17E+03 ± 1.15E+03	-	4.47E+03 ± 8.61E+02	-	3.94E+03 ± 1.27E+03	-	4.34E+03 ± 8.98E+02	-
F27	8.06E+02 ± 1.35E+02	9.29E+02 ± 1.24E+02	-	9.48E+02 ± 3.26E+02	-	8.76E+02 ± 1.07E+02	-	9.57E+02 ± 1.36E+02	-
F28	1.09E+03 ± 2.84E+02	8.00E+02 ± 1.32E+02	+	7.62E+02 ± 1.57E+02	+	8.36E+02 ± 1.72E+02	+	8.09E+02 ± 1.73E+02	+
F29	**1.27E+03** ± 2.17E+02	1.61E+03 ± 2.97E+02	-	1.53E+03 ± 3.58E+02	-	1.59E+03 ± 2.98E+02	-	1.60E+03 ± 4.10E+02	-
F30	6.51E+07 ± 2.21E+07	3.55E+07 ± 1.24E+07	+	3.53E+07 ± 1.09E+07	+	4.11E+07 ± 1.36E+07	+	3.75E+07 ± 1.63E+07	+
W/T/L	–	11/8/10		14/6/9		12/6/11		11/7/11	

performance. From Fig. 10, it is apparent that CGWO obtains a faster convergence speed in comparison with its peers.

E. ANALYSIS OF POPULATION DISTRIBUTION

To give more insights into the search dynamics of CLS incorporated GWO, the convergence trajectories of both CGWO and GWO on the benchmark function F1 with two dimensions are illustrated in Fig. 8. In it, the contour map of F1 together with its global optimal solution is illustrated. Clearly, F1 is a unimodal function, and it is so simple and illuminating to depict the search dynamics of GWO and CWGOs. Accordingly, three typical solution distribution states at the early,

TABLE 11. Performance comparison among GWO, CGWOs, and MCGWO in terms of solution accuracy on IEEE CEC2017 with $D = 50$ (Continued 1).

	CGWO5		CGWO6		CGWO7		CGWO8		CGWO9	
	mean ± std		mean ± std		mean ± std		mean ± std		mean ± std	
F1	**3.16E+08** ± 4.58E+08	+	7.42E+08 ± 8.12E+08	+	4.98E+08 ± 5.38E+08	+	1.81E+09 ± 1.61E+09	+	9.62E+08 ± 8.19E+08	+
F3	1.46E+04 ± 6.04E+03	+	1.54E+04 ± 5.76E+03	+	1.39E+04 ± 5.28E+03	+	3.48E+04 ± 1.09E+04	+	1.91E+04 ± 7.07E+03	+
F4	**2.07E+02** ± 3.88E+01	+	2.88E+02 ± 8.13E+01	+	2.76E+02 ± 1.02E+02	+	4.05E+02 ± 1.99E+02	=	3.35E+02 ± 9.60E+01	+
F5	2.15E+02 ± 7.49E+01	-	2.07E+02 ± 4.38E+01	-	2.03E+02 ± 5.37E+01	=	2.07E+02 ± 4.15E+01	-	2.07E+02 ± 3.82E+01	-
F6	1.08E+01 ± 3.47E+00	=	1.62E+01 ± 5.61E+00	-	1.72E+01 ± 5.26E+00	-	1.79E+01 ± 4.61E+00	-	1.87E+01 ± 5.88E+00	-
F7	3.67E+02 ± 9.13E+01	-	3.43E+02 ± 8.82E+01	-	3.47E+02 ± 8.44E+01	-	3.12E+02 ± 5.83E+01	-	3.01E+02 ± 6.65E+01	=
F8	1.99E+02 ± 5.20E+01	=	2.11E+02 ± 5.29E+01	=	1.99E+02 ± 4.22E+01	=	2.16E+02 ± 3.18E+01	-	2.08E+02 ± 4.24E+01	=
F9	4.73E+03 ± 1.51E+03	-	6.26E+03 ± 2.26E+03	-	6.51E+03 ± 2.80E+03	-	5.79E+03 ± 2.60E+03	-	5.57E+03 ± 3.07E+03	=
F10	5.81E+03 ± 1.12E+03	=	5.67E+03 ± 1.04E+03	=	5.78E+03 ± 1.40E+03	=	6.05E+03 ± 1.54E+03	=	5.90E+03 ± 1.67E+03	=
F11	3.34E+02 ± 6.98E+01	+	3.46E+02 ± 8.78E+01	+	3.59E+02 ± 1.79E+02	+	5.24E+02 ± 2.06E+02	+	4.38E+02 ± 1.40E+02	+
F12	6.00E+07 ± 3.69E+07	+	7.77E+07 ± 5.61E+07	+	8.06E+07 ± 5.34E+07	+	2.06E+08 ± 1.45E+08	+	9.56E+07 ± 6.95E+07	+
F13	**9.35E+05** ± 8.65E+05	+	1.16E+06 ± 1.38E+06	+	1.21E+06 ± 1.40E+06	+	2.20E+07 ± 3.88E+07	+	3.97E+06 ± 5.39E+06	+
F14	3.62E+04 ± 4.16E+04	+	3.78E+04 ± 3.35E+04	+	2.50E+04 ± 3.05E+04	+	1.21E+05 ± 8.84E+04	+	4.99E+04 ± 5.77E+04	+
F15	7.80E+04 ± 4.88E+04	=	6.41E+04 ± 3.82E+04	=	5.03E+04 ± 2.52E+04	=	1.02E+06 ± 2.70E+06	=	8.99E+04 ± 6.07E+04	=
F16	1.27E+03 ± 3.84E+02	=	1.26E+03 ± 3.77E+02	=	1.21E+03 ± 3.18E+02	=	1.29E+03 ± 3.03E+02	=	1.24E+03 ± 3.04E+02	=
F17	1.05E+03 ± 2.58E+02	-	9.79E+02 ± 2.77E+02	=	9.45E+02 ± 2.52E+02	=	9.71E+02 ± 2.89E+02	=	9.46E+02 ± 2.55E+02	=
F18	3.43E+05 ± 2.34E+05	+	3.14E+05 ± 2.64E+05	+	3.60E+05 ± 3.90E+05	+	1.19E+06 ± 7.47E+05	=	5.30E+05 ± 4.03E+05	+
F19	**5.85E+05** ± 3.85E+05	+	6.56E+05 ± 4.17E+05	+	6.33E+05 ± 3.68E+05	+	8.37E+05 ± 6.30E+05	=	7.61E+05 ± 5.61E+05	=
F20	8.08E+02 ± 2.53E+02	=	7.20E+02 ± 2.28E+02	=	6.85E+02 ± 2.20E+02	=	**6.57E+02** ± 2.24E+02	+	7.38E+02 ± 2.34E+02	=
F21	4.04E+02 ± 5.85E+01	=	4.02E+02 ± 5.08E+01	=	3.98E+02 ± 5.38E+01	=	3.94E+02 ± 2.53E+01	=	3.95E+02 ± 3.37E+01	=
F22	5.72E+03 ± 2.26E+03	=	**4.82E+03** ± 3.03E+03	=	4.95E+03 ± 3.11E+03	=	5.23E+03 ± 3.02E+03	=	6.07E+03 ± 2.52E+03	-
F23	6.60E+02 ± 9.40E+01	-	8.19E+02 ± 2.25E+02	-	7.40E+02 ± 6.49E+01	-	7.34E+02 ± 5.85E+01	-	7.38E+02 ± 7.69E+01	-
F24	7.53E+02 ± 9.08E+01	-	8.07E+02 ± 8.69E+01	-	7.97E+02 ± 8.72E+01	-	8.26E+02 ± 2.24E+02	-	7.72E+02 ± 7.24E+01	-
F25	**6.22E+02** ± 6.73E+01	+	7.17E+02 ± 8.21E+01	+	7.14E+02 ± 7.34E+01	+	8.87E+02 ± 1.55E+02	=	7.73E+02 ± 1.10E+02	+
F26	3.79E+03 ± 8.88E+02	-	4.12E+03 ± 8.62E+02	-	4.23E+03 ± 9.47E+02	-	4.53E+03 ± 7.22E+02	-	4.24E+03 ± 8.84E+02	-
F27	7.24E+02 ± 8.00E+01	+	9.04E+02 ± 1.39E+02	-	9.06E+02 ± 1.70E+02	-	8.63E+02 ± 1.20E+02	-	8.42E+02 ± 1.39E+02	-
F28	6.22E+02 ± 8.24E+01	+	8.11E+02 ± 1.74E+02	+	7.72E+02 ± 1.45E+02	+	9.74E+02 ± 1.90E+02	=	8.63E+02 ± 1.89E+02	+
F29	1.33E+03 ± 3.62E+02	=	1.52E+03 ± 3.26E+02	-	1.53E+03 ± 3.33E+02	-	1.51E+03 ± 3.35E+02	-	1.55E+03 ± 3.10E+02	-
F30	**2.33E+07** ± 7.35E+06	+	3.71E+07 ± 1.57E+07	+	3.35E+07 ± 1.28E+07	+	7.01E+07 ± 1.98E+07	=	4.17E+07 ± 1.44E+07	+
W/T/L	13/9/7		12/8/9		12/9/8		6/14/9		11/10/8	

TABLE 12. Performance comparison among GWO, CGWOs, and MCGWO in terms of solution accuracy on IEEE CEC2017 with $D = 50$ (Continued 2).

	CGWO10		CGWO11		CGWO12		MCGWO	
	mean ± std		mean ± std		mean ± std		mean ± std	
F1	4.51E+08 ± 4.15E+08	+	7.14E+09 ± 3.18E+09	=	8.09E+08 ± 7.06E+08	+	5.36E+08 ± 6.18E+08	+
F3	1.49E+04 ± 5.88E+03	+	7.35E+04 ± 1.35E+04	=	1.83E+04 ± 6.70E+03	+	1.91E+04 ± 8.11E+03	+
F4	2.75E+02 ± 9.07E+01	+	1.03E+03 ± 4.92E+02	-	3.07E+02 ± 1.16E+02	+	2.14E+02 ± 6.33E+01	+
F5	2.07E+02 ± 3.95E+01	-	2.09E+02 ± 3.12E+01	-	2.09E+02 ± 3.13E+01	-	2.06E+02 ± 7.01E+01	=
F6	1.62E+01 ± 5.92E+00	-	1.81E+01 ± 5.14E+00	-	1.83E+01 ± 5.23E+00	-	1.21E+01 ± 4.48E+00	=
F7	3.32E+02 ± 6.57E+01	-	3.10E+02 ± 5.92E+01	=	3.03E+02 ± 4.38E+01	=	3.88E+02 ± 8.27E+01	-
F8	2.07E+02 ± 3.79E+01	=	2.08E+02 ± 3.46E+01	=	2.22E+02 ± 4.15E+01	-	2.05E+02 ± 5.56E+01	=
F9	4.73E+03 ± 2.15E+03	=	5.14E+03 ± 2.64E+03	=	5.58E+03 ± 3.17E+03	-	5.66E+03 ± 2.61E+03	-
F10	**5.63E+03** ± 7.97E+02	=	5.98E+03 ± 6.75E+02	=	5.90E+03 ± 9.80E+02	=	6.10E+03 ± 1.32E+03	=
F11	**3.22E+02** ± 9.11E+01	+	2.40E+03 ± 1.47E+03	-	4.29E+02 ± 1.54E+02	+	3.31E+02 ± 7.93E+01	+
F12	7.64E+07 ± 5.24E+07	+	6.83E+08 ± 8.09E+08	-	1.25E+08 ± 1.01E+08	+	**5.94E+07** ± 4.86E+07	+
F13	1.44E+06 ± 1.83E+06	+	1.42E+08 ± 1.15E+08	-	3.90E+06 ± 5.57E+06	+	1.00E+06 ± 9.70E+05	+
F14	**2.03E+04** ± 2.05E+04	+	6.35E+05 ± 9.72E+05	=	4.89E+04 ± 5.04E+04	+	5.11E+04 ± 4.29E+04	+
F15	**4.08E+04** ± 2.06E+04	+	4.08E+06 ± 7.35E+06	=	8.29E+04 ± 4.65E+04	=	8.08E+04 ± 4.41E+04	=
F16	1.17E+03 ± 2.78E+02	=	1.40E+03 ± 3.68E+02	=	1.22E+03 ± 3.13E+02	=	1.21E+03 ± 3.46E+02	=
F17	9.83E+02 ± 2.76E+02	=	9.30E+02 ± 2.33E+02	=	**9.13E+02** ± 2.40E+02	=	9.56E+02 ± 2.42E+02	=
F18	**2.81E+05** ± 1.88E+05	+	2.96E+06 ± 4.13E+06	=	5.02E+05 ± 2.80E+05	+	3.57E+05 ± 2.02E+05	+
F19	7.14E+05 ± 4.53E+05	=	1.78E+06 ± 5.54E+06	=	7.41E+05 ± 4.80E+05	=	6.30E+05 ± 4.11E+05	+
F20	7.08E+02 ± 2.40E+02	=	7.06E+02 ± 2.29E+02	=	6.74E+02 ± 2.13E+02	=	7.87E+02 ± 2.71E+02	=
F21	3.90E+02 ± 3.32E+01	=	4.08E+02 ± 4.47E+01	-	3.94E+02 ± 2.08E+01	=	4.16E+02 ± 7.38E+01	=
F22	5.29E+03 ± 2.97E+03	=	6.32E+03 ± 1.88E+03	-	5.03E+03 ± 3.20E+03	=	6.59E+03 ± 2.11E+03	-
F23	7.71E+02 ± 1.92E+02	-	7.66E+02 ± 8.60E+01	-	7.23E+02 ± 5.87E+01	-	6.81E+02 ± 9.13E+01	-
F24	8.18E+02 ± 8.98E+01	-	8.66E+02 ± 2.31E+02	-	8.01E+02 ± 6.99E+01	-	7.75E+02 ± 9.96E+01	-
F25	7.31E+02 ± 8.48E+01	+	1.20E+03 ± 2.75E+02	-	7.75E+02 ± 1.03E+02	+	**6.19E+02** ± 5.92E+01	+
F26	4.05E+03 ± 9.41E+02	-	4.48E+03 ± 6.12E+02	-	4.44E+03 ± 8.02E+02	-	3.71E+03 ± 7.92E+02	-
F27	9.37E+02 ± 1.38E+02	-	1.06E+03 ± 1.18E+02	-	8.35E+02 ± 9.88E+01	-	**7.22E+02** ± 5.71E+01	+
F28	7.41E+02 ± 1.28E+02	+	1.53E+03 ± 3.23E+02	-	8.39E+02 ± 1.72E+02	+	6.23E+02 ± 8.67E+01	+
F29	1.56E+03 ± 3.81E+02	+	1.57E+03 ± 2.78E+02	-	1.45E+03 ± 3.56E+02	-	1.34E+03 ± 3.40E+02	=
F30	3.31E+07 ± 1.29E+07	+	1.04E+08 ± 4.03E+07	-	4.99E+07 ± 1.93E+07	+	2.50E+07 ± 8.64E+06	+
W/T/L	12/9/8		0/13/16		11/9/9		13/10/6	

middle, and later search process are given, to comparatively indicate the search dynamics of the algorithms.

From it, we can clearly observe that CGWO has two significant differences in comparison with GWO:

(1) At the beginning of the iteration (e.g., at the 1000th number of function evaluation (NFE)), CLS obviously expanded the population distribution range, proving that CLS has enabled GWO to possess an improved population diversity, and thus to be more capable to escape from the local region.

(2) In the latter part of the iteration, due to the quick convergence characteristic of GWO, the population

TABLE 13. Performance comparison among GWO, CGWOs, and MCGWO in terms of solution accuracy on IEEE CEC2017 with $D = 100$.

	GWO	GWO1		GWO2		GWO3		GWO4	
	mean ± std	mean ± std		mean ± std		mean ± std		mean ± std	
F1	3.05E+10 ± 7.66E+09	9.49E+09 ± 3.61E+09	+	8.37E+09 ± 4.50E+09	+	9.22E+09 ± 4.00E+09	+	1.04E+10 ± 5.18E+09	+
F3	2.06E+05 ± 1.97E+04	1.06E+05 ± 2.18E+04	+	**8.18E+04** ± 1.28E+04	+	9.17E+04 ± 1.65E+04	+	1.15E+05 ± 2.02E+04	+
F4	2.59E+03 ± 8.32E+02	1.45E+03 ± 4.21E+02	+	1.48E+03 ± 4.85E+02	+	1.48E+03 ± 4.81E+02	+	1.65E+03 ± 5.56E+02	+
F5	**5.42E+02** ± 6.07E+01	5.87E+02 ± 1.16E+02	-	6.12E+02 ± 1.27E+02	=	5.95E+02 ± 6.08E+01	-	6.08E+02 ± 1.60E+02	-
F6	**2.76E+01** ± 4.05E+00	3.61E+01 ± 5.26E+00	-	3.79E+01 ± 5.32E+00	-	3.74E+01 ± 4.34E+00	-	3.63E+01 ± 4.45E+00	-
F7	1.04E+03 ± 9.37E+01	1.05E+03 ± 2.09E+02	=	1.12E+03 ± 2.20E+02	=	1.10E+03 ± 1.65E+02	-	1.09E+03 ± 1.99E+02	=
F8	**5.69E+02** ± 6.52E+01	5.77E+02 ± 6.97E+01	=	6.24E+02 ± 1.19E+02	-	6.26E+02 ± 1.12E+02	-	5.80E+02 ± 9.61E+01	=
F9	**2.10E+04** ± 1.03E+04	3.42E+04 ± 9.38E+03	-	3.26E+04 ± 6.33E+03	-	3.42E+04 ± 6.32E+03	-	3.68E+04 ± 6.69E+03	-
F10	1.42E+04 ± 2.72E+03	1.39E+04 ± 2.13E+03	=	**1.38E+04** ± 1.51E+03	=	1.50E+04 ± 3.75E+03	=	1.69E+04 ± 6.17E+03	=
F11	3.78E+04 ± 9.65E+03	8.60E+03 ± 4.21E+03	+	5.12E+03 ± 2.23E+03	+	6.19E+03 ± 2.60E+03	+	9.58E+03 ± 4.68E+03	+
F12	4.68E+09 ± 3.09E+09	9.60E+08 ± 6.58E+08	+	7.26E+08 ± 4.25E+08	+	7.71E+08 ± 5.06E+08	+	1.06E+09 ± 6.92E+08	+
F13	5.17E+08 ± 5.07E+08	1.11E+07 ± 2.32E+07	+	6.40E+06 ± 9.03E+06	+	9.01E+06 ± 1.46E+07	+	1.69E+07 ± 2.91E+07	+
F14	4.31E+06 ± 3.53E+06	7.14E+05 ± 3.06E+05	+	**4.15E+05** ± 1.93E+05	+	6.51E+05 ± 3.69E+05	+	8.42E+05 ± 5.56E+05	+
F15	6.86E+07 ± 1.24E+08	2.98E+06 ± 5.86E+06	+	1.23E+06 ± 1.45E+06	+	1.30E+06 ± 1.86E+06	+	3.19E+06 ± 6.92E+06	+
F16	**3.85E+03** ± 5.26E+02	4.16E+03 ± 9.74E+02	-	4.22E+03 ± 1.04E+03	=	3.95E+03 ± 1.07E+03	=	4.27E+03 ± 1.09E+03	-
F17	**2.77E+03** ± 5.24E+02	3.06E+03 ± 7.91E+02	-	3.01E+03 ± 7.52E+02	=	3.02E+03 ± 7.20E+02	=	2.91E+03 ± 7.06E+02	=
F18	3.91E+06 ± 2.60E+06	7.72E+05 ± 3.42E+05	+	**6.52E+05** ± 2.97E+05	+	8.02E+05 ± 4.07E+05	+	8.84E+05 ± 3.93E+05	+
F19	7.03E+07 ± 1.05E+08	5.03E+06 ± 3.23E+06	+	5.17E+06 ± 2.68E+06	+	5.58E+06 ± 5.27E+06	+	6.42E+06 ± 7.49E+06	+
F20	2.40E+03 ± 6.84E+02	3.21E+03 ± 8.86E+02	-	2.36E+03 ± 6.51E+02	=	2.45E+03 ± 7.35E+02	=	2.78E+03 ± 8.03E+02	=
F21	**7.63E+02** ± 5.94E+01	8.82E+02 ± 1.55E+02	-	8.59E+02 ± 1.05E+02	-	8.68E+02 ± 8.71E+01	-	9.18E+02 ± 2.05E+02	-
F22	1.63E+04 ± 2.99E+03	1.77E+04 ± 4.88E+03	=	1.65E+04 ± 3.34E+03	=	1.63E+04 ± 4.68E+03	=	1.85E+04 ± 5.98E+03	=
F23	1.17E+03 ± 3.88E+02	1.50E+03 ± 4.95E+02	-	1.42E+03 ± 2.35E+02	-	1.46E+03 ± 4.01E+02	-	1.58E+03 ± 5.11E+02	-
F24	1.64E+03 ± 8.41E+02	2.21E+03 ± 7.96E+02	-	2.04E+03 ± 2.33E+02	-	2.25E+03 ± 8.96E+02	-	2.23E+03 ± 3.48E+02	-
F25	2.70E+03 ± 5.21E+02	1.89E+03 ± 2.74E+02	+	1.83E+03 ± 2.86E+02	+	1.96E+03 ± 3.66E+02	+	2.07E+03 ± 3.15E+02	+
F26	**1.02E+04** ± 1.04E+03	1.45E+04 ± 1.96E+03	-	1.51E+04 ± 2.26E+03	-	1.52E+04 ± 3.14E+03	-	1.46E+04 ± 1.81E+03	-
F27	1.14E+03 ± 1.04E+02	1.38E+03 ± 1.62E+02	-	1.37E+03 ± 2.25E+02	-	1.46E+03 ± 9.46E+02	-	1.44E+03 ± 1.66E+02	-
F28	3.86E+03 ± 1.10E+03	2.18E+03 ± 5.84E+02	+	2.07E+03 ± 4.74E+02	+	2.22E+03 ± 6.98E+02	+	2.50E+03 ± 6.06E+02	+
F29	4.50E+03 ± 5.95E+02	4.97E+03 ± 7.83E+02	-	5.11E+03 ± 6.15E+02	-	5.02E+03 ± 9.73E+02	-	4.98E+03 ± 5.28E+02	-
F30	3.40E+08 ± 2.80E+08	8.74E+07 ± 5.19E+07	+	8.23E+07 ± 4.08E+07	+	8.72E+07 ± 4.66E+07	+	8.01E+07 ± 4.24E+07	+
W/T/L	–	13/4/12		13/7/9		13/5/11		13/6/10	

TABLE 14. Performance comparison among GWO, CGWOs, and MCGWO in terms of solution accuracy on IEEE CEC2017 with $D = 100$ (Continued 1).

	CGWO5		CGWO6		CGWO7		CGWO8		CGWO9	
	mean ± std		mean ± std		mean ± std		mean ± std		mean ± std	
F1	5.43E+09 ± 2.94E+09	+	8.16E+09 ± 4.16E+09	+	7.80E+09 ± 3.65E+09	+	1.30E+10 ± 5.70E+09	+	1.01E+10 ± 5.19E+09	+
F3	8.82E+04 ± 1.51E+04	+	8.61E+04 ± 1.40E+04	+	8.78E+04 ± 1.81E+04	+	1.09E+05 ± 2.47E+04	+	9.41E+04 ± 1.71E+04	+
F4	**7.37E+02** ± 1.68E+02	+	1.35E+03 ± 3.56E+02	+	1.39E+03 ± 4.32E+02	+	2.08E+03 ± 7.25E+02	+	1.53E+03 ± 5.12E+02	+
F5	5.77E+02 ± 1.51E+02	=	5.73E+02 ± 5.43E+01	-	5.83E+02 ± 1.12E+02	-	6.02E+02 ± 6.34E+01	-	5.89E+02 ± 5.06E+01	-
F6	2.78E+01 ± 5.13E+00	=	3.71E+01 ± 5.11E+00	-	3.62E+01 ± 4.32E+00	-	3.73E+01 ± 5.10E+00	-	3.64E+01 ± 5.24E+00	-
F7	1.10E+03 ± 2.23E+02	=	1.12E+03 ± 2.19E+02	-	1.08E+03 ± 2.29E+02	=	1.04E+03 ± 8.82E+01	=	1.08E+03 ± 1.74E+02	=
F8	5.98E+02 ± 1.66E+02	=	5.82E+02 ± 1.01E+02	=	6.08E+02 ± 1.19E+02	=	6.36E+02 ± 7.51E+01	-	6.05E+02 ± 6.04E+01	-
F9	2.97E+04 ± 6.73E+03	-	3.37E+04 ± 6.41E+03	-	3.25E+04 ± 6.70E+03	-	2.76E+04 ± 1.07E+04	-	2.88E+04 ± 8.70E+03	-
F10	1.49E+04 ± 3.51E+03	=	1.43E+04 ± 3.30E+03	=	1.56E+04 ± 4.29E+03	-	1.47E+04 ± 1.66E+03	-	1.40E+04 ± 1.88E+03	=
F11	**4.14E+03** ± 1.44E+03	+	6.27E+03 ± 2.58E+03	+	5.89E+03 ± 2.39E+03	+	8.95E+03 ± 4.41E+03	+	6.07E+03 ± 2.41E+03	+
F12	**4.61E+08** ± 2.66E+08	+	6.83E+08 ± 5.46E+08	+	8.64E+08 ± 9.10E+08	+	1.37E+09 ± 8.29E+08	+	7.42E+08 ± 4.43E+08	+
F13	9.42E+06 ± 1.88E+07	+	6.43E+06 ± 7.76E+06	+	4.79E+06 ± 3.80E+06	+	2.14E+07 ± 4.06E+07	+	**4.76E+06** ± 5.52E+06	+
F14	5.21E+05 ± 2.60E+05	+	6.45E+05 ± 3.23E+05	+	6.25E+05 ± 4.07E+05	+	6.80E+05 ± 3.29E+05	+	5.57E+05 ± 2.48E+05	+
F15	1.38E+06 ± 1.82E+06	+	1.78E+06 ± 3.41E+06	+	**8.34E+05** ± 9.76E+05	+	2.17E+06 ± 2.68E+06	+	1.48E+06 ± 2.12E+06	+
F16	4.00E+03 ± 1.05E+03	=	4.40E+03 ± 1.47E+03	=	4.28E+03 ± 1.15E+03	=	3.92E+03 ± 6.16E+02	=	4.08E+03 ± 1.02E+03	=
F17	2.87E+03 ± 6.46E+02	=	2.93E+03 ± 7.51E+02	=	2.87E+03 ± 6.79E+02	=	2.86E+03 ± 5.78E+02	=	2.83E+03 ± 6.37E+02	=
F18	9.19E+05 ± 5.09E+05	+	7.73E+05 ± 3.91E+05	+	8.51E+05 ± 3.95E+05	+	1.12E+06 ± 5.32E+05	+	7.26E+05 ± 3.61E+05	+
F19	4.66E+06 ± 2.13E+06	+	4.87E+06 ± 2.64E+06	+	5.28E+06 ± 2.66E+06	+	5.24E+06 ± 3.07E+06	+	5.18E+06 ± 3.53E+06	+
F20	2.92E+03 ± 7.26E+02	-	2.68E+03 ± 7.66E+02	=	2.47E+03 ± 7.34E+02	=	2.40E+03 ± 5.54E+02	=	2.43E+03 ± 5.52E+02	=
F21	8.24E+02 ± 1.84E+02	=	8.62E+02 ± 1.33E+02	-	8.67E+02 ± 7.75E+01	-	8.53E+02 ± 7.46E+01	-	8.56E+02 ± 7.59E+01	-
F22	1.76E+04 ± 4.46E+03	=	1.62E+04 ± 5.05E+03	=	1.62E+04 ± 5.22E+03	=	1.63E+04 ± 4.73E+03	=	1.69E+04 ± 4.49E+03	=
F23	1.10E+03 ± 1.01E+02	+	1.36E+03 ± 1.29E+02	-	1.51E+03 ± 4.70E+02	-	1.63E+03 ± 8.15E+02	-	1.38E+03 ± 1.59E+02	-
F24	1.62E+03 ± 2.09E+02	=	2.04E+03 ± 3.44E+02	-	2.15E+03 ± 3.25E+02	-	1.98E+03 ± 2.40E+02	-	1.97E+03 ± 1.97E+02	-
F25	**1.41E+03** ± 1.60E+02	+	1.91E+03 ± 2.82E+02	+	1.85E+03 ± 3.14E+02	+	2.25E+03 ± 3.06E+02	+	1.99E+03 ± 3.02E+02	+
F26	1.03E+04 ± 1.71E+03	=	1.49E+04 ± 2.01E+03	-	1.45E+04 ± 2.29E+03	-	1.52E+04 ± 2.48E+03	-	1.48E+04 ± 2.56E+03	-
F27	**1.01E+03** ± 9.13E+01	+	1.34E+03 ± 2.14E+02	-	1.32E+03 ± 2.28E+02	-	1.19E+03 ± 1.40E+02	-	1.23E+03 ± 1.63E+02	-
F28	**1.22E+03** ± 2.30E+02	+	2.21E+03 ± 7.64E+02	+	2.15E+03 ± 6.28E+02	+	2.74E+03 ± 7.55E+02	+	2.17E+03 ± 5.39E+02	+
F29	**4.41E+03** ± 6.15E+02	=	4.96E+03 ± 5.98E+02	-	5.21E+03 ± 8.36E+02	-	5.03E+03 ± 5.86E+02	-	4.86E+03 ± 5.85E+02	-
F30	**5.32E+07** ± 2.90E+07	+	8.35E+07 ± 4.39E+07	+	7.41E+07 ± 3.93E+07	+	1.25E+08 ± 6.07E+07	+	8.58E+07 ± 4.37E+07	+
W/T/L	15/12/2		13/6/10		13/6/10		13/5/11		13/6/10	

of both CGWO and GWO can tend to be distributed around the global optimum. But more solutions can be generated by CGWO than GWO around the global optimal one, as observed from Fig. 8(a3) and (b3). This proves that CLS makes CGWO's exploration of the current optimal solution nearby space more comprehensively, which strengthens the convergence of the algorithm and achieves a better balance between exploration and exploitation for the search.

The above two observations also prove the effectiveness of CLS on GWO.

TABLE 15. Performance comparison among GWO, CGWOs, and MCGWO in terms of solution accuracy on IEEE CEC2017 with $D = 100$ (Continued 2).

	CGWO10		CGWO11		CGWO12		MCGWO	
	mean ± std		mean ± std		mean ± std		mean ± std	
F1	6.60E+09 ± 3.89E+09	+	4.42E+10 ± 8.04E+09	-	9.19E+09 ± 3.76E+09	+	**5.37E+09** ± 3.24E+09	+
F3	8.61E+04 ± 1.72E+04	+	1.98E+05 ± 2.43E+04	=	8.72E+04 ± 2.00E+04	+	9.52E+04 ± 1.76E+04	+
F4	1.33E+03 ± 3.49E+02	+	5.74E+03 ± 1.51E+03	-	1.56E+03 ± 4.64E+02	+	7.45E+02 ± 1.64E+02	+
F5	6.02E+02 ± 1.12E+02	-	6.35E+02 ± 7.15E+01	-	5.96E+02 ± 5.48E+01	-	5.52E+02 ± 9.84E+01	-
F6	3.72E+01 ± 5.26E+00	-	3.69E+01 ± 4.36E+00	-	3.74E+01 ± 4.80E+00	-	2.90E+01 ± 4.41E+00	-
F7	1.12E+03 ± 2.22E+02	=	**1.03E+03** ± 1.01E+02	=	1.10E+03 ± 1.57E+02	=	1.09E+03 ± 2.11E+02	=
F8	6.22E+02 ± 1.15E+02	-	6.70E+02 ± 5.76E+01	-	6.15E+02 ± 5.51E+01	-	5.74E+02 ± 1.16E+02	-
F9	3.28E+04 ± 6.10E+03	-	2.65E+04 ± 1.09E+04	-	2.81E+04 ± 8.09E+03	-	2.91E+04 ± 5.99E+03	-
F10	1.46E+04 ± 2.38E+03	=	1.44E+04 ± 1.37E+03	-	1.48E+04 ± 2.76E+03	=	1.54E+04 ± 4.05E+03	=
F11	5.88E+03 ± 2.09E+03	+	4.23E+04 ± 1.17E+04	-	6.21E+03 ± 2.71E+03	+	5.28E+03 ± 2.23E+03	+
F12	6.27E+08 ± 2.99E+08	+	8.74E+09 ± 4.93E+09	-	7.87E+08 ± 5.26E+08	+	5.55E+08 ± 4.22E+08	+
F13	6.79E+06 ± 7.88E+06	+	5.73E+08 ± 5.56E+08	=	9.29E+06 ± 2.53E+07	+	1.02E+07 ± 2.07E+07	+
F14	5.16E+05 ± 3.15E+05	+	4.39E+06 ± 3.18E+06	=	5.59E+05 ± 2.99E+05	+	5.62E+05 ± 3.11E+05	+
F15	9.90E+05 ± 1.29E+06	+	1.40E+08 ± 2.01E+08	-	1.82E+06 ± 2.73E+06	+	1.42E+06 ± 2.69E+06	+
F16	4.35E+03 ± 1.23E+03	=	4.38E+03 ± 7.30E+02	-	3.88E+03 ± 5.35E+02	=	4.27E+03 ± 1.41E+03	=
F17	3.17E+03 ± 7.85E+02	-	2.90E+03 ± 5.33E+02	-	2.81E+03 ± 6.06E+02	=	2.87E+03 ± 7.37E+02	=
F18	6.99E+05 ± 3.02E+05	+	3.92E+06 ± 3.00E+06	=	8.59E+05 ± 3.77E+05	+	8.39E+05 ± 4.29E+05	+
F19	**4.32E+06** ± 2.49E+06	+	1.02E+08 ± 2.15E+08	=	4.77E+06 ± 3.21E+06	+	5.21E+06 ± 2.46E+06	+
F20	2.61E+03 ± 6.61E+02	=	2.34E+03 ± 6.12E+02	=	**2.25E+03** ± 5.73E+02	=	2.90E+03 ± 9.66E+02	=
F21	8.75E+02 ± 1.38E+02	-	8.86E+02 ± 6.58E+01	-	8.42E+02 ± 8.70E+01	-	7.92E+02 ± 1.33E+02	-
F22	**1.59E+04** ± 5.21E+03	=	1.72E+04 ± 3.41E+03	=	1.69E+04 ± 2.89E+03	=	1.73E+04 ± 6.40E+03	=
F23	1.53E+03 ± 6.36E+02	-	1.67E+03 ± 5.32E+02	-	1.52E+03 ± 6.62E+02	-	**1.09E+03** ± 8.58E+01	-
F24	2.20E+03 ± 7.90E+02	-	2.49E+03 ± 2.13E+02	-	2.03E+03 ± 1.77E+02	-	**1.62E+03** ± 1.93E+02	-
F25	1.76E+03 ± 2.49E+02	+	3.65E+03 ± 6.93E+02	-	2.03E+03 ± 3.06E+02	+	1.42E+03 ± 1.71E+02	+
F26	1.44E+04 ± 2.03E+03	-	1.64E+04 ± 1.65E+03	-	1.43E+04 ± 2.22E+03	-	1.07E+04 ± 1.74E+03	-
F27	1.31E+03 ± 1.62E+02	-	1.79E+03 ± 1.98E+02	-	1.19E+03 ± 1.35E+02	=	1.01E+03 ± 1.13E+02	=
F28	2.06E+03 ± 5.70E+02	+	6.07E+03 ± 1.27E+03	-	2.18E+03 ± 5.58E+02	+	1.27E+03 ± 1.84E+02	+
F29	4.97E+03 ± 7.19E+02	-	5.02E+03 ± 6.30E+02	-	4.80E+03 ± 6.35E+02	-	4.43E+03 ± 7.08E+02	-
F30	8.03E+07 ± 4.53E+07	+	7.40E+08 ± 7.83E+08	-	8.28E+07 ± 4.47E+07	+	5.91E+07 ± 2.87E+07	+
W/T/L	13/5/11		0/10/19		13/7/9		13/7/9	

TABLE 16. Performance comparison between CGWO2 and other competitor algorithms in terms of solution accuracy on IEEE CEC2017 with $D = 30$.

	CGWO2	GWO		PSO		SCA		WFS	
	mean ± std	mean ± std		mean ± std		mean ± std		mean ± std	
F1	**7.84E+07** ± 9.66E+07	9.89E+08 ± 7.80E+08	+	5.11E+10 ± 4.75E+09	+	1.23E+10 ± 1.89E+09	+	7.06E+08 ± 3.32E+08	+
F3	**2.45E+03** ± 1.76E+03	2.99E+04 ± 1.05E+04	+	1.07E+05 ± 1.34E+04	+	3.52E+04 ± 6.48E+03	+	1.50E+04 ± 4.39E+03	+
F4	**1.10E+02** ± 2.12E+01	1.62E+02 ± 4.26E+01	+	1.01E+04 ± 1.71E+03	+	9.43E+02 ± 2.41E+02	+	2.45E+02 ± 5.22E+01	+
F5	**8.58E+01** ± 2.27E+01	8.67E+01 ± 3.32E+01	=	4.27E+02 ± 2.22E+01	+	2.78E+02 ± 2.06E+01	+	1.45E+02 ± 2.98E+01	+
F6	**5.44E+00** ± 2.99E+00	4.43E+00 ± 2.72E+00	-	8.50E+01 ± 4.51E+00	+	4.96E+01 ± 5.78E+00	+	2.55E+01 ± 5.93E+00	+
F7	1.54E+02 ± 4.44E+01	**1.32E+02** ± 3.03E+01	=	1.49E+03 ± 8.33E+01	+	4.24E+02 ± 3.85E+01	+	2.34E+02 ± 3.75E+01	+
F8	**7.24E+01** ± 1.73E+01	7.65E+01 ± 2.34E+01	=	3.92E+02 ± 2.02E+01	+	2.50E+02 ± 1.86E+01	+	1.35E+02 ± 3.10E+01	+
F9	5.10E+02 ± 3.22E+02	**4.95E+02** ± 3.89E+02	=	1.37E+04 ± 1.36E+03	+	4.32E+03 ± 9.18E+02	+	1.77E+03 ± 1.13E+03	+
F10	3.20E+03 ± 9.88E+02	**3.18E+03** ± 9.33E+02	+	7.20E+03 ± 3.30E+02	+	7.22E+03 ± 2.89E+02	+	4.60E+03 ± 6.50E+02	+
F11	**1.60E+02** ± 3.44E+01	4.12E+02 ± 3.93E+02	+	6.15E+03 ± 1.11E+03	+	1.03E+03 ± 4.36E+02	+	3.21E+02 ± 6.94E+01	+
F12	**8.91E+06** ± 6.79E+06	3.28E+07 ± 3.58E+07	+	5.96E+09 ± 1.02E+09	+	1.04E+09 ± 2.39E+08	+	9.80E+07 ± 7.88E+07	+
F13	**1.90E+05** ± 1.14E+05	3.74E+06 ± 1.92E+07	-	2.36E+09 ± 7.88E+08	+	4.17E+08 ± 1.85E+08	+	7.47E+05 ± 8.14E+05	+
F14	**5.37E+03** ± 8.57E+03	1.38E+05 ± 2.83E+05	+	5.13E+05 ± 2.65E+05	+	1.37E+05 ± 7.29E+04	+	7.67E+03 ± 7.82E+03	+
F15	**3.49E+04** ± 2.40E+04	1.99E+05 ± 4.71E+05	+	2.06E+08 ± 9.44E+07	+	1.20E+07 ± 9.82E+06	+	1.55E+05 ± 1.58E+05	+
F16	**6.72E+02** ± 2.14E+02	7.02E+02 ± 2.69E+02	=	2.92E+03 ± 2.26E+02	+	2.00E+03 ± 2.17E+02	+	9.72E+02 ± 2.50E+02	+
F17	**2.23E+02** ± 1.16E+02	2.53E+02 ± 1.17E+02	=	1.34E+03 ± 1.66E+02	+	6.99E+02 ± 1.65E+02	+	3.22E+02 ± 1.07E+02	+
F18	**9.56E+04** ± 6.60E+04	5.46E+05 ± 7.05E+05	+	9.25E+06 ± 3.91E+06	+	3.33E+06 ± 1.54E+06	+	2.16E+05 ± 1.48E+05	+
F19	**1.08E+05** ± 1.43E+05	6.00E+05 ± 6.49E+05	+	3.38E+08 ± 1.27E+08	+	2.47E+07 ± 1.33E+07	+	1.10E+06 ± 1.13E+06	+
F20	**2.98E+02** ± 1.23E+02	3.47E+02 ± 1.27E+02	=	8.26E+02 ± 7.56E+01	+	6.32E+02 ± 1.28E+02	+	3.92E+02 ± 1.08E+02	+
F21	**2.76E+02** ± 1.97E+01	2.76E+02 ± 2.02E+01	=	5.87E+02 ± 2.92E+01	+	4.56E+02 ± 1.68E+01	+	3.33E+02 ± 2.85E+01	+
F22	1.69E+03 ± 1.75E+03	1.45E+03 ± 1.45E+03	=	5.57E+03 ± 5.42E+02	+	5.88E+03 ± 2.51E+03	+	**3.03E+02** ± 8.44E+01	=
F23	**4.34E+02** ± 3.31E+01	4.38E+02 ± 2.75E+01	=	9.43E+02 ± 4.12E+01	+	6.84E+02 ± 2.39E+01	+	5.31E+02 ± 3.99E+01	+
F24	5.19E+02 ± 4.22E+01	**5.16E+02** ± 5.09E+01	=	1.05E+03 ± 5.57E+01	+	7.64E+02 ± 2.41E+01	+	5.79E+02 ± 3.66E+01	+
F25	**4.18E+02** ± 1.83E+01	4.52E+02 ± 2.27E+01	+	3.83E+03 ± 6.90E+02	+	7.05E+02 ± 7.41E+01	+	5.23E+02 ± 3.62E+01	+
F26	**1.78E+03** ± 4.67E+02	1.82E+03 ± 3.45E+02	=	7.02E+03 ± 4.89E+02	+	4.33E+03 ± 3.17E+02	+	2.54E+03 ± 7.02E+02	+
F27	**5.28E+02** ± 1.47E+01	5.38E+02 ± 1.81E+01	+	1.10E+03 ± 8.80E+01	+	7.05E+02 ± 4.12E+01	+	6.19E+02 ± 3.09E+01	+
F28	**4.73E+02** ± 3.30E+01	5.59E+02 ± 5.61E+01	+	3.49E+03 ± 4.16E+02	+	1.02E+03 ± 1.15E+02	+	6.33E+02 ± 7.87E+01	+
F29	**7.80E+02** ± 1.17E+02	7.82E+02 ± 1.53E+02	=	2.62E+03 ± 1.92E+02	+	1.75E+03 ± 2.52E+02	+	1.01E+03 ± 1.44E+02	+
F30	**1.55E+06** ± 1.16E+06	5.65E+06 ± 5.05E+06	+	2.86E+08 ± 7.83E+07	+	6.67E+07 ± 2.42E+07	+	6.90E+06 ± 5.63E+06	+
W/T/L	–	14/13/2		29/0/0		29/0/0		28/1/0	

F. COMPUTATIONAL COMPLEXITY

The above sections verify the performance of CGWO on the benchmark functions. In this section, we analysis the time complexity of CGWOs and MCGWO:

(1) Population initialization requires time complexity $O(N)$ and N is the population size.

(2) The time complexity of population boundary control requires $O(N)$.

TABLE 17. Performance comparison between CGWO5 and other competitor algorithms in terms of solution accuracy on IEEE CEC2017 with $D = 50$.

	CGWO5	GWO		PSO		SCA		WFS	
	mean ± std	mean ± std		mean ± std		mean ± std		mean ± std	
F1	**3.16E+08** ± 4.58E+08	4.99E+09 ± 2.61E+09	+	1.36E+11 ± 1.14E+10	+	3.84E+10 ± 5.66E+09	+	1.67E+09 ± 5.86E+08	+
F3	**1.46E+04** ± 6.04E+03	7.19E+04 ± 1.58E+04	+	2.27E+05 ± 2.40E+04	+	1.01E+05 ± 1.57E+04	+	3.40E+04 ± 5.75E+03	+
F4	**2.07E+02** ± 3.88E+01	4.57E+02 ± 1.83E+02	+	3.34E+04 ± 4.36E+03	+	5.66E+03 ± 1.35E+03	+	4.72E+02 ± 1.19E+02	+
F5	2.15E+02 ± 7.49E+01	**1.90E+02** ± 3.75E+01	-	8.39E+02 ± 2.66E+01	+	5.51E+02 ± 2.92E+01	+	2.75E+02 ± 4.78E+01	+
F6	1.08E+01 ± 3.47E+00	**1.07E+01** ± 3.01E+00	=	1.04E+02 ± 4.43E+00	+	6.87E+01 ± 4.96E+00	+	3.17E+01 ± 8.12E+00	+
F7	3.67E+02 ± 9.13E+01	**3.01E+02** ± 6.11E+01	-	3.40E+03 ± 1.41E+02	+	9.07E+02 ± 6.45E+01	+	4.31E+02 ± 6.07E+01	+
F8	1.99E+02 ± 5.20E+01	**1.99E+02** ± 4.68E+01	=	8.39E+02 ± 3.13E+01	+	5.51E+02 ± 3.04E+01	+	2.73E+02 ± 4.18E+01	+
F9	4.73E+03 ± 1.51E+03	**4.19E+03** ± 1.87E+03	-	4.81E+04 ± 3.76E+03	+	2.08E+04 ± 3.71E+03	+	7.83E+03 ± 4.20E+03	+
F10	5.81E+03 ± 1.12E+03	**5.79E+03** ± 8.45E+02	=	1.34E+04 ± 3.80E+02	+	1.33E+04 ± 4.10E+02	+	8.58E+03 ± 1.04E+03	+
F11	**3.34E+02** ± 6.98E+01	1.79E+03 ± 1.15E+03	+	2.12E+04 ± 3.02E+03	+	4.80E+03 ± 1.21E+03	+	6.74E+02 ± 1.19E+02	+
F12	**6.00E+07** ± 3.69E+07	3.85E+08 ± 5.98E+08	+	4.40E+10 ± 4.61E+09	+	1.15E+10 ± 2.63E+09	+	3.32E+08 ± 1.77E+08	+
F13	**9.35E+05** ± 8.65E+05	9.06E+07 ± 1.31E+08	+	1.50E+10 ± 3.03E+09	+	2.66E+09 ± 8.45E+08	+	3.38E+06 ± 3.23E+06	+
F14	**3.62E+04** ± 4.16E+04	4.25E+05 ± 4.26E+05	+	8.69E+06 ± 3.87E+06	+	1.94E+06 ± 9.17E+05	+	1.11E+05 ± 9.38E+04	+
F15	**7.80E+04** ± 4.88E+04	6.73E+06 ± 1.31E+07	=	3.53E+09 ± 8.30E+08	+	3.40E+08 ± 1.46E+08	+	9.22E+05 ± 1.02E+06	+
F16	**1.27E+03** ± 3.84E+02	1.33E+03 ± 4.16E+02	=	5.56E+03 ± 3.81E+02	+	3.79E+03 ± 3.44E+02	+	1.79E+03 ± 4.81E+02	+
F17	1.05E+03 ± 2.58E+02	**9.15E+02** ± 1.91E+02	-	5.19E+03 ± 5.00E+02	+	2.58E+03 ± 2.44E+02	+	1.29E+03 ± 2.47E+02	+
F18	**3.43E+05** ± 2.34E+05	2.18E+06 ± 2.22E+06	+	4.89E+07 ± 1.76E+07	+	1.42E+07 ± 6.72E+06	+	1.60E+06 ± 9.38E+05	+
F19	**5.85E+05** ± 3.85E+05	1.99E+06 ± 4.52E+06	+	1.49E+09 ± 3.94E+08	+	2.20E+08 ± 1.06E+08	+	1.95E+06 ± 1.40E+06	+
F20	8.08E+02 ± 2.53E+02	**7.31E+02** ± 2.54E+02	=	2.00E+03 ± 1.35E+02	+	1.77E+03 ± 1.89E+02	+	9.67E+02 ± 2.35E+02	+
F21	4.04E+02 ± 5.85E+01	**3.86E+02** ± 3.00E+01	=	1.03E+03 ± 3.82E+01	+	7.62E+02 ± 3.22E+01	+	4.73E+02 ± 4.94E+01	+
F22	5.72E+03 ± 2.26E+03	**5.69E+03** ± 1.53E+03	=	1.40E+04 ± 3.71E+02	+	1.37E+04 ± 3.71E+02	+	8.18E+03 ± 2.08E+03	+
F23	6.60E+02 ± 9.40E+01	**6.27E+02** ± 5.96E+01	-	1.74E+03 ± 6.63E+01	+	1.21E+03 ± 5.62E+01	+	8.06E+02 ± 6.41E+01	+
F24	7.53E+02 ± 9.08E+01	**7.13E+02** ± 1.00E+02	-	1.88E+03 ± 8.47E+01	+	1.26E+03 ± 5.27E+01	+	8.49E+02 ± 7.06E+01	+
F25	**6.22E+02** ± 6.73E+01	8.80E+02 ± 1.88E+02	+	2.10E+04 ± 2.46E+03	+	3.43E+03 ± 5.34E+02	+	9.04E+02 ± 9.02E+01	+
F26	3.79E+03 ± 8.88E+02	**3.30E+03** ± 4.96E+02	-	1.53E+04 ± 6.29E+02	+	9.06E+03 ± 6.19E+02	+	4.74E+03 ± 5.02E+02	+
F27	**7.24E+02** ± 8.00E+01	8.06E+02 ± 1.35E+02	+	2.77E+03 ± 1.98E+02	+	1.67E+03 ± 1.66E+02	+	1.07E+03 ± 1.12E+02	+
F28	**6.22E+02** ± 8.24E+01	1.09E+03 ± 2.84E+02	+	1.01E+04 ± 7.69E+02	+	3.53E+03 ± 4.53E+02	+	1.26E+03 ± 3.10E+02	+
F29	1.33E+03 ± 3.62E+02	**1.27E+03** ± 2.17E+02	=	8.70E+03 ± 1.30E+03	+	4.26E+03 ± 5.92E+02	+	1.98E+03 ± 3.94E+02	+
F30	**2.33E+07** ± 7.35E+06	6.51E+07 ± 2.21E+07	+	2.74E+09 ± 5.64E+08	+	5.83E+08 ± 1.84E+08	+	2.00E+08 ± 3.75E+07	+
W/T/L	–	13/9/7		29/0/0		29/0/0		29/0/0	

TABLE 18. Performance comparison between CGWO5 and other competitor algorithms in terms of solution accuracy on IEEE CEC2017 with $D = 100$.

	CGWO5	GWO		PSO		SCA		WFS	
	mean ± std	mean ± std		mean ± std		mean ± std		mean ± std	
F1	5.43E+09 ± 2.94E+09	3.05E+10 ± 7.66E+09	+	3.99E+11 ± 2.17E+10	+	1.53E+11 ± 9.58E+09	+	**3.90E+09** ± 1.44E+09	-
F3	**8.82E+04** ± 1.51E+04	2.06E+05 ± 1.97E+04	+	5.93E+05 ± 2.97E+04	+	2.86E+05 ± 2.00E+04	+	1.12E+05 ± 1.21E+04	+
F4	**7.37E+02** ± 1.68E+02	2.59E+03 ± 8.32E+02	+	1.21E+05 ± 1.09E+04	+	2.59E+04 ± 3.42E+03	+	8.70E+02 ± 1.06E+02	+
F5	5.77E+02 ± 1.51E+02	**5.42E+02** ± 6.07E+01	=	2.00E+03 ± 4.48E+01	+	1.35E+03 ± 5.76E+01	+	5.66E+02 ± 8.52E+01	=
F6	2.78E+01 ± 5.13E+00	**2.76E+01** ± 4.05E+00	=	1.26E+02 ± 2.69E+00	+	8.89E+01 ± 3.95E+00	+	2.77E+01 ± 7.59E+00	=
F7	1.10E+03 ± 2.23E+02	**1.04E+03** ± 9.37E+01	=	9.10E+03 ± 3.46E+02	+	2.69E+03 ± 1.20E+02	+	1.19E+03 ± 1.39E+02	+
F8	5.98E+02 ± 1.66E+02	**5.69E+02** ± 6.52E+01	=	2.09E+03 ± 5.18E+01	+	1.40E+03 ± 5.56E+01	+	5.92E+02 ± 9.03E+01	=
F9	2.97E+04 ± 6.73E+03	2.10E+04 ± 1.03E+04	-	1.36E+05 ± 6.60E+03	+	6.67E+04 ± 6.91E+03	+	**1.14E+04** ± 4.46E+03	-
F10	1.49E+04 ± 3.51E+03	**1.42E+04** ± 2.72E+03	=	3.04E+04 ± 4.62E+02	+	3.02E+04 ± 4.87E+02	+	1.93E+04 ± 1.66E+03	+
F11	**4.14E+03** ± 1.44E+03	3.78E+04 ± 9.65E+03	+	2.27E+05 ± 1.96E+04	+	6.98E+04 ± 1.10E+04	+	4.57E+03 ± 7.80E+02	=
F12	**4.61E+08** ± 2.66E+08	4.68E+09 ± 3.09E+09	+	1.72E+11 ± 1.35E+10	+	5.33E+10 ± 7.14E+09	+	1.30E+09 ± 2.86E+08	+
F13	9.42E+06 ± 1.88E+07	5.17E+08 ± 5.07E+08	+	3.49E+10 ± 4.02E+09	+	8.22E+09 ± 1.30E+09	+	**7.82E+06** ± 4.67E+06	=
F14	**5.21E+05** ± 2.60E+05	4.31E+06 ± 3.53E+06	+	7.20E+07 ± 1.39E+07	+	1.80E+07 ± 6.98E+06	+	2.04E+06 ± 8.20E+05	+
F15	**1.38E+06** ± 1.82E+06	6.86E+07 ± 1.24E+08	+	1.31E+10 ± 1.92E+09	+	2.57E+09 ± 6.87E+08	+	1.91E+06 ± 1.67E+06	+
F16	4.00E+03 ± 1.05E+03	**3.85E+03** ± 5.26E+02	=	1.68E+04 ± 1.03E+03	+	1.08E+04 ± 6.02E+02	+	4.49E+03 ± 5.93E+02	+
F17	2.87E+03 ± 6.46E+02	**2.77E+03** ± 5.24E+02	=	3.59E+05 ± 2.11E+05	+	9.13E+03 ± 1.20E+03	+	2.91E+03 ± 4.91E+02	=
F18	**9.19E+05** ± 5.09E+05	3.91E+06 ± 2.60E+06	+	1.32E+08 ± 2.65E+07	+	3.34E+07 ± 1.15E+07	+	2.47E+06 ± 1.22E+06	+
F19	**4.66E+06** ± 2.13E+06	7.03E+07 ± 1.05E+08	+	1.35E+10 ± 1.51E+09	+	2.11E+09 ± 6.41E+08	+	6.48E+06 ± 3.87E+06	+
F20	2.92E+03 ± 7.26E+02	**2.40E+03** ± 6.84E+02	-	5.42E+03 ± 3.11E+02	+	5.12E+03 ± 2.62E+02	+	2.90E+03 ± 4.79E+02	=
F21	8.24E+02 ± 1.84E+02	**7.63E+02** ± 5.94E+01	=	2.45E+03 ± 5.93E+01	+	1.78E+03 ± 6.64E+01	+	8.65E+02 ± 9.72E+01	+
F22	1.76E+04 ± 4.46E+03	**1.63E+04** ± 2.99E+03	=	3.15E+04 ± 4.66E+02	+	3.13E+04 ± 6.22E+02	+	2.07E+04 ± 1.34E+03	+
F23	**1.10E+03** ± 1.01E+02	1.17E+03 ± 3.88E+02	+	3.59E+03 ± 1.48E+02	+	2.47E+03 ± 8.35E+01	+	1.50E+03 ± 1.44E+02	+
F24	**1.62E+03** ± 2.09E+02	1.64E+03 ± 8.41E+02	=	6.63E+03 ± 3.46E+02	+	3.90E+03 ± 1.74E+02	+	2.00E+03 ± 1.71E+02	+
F25	**1.41E+03** ± 1.60E+02	2.70E+03 ± 5.21E+02	+	6.30E+04 ± 4.05E+03	+	1.15E+04 ± 1.53E+03	+	1.67E+03 ± 1.09E+02	+
F26	1.03E+04 ± 1.71E+03	**1.02E+04** ± 1.04E+03	=	5.01E+04 ± 2.65E+03	+	2.92E+04 ± 1.47E+03	+	1.10E+04 ± 1.10E+03	+
F27	**1.01E+03** ± 9.13E+01	1.14E+03 ± 1.04E+02	+	7.10E+03 ± 4.05E+02	+	4.02E+03 ± 3.77E+02	+	1.42E+03 ± 1.31E+02	+
F28	**1.22E+03** ± 2.30E+02	3.86E+03 ± 1.10E+03	+	4.07E+04 ± 1.74E+03	+	1.55E+04 ± 1.59E+03	+	2.15E+03 ± 3.57E+02	+
F29	**4.41E+03** ± 6.15E+02	4.50E+03 ± 5.95E+02	=	1.08E+05 ± 4.20E+04	+	1.26E+04 ± 1.25E+03	+	5.32E+03 ± 6.49E+02	+
F30	**5.32E+07** ± 2.90E+07	3.40E+08 ± 2.80E+08	+	2.21E+10 ± 3.00E+09	+	5.49E+09 ± 1.00E+09	+	2.99E+08 ± 7.82E+07	+
W/T/L	–	15/12/2		29/0/0		29/0/0		20/7/2	

(3) Population evaluation and selection of the $\vec{x_\alpha}$, $\vec{x_\beta}$ and $\vec{x_\delta}$ require time complexity $O(N)$.

(4) The time complexity of chaotic map selection in MCGWO is $O(J)$, where J is the number of chaotic maps.

(5) The time complexity of CLS boundary control is $O(1)$.

(6) The GWO requires time complexity $O(N)$ to generate offspring.

When the algorithm terminates after T iterations, the total time complexity is shown as follows:

$$\begin{aligned} &O(N) + T[O(N) + O(N) + O(J) + O(1) + O(N)] \\ &= O(N) + 3T \cdot O(N) + T \cdot O(J) + T \cdot O(1) \\ &= (3T+1) \cdot O(N) + T \cdot O(J) + T \cdot O(1) \end{aligned} \tag{25}$$

In this study, J is less than N; therefore, the total time complexity of both CGWOs and MCGWO is $O(N)$.

VI. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we propose a number of chaotic grey wolf optimization algorithms (CGWOs). The main feature of the improvement in the algorithm is the incorporation of chaotic local search. We have adopted two approaches to perform CLS: One is to directly use a single chaotic map induced CLS, and the other is to implement CLS by selectively using multiple chaotic maps based on the accumulated success information. Extensive experiments are conducted based on IEEE CEC2017, and the results show that the use of CLS can speed up the global convergence of GWO and also give it the ability to jump out of a local optimum. Additionally, the time complexity calculated for CGWOs is almost the same as that for GWO.

This comprehensive comparative study not only proposes effective CGWOs which have been demonstrated their superiority over some other state-of-the-art meta-heuristic algorithms, but also gives some valuable findings: 1) single chaotic map incorporation scheme can perform better than the multiple chaotic maps incorporation scheme once a suitable map is chosen; and 2) in CGWOs, PWLCM is the most promising candidate for lower dimensional problems while Gaussian map performs better for higher ones.

This study also opens the door to the following researches: 1) More sophisticated incorporation method using multiple chaotic maps should be designed by not only the interaction with the tackled problem (as done in this study), but also the inherent property of the chaotic maps (e.g., its maximum Lyapunov exponent), and 2) The effectiveness of CGWOs should also be verified on real-world application problems.

APPENDIX

DETAILED EXPERIMENTAL RESULTS

Tables 7, 8, 9 give the performance comparison results among GWO, CGWOs, and MCGWO in terms of solution accuracy on IEEE CEC2017 benchmark functions with $D = 30$. Tables 10, 11, 12 give the performance comparison results among GWO, CGWOs, and MCGWO in terms of solution accuracy on functions with $D = 50$. Tables 13, 14, 15 give the performance comparison results among GWO, CGWOs, and MCGWO in terms of solution accuracy on functions with $D = 100$. Tables 16, 17, 18 give the detailed comparative results among CGWO and its competitors on functions with dimensions of 30, 50, and 100, respectively.

REFERENCES

[1] J. D. Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P. N. Suganthan, C. A. C. Coello, and F. Herrera, "Bio-inspired computation: Where we stand and what's next," *Swarm Evol. Comput.*, vol. 48, pp. 220–250, Aug. 2019.

[2] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Comput. Ind. Eng.*, vol. 137, Nov. 2019, Art. no. 106040.

[3] X.-S. Yang, "Nature-inspired optimization algorithms: Challenges and open problems," *J. Comput. Sci.*, vol. 46, Oct. 2020, Art. no. 101104.

[4] S. Y. Yuen and C. K. Chow, "A genetic algorithm that adaptively mutates and never revisits," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 454–472, Apr. 2009.

[5] S. Gao, S. Song, J. Cheng, Y. Todo, and M. Zhou, "Incorporation of solvent effect into multi-objective evolutionary algorithm for improved protein structure prediction," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 4, pp. 1365–1378, Jul. 2018.

[6] J. Sun, S. Gao, H. Dai, J. Cheng, M. Zhou, and J. Wang, "Bi-objective elite differential evolution algorithm for multivalued logic networks," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 233–246, Jan. 2020.

[7] Z. Xu, S. Gao, H. Yang, and Z. Lei, "SCJADE: Yet another state-of-the-art differential evolution algorithm," *IEEJ Trans. Electr. Electron. Eng.*, vol. 16, no. 4, pp. 644–646, Apr. 2021.

[8] S. Gao, K. Wang, S. Tao, T. Jin, H. Dai, and J. Cheng, "A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models," *Energy Convers. Manage.*, vol. 230, Feb. 2021, Art. no. 113784.

[9] H. Yang, S. Gao, R.-L. Wang, and Y. Todo, "A ladder spherical evolution search algorithm," *IEICE Trans. Inf. Syst.*, vol. E104.D, no. 3, pp. 461–464, 2021.

[10] S. Gao, R.-L. Wang, M. Ishii, and Z. Tang, "An artificial immune system with feedback mechanisms for effective handling of population size," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. E93-A, no. 2, pp. 532–541, 2010.

[11] S. Song, J. Ji, X. Chen, S. Gao, Z. Tang, and Y. Todo, "Adoption of an improved PSO to explore a compound multi-objective energy function in protein structure prediction," *Appl. Soft Comput.*, vol. 72, pp. 539–551, Nov. 2018.

[12] S. Gao, W. Wang, H. Dai, F. Li, and Z. Tang, "Improved clonal selection algorithm combined with ant colony optimization," *IEICE Trans. Inf. Syst.*, vol. E91-D, no. 6, pp. 1813–1823, Jun. 2008.

[13] B. Morales-Castañeda, D. Zaldívar, E. Cuevas, O. Maciel-Castillo, I. Aranguren, and F. Fausto, "An improved simulated annealing algorithm based on ancient metallurgy techniques," *Appl. Soft Comput.*, vol. 84, Nov. 2019, Art. no. 105761.

[14] Z. Lei, S. Gao, S. Gupta, J. Cheng, and G. Yang, "An aggregative learning gravitational search algorithm with self-adaptive gravitational constants," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113396.

[15] A. Manju and M. J. Nigam, "Applications of quantum inspired computational intelligence: A survey," *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 79–156, Jun. 2014.

[16] Aorigele, Z. Tang, Y. Todo, and S. Gao, "A hybrid discrete imperialist competition algorithm for gene selection for microarray data," *Current Proteomics*, vol. 15, no. 2, pp. 99–110, Mar. 2018.

[17] Y. Yu, S. Gao, Y. Wang, J. Cheng, and Y. Todo, "ASBSO: An improved brain storm optimization with flexible search length and memory-based selection," *IEEE Access*, vol. 6, pp. 36977–36994, 2018.

[18] C. Soza, R. L. Becerra, M. C. Riff, and C. A. C. Coello, "Solving timetabling problems using a cultural algorithm," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 337–344, Jan. 2011.

[19] Y. Yu, S. Gao, Y. Wang, Z. Lei, J. Cheng, and Y. Todo, "A multiple diversity-driven brain storm optimization algorithm with adaptive parameters," *IEEE Access*, vol. 7, pp. 126871–126888, 2019.

[20] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, no. 6791, pp. 39–42, Jul. 2000.

[21] Y. Wang, Y. Yu, S. Cao, X. Zhang, and S. Gao, "A review of applications of artificial intelligent algorithms in wind farms," *Artif. Intell. Rev.*, vol. 53, no. 5, pp. 3447–3500, Jun. 2020.

[22] A. E. Eiben and J. Smith, "From evolutionary computation to the evolution of things," *Nature*, vol. 521, no. 7553, pp. 476–482, May 2015.

[23] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Inf. Sci.*, vol. 237, pp. 82–117, Jul. 2013.

[24] Q. Li, S.-Y. Liu, and X.-S. Yang, "Influence of initialization on the performance of metaheuristic optimizers," *Appl. Soft Comput.*, vol. 91, Jun. 2020, Art. no. 106193.

[25] G. Song, Z. Wang, F. Han, S. Ding, and M. A. Iqbal, "Music auto-tagging using deep recurrent neural networks," *Neurocomputing*, vol. 292, pp. 104–110, May 2018.

[26] B. Khan, F. Han, Z. Wang, and R. J. Masood, "Bio-inspired approach to invariant recognition and classification of fabric weave patterns and yarn color," *Assem. Autom.*, vol. 36, no. 2, pp. 152–158, Apr. 2016.

[27] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 601–614, Feb. 2019.

[28] J. Cheng, G. Yuan, M. Zhou, S. Gao, C. Liu, H. Duan, and Q. Zeng, "Accessibility analysis and modeling for IoV in an urban scene," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4246–4256, Apr. 2020.

[29] K. Sörensen, "Metaheuristics—The metaphor exposed," *Int. Trans. Oper. Res.*, vol. 22, no. 1, pp. 3–18, Jan. 2015.

[30] M. Gendreau and J.-Y. Potvin, "Metaheuristics in combinatorial optimization," *Ann. Oper. Res.*, vol. 140, no. 1, pp. 189–213, 2005.

[31] G. Zäpfel, R. Braune, and M. Bögl, *Metaheuristic Search Concepts: A Tutorial With Applications to Production and Logistics*. Berlin, Germany: Springer, 2010.

[32] W. Shan, Z. Qiao, A. A. Heidari, H. Chen, H. Turabieh, and Y. Teng, "Double adaptive weights for stabilization of moth flame optimizer: Balance analysis, engineering cases, and medical diagnosis," *Knowl.-Based Syst.*, vol. 214, Feb. 2021, Art. no. 106728.

[33] G. Karafotias, M. Hoogendoorn, and A. E. Eiben, "Parameter control in evolutionary algorithms: Trends and challenges," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 167–187, Apr. 2015.

[34] A. E. Eiben and S. K. Smit, "Parameter tuning for configuring and analyzing evolutionary algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 19–31, Mar. 2011.

[35] A. Aleti and I. Moser, "A systematic literature review of adaptive parameter control methods for evolutionary algorithms," *ACM Comput. Surveys*, vol. 49, no. 3, pp. 1–35, Dec. 2016.

[36] Y. Yu, S. Gao, Y. Wang, and Y. Todo, "Global optimum-based search differential evolution," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 2, pp. 379–394, Mar. 2019.

[37] A. P. Piotrowski, "Review of differential evolution population size," *Swarm Evol. Comput.*, vol. 32, pp. 1–24, Feb. 2017.

[38] E.-N. Dragoi and V. Dafinescu, "Parameter control and hybridization techniques in differential evolution: A survey," *Artif. Intell. Rev.*, vol. 45, no. 4, pp. 447–470, Apr. 2016.

[39] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 71–78.

[40] J. Tu, H. Chen, J. Liu, A. A. Heidari, X. Zhang, M. Wang, R. Ruby, and Q.-V. Pham, "Evolutionary biogeography-based whale optimization methods with communication structure: Towards measuring the balance," *Knowl.-Based Syst.*, vol. 212, Jan. 2021, Art. no. 106642.

[41] J. L. Payne, M. Giacobini, and J. H. Moore, "Complex and dynamic population structures: Synthesis, open questions, and future directions," *Soft Comput.*, vol. 17, no. 7, pp. 1109–1120, Jul. 2013.

[42] N. Lynn, M. Z. Ali, and P. N. Suganthan, "Population topologies for particle swarm optimization and differential evolution," *Swarm Evol. Comput.*, vol. 39, pp. 24–35, Apr. 2018.

[43] Y. Wang, S. Gao, M. Zhou, and Y. Yu, "A multi-layered gravitational search algorithm for function optimization and real-world problems," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 1, pp. 94–109, Jan. 2021.

[44] Y. Wang, Y. Yu, S. Gao, H. Pan, and G. Yang, "A hierarchical gravitational search algorithm with an effective gravitational constant," *Swarm Evol. Comput.*, vol. 46, pp. 118–139, May 2019.

[45] J. Ji, S. Song, C. Tang, S. Gao, Z. Tang, and Y. Todo, "An artificial bee colony algorithm search guided by scale-free networks," *Inf. Sci.*, vol. 473, pp. 142–165, Jan. 2019.

[46] Y. Wang, S. Gao, Y. Yu, Z. Cai, and Z. Wang, "A gravitational search algorithm with hierarchy and distributed framework," *Knowl.-Based Syst.*, vol. 218, Apr. 2021, Art. no. 106877.

[47] B. Morales-Castañeda, D. Zaldívar, E. Cuevas, F. Fausto, and A. Rodríguez, "A better balance in metaheuristic algorithms: Does it exist?" *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100671.

[48] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Comput. Surveys*, vol. 45, no. 3, pp. 1–33, Jun. 2013.

[49] Y. Wang, S. Gao, Y. Yu, and Z. Xu, "The discovery of population interaction with a power law distribution in brain storm optimization," *Memetic Comput.*, vol. 11, no. 1, pp. 65–87, Mar. 2019.

[50] S. Gao, Y. Wang, J. Wang, and J. Cheng, "Understanding differential evolution: A Poisson law derived from population interaction network," *J. Comput. Sci.*, vol. 21, pp. 140–149, Jul. 2017.

[51] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Ensemble strategies for population-based optimization algorithms—A survey," *Swarm Evol. Comput.*, vol. 44, pp. 695–711, Feb. 2019.

[52] F. Neri and C. Cotta, "Memetic algorithms and memetic computing optimization: A literature review," *Swarm Evol. Comput.*, vol. 2, pp. 1–14, Feb. 2012.

[53] Z. Xu, Y. Wang, S. Li, Y. Liu, Y. Todo, and S. Gao, "Immune algorithm combined with estimation of distribution for traveling salesman problem," *IEEJ Trans. Electr. Electron. Eng.*, vol. 11, no. S1, pp. S142–S154, Jun. 2016.

[54] S. Wang, G. Liu, and S. Gao, "A hybrid discrete imperialist competition algorithm for fuzzy job-shop scheduling problems," *IEEE Access*, vol. 4, pp. 9320–9331, 2016.

[55] S. Gao, Y. Wang, J. Cheng, Y. Inazumi, and Z. Tang, "Ant colony optimization with clustering for solving the dynamic location routing problem," *Appl. Math. Comput.*, vol. 285, pp. 149–173, Jul. 2016.

[56] S. Gupta and K. Deep, "Random walk grey wolf optimizer for constrained engineering optimization problems," *Comput. Intell.*, vol. 34, no. 4, pp. 1025–1045, Nov. 2018.

[57] H. Haklı and H. Uğuz, "A novel particle swarm optimization algorithm with Levy flight," *Appl. Soft Comput.*, vol. 23, pp. 333–345, Oct. 2014.

[58] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[59] A. Kaveh, "Chaos embedded metaheuristic algorithms," in *Advances in Metaheuristic Algorithms for Optimal Design of Structures*. Cham, Switzerland: Springer, 2017, pp. 375–398.

[60] A. Saxena, "A comprehensive study of chaos embedded bridging mechanisms and crossover operators for grasshopper optimisation algorithm," *Expert Syst. Appl.*, vol. 132, pp. 166–188, Oct. 2019.

[61] A. Saxena, S. Shekhawat, A. Sharma, H. Sharma, and R. Kumar, "Chaotic step length artificial bee colony algorithms for protein structure prediction," *J. Interdiscipl. Math.*, vol. 23, no. 2, pp. 617–629, Feb. 2020.

[62] F. Han and Q.-S. Lu, "An improved chaos optimization algorithm and its application in the economic load dispatch problem," *Int. J. Comput. Math.*, vol. 85, no. 6, pp. 969–982, Jun. 2008.

[63] F. Han, Q. Lu, M. Wiercigroch, and Q. Ji, "Chaotic burst synchronization in heterogeneous small-world neuronal network with noise," *Int. J. Non-Linear Mech.*, vol. 44, no. 3, pp. 298–303, Apr. 2009.

[64] D. Zhao, L. Liu, F. Yu, A. A. Heidari, M. Wang, G. Liang, K. Muhammad, and H. Chen, "Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D kapur entropy," *Knowl.-Based Syst.*, vol. 216, Mar. 2021, Art. no. 106510.

[65] R. Caponetto, L. Fortuna, S. Fazzino, and M. G. Xibilia, "Chaotic sequences to improve the performance of evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 3, pp. 289–304, Jun. 2003.

[66] J. Chuanwen and E. Bompard, "A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimisation," *Math. Comput. Simul.*, vol. 68, no. 1, pp. 57–65, Feb. 2005.

[67] J. Chuanwen and E. Bompard, "A self-adaptive chaotic particle swarm algorithm for short term hydroelectric system scheduling in deregulated environment," *Energy Convers. Manage.*, vol. 46, no. 17, pp. 2689–2696, Oct. 2005.

[68] A. H. Gandomi, G. J. Yun, X.-S. Yang, and S. Talatahari, "Chaos-enhanced accelerated particle swarm optimization," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 2, pp. 327–340, Feb. 2013.

[69] X. Zhao, X. Zhang, Z. Cai, X. Tian, X. Wang, Y. Huang, H. Chen, and L. Hu, "Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients," *Comput. Biol. Chem.*, vol. 78, pp. 481–490, Feb. 2019.

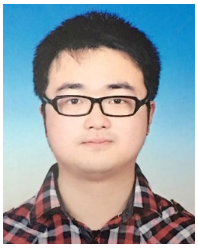
[70] T. Xiang, X. Liao, and K.-W. Wong, "An improved particle swarm optimization algorithm combined with piecewise linear chaotic map," *Appl. Math. Comput.*, vol. 190, no. 2, pp. 1637–1645, Jul. 2007.

[71] Y. Wang, S. Gao, Y. Yu, Z. Wang, J. Cheng, and T. Yuki, "A gravitational search algorithm with chaotic neural oscillators," *IEEE Access*, vol. 8, pp. 25938–25948, 2020.

[72] J. Ji, S. Gao, S. Wang, Y. Tang, H. Yu, and Y. Todo, "Self-adaptive gravitational search algorithm with a modified chaotic local search," *IEEE Access*, vol. 5, pp. 17881–17895, 2017.

[73] S. Gao, C. Vairappan, Y. Wang, Q. Cao, and Z. Tang, "Gravitational search algorithm combined with chaos for unconstrained numerical optimization," *Appl. Math. Comput.*, vol. 231, pp. 48–62, Mar. 2014.

[74] B. Alatas, "Chaotic bee colony algorithms for global numerical optimization," *Expert Syst. Appl.*, vol. 37, no. 8, pp. 5682–5687, Aug. 2010.
[75] Y. Yu, S. Gao, S. Cheng, Y. Wang, S. Song, and F. Yuan, "CBSO: A memetic brain storm optimization with chaotic local search," *Memetic Comput.*, vol. 10, no. 4, pp. 353–367, Dec. 2018.
[76] D. Jia, G. Zheng, and M. K. Khan, "An effective memetic differential evolution algorithm based on chaotic local search," *Inf. Sci.*, vol. 181, no. 15, pp. 3175–3187, Aug. 2011.
[77] Y. Liu, Y. Shi, H. Chen, A. A. Heidari, W. Gui, M. Wang, H. Chen, and C. Li, "Chaos-assisted multi-population salp swarm algorithms: Framework and case studies," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114369.
[78] F. Zhao, Y. Liu, Z. Shao, X. Jiang, C. Zhang, and J. Wang, "A chaotic local search based bacterial foraging algorithm and its application to a permutation flow-shop scheduling problem," *Int. J. Comput. Integr. Manuf.*, vol. 29, no. 9, pp. 962–981, Sep. 2016.
[79] L. dos Santos Coelho and V. C. Mariani, "Self-adaptive differential evolution using chaotic local search for solving power economic dispatch with nonsmooth fuel cost function," in *Advances in Differential Evolution*. Berlin, Germany: Springer, 2008, pp. 275–286.
[80] S. Gao, J. Zhang, Z. Tnag, and Q. Cao, "A chaotic dynamic local search method for learning multiple-valued logic networks," *J. Multiple-Valued Logic Soft Comput.*, vol. 15, no. 5, pp. 433–449, 2009.
[81] I. Al-Jadir, K. W. Wong, C. C. Fung, and H. Xie, "Differential evolution memetic document clustering using chaotic logistic local search," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2017, pp. 213–221.
[82] Z. Zhang, Z. Tang, S. Gao, and G. Yang, "An algorithm of chaotic dynamic adaptive local search method for Elman neural network," *Int. J. Innov. Comput., Inf. Control*, vol. 7, no. 2, pp. 647–656, 2011.
[83] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, "Chaotic local search-based differential evolution algorithms for optimization," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 6, pp. 3954–3967, Jun. 2021, doi: 10.1109/tsmc.2019.2956121.
[84] Z. Song, S. Gao, Y. Yu, J. Sun, and Y. Todo, "Multiple chaos embedded gravitational search algorithm," *IEICE Trans. Inf. Syst.*, vol. E100.D, no. 4, pp. 888–900, 2017.
[85] S. Wang, S. Song, Y. Yu, Z. Xu, H. Yachi, and S. Gao, "Multiple chaotic cuckoo search algorithm," in *Proc. Int. Conf. Swarm Intell.* Cham, Switzerland: Springer, 2017, pp. 531–542.
[86] J. Yi, X. Li, C.-H. Chu, and L. Gao, "Parallel chaotic local search enhanced harmony search algorithm for engineering design optimization," *J. Intell. Manuf.*, vol. 30, no. 1, pp. 405–428, Jan. 2019.
[87] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[88] X. Sun, Z. Jin, Y. Cai, Z. Yang, and L. Chen, "Grey wolf optimization algorithm based state feedback control for a bearingless permanent magnet synchronous machine," *IEEE Trans. Power Electron.*, vol. 35, no. 12, pp. 13631–13640, Dec. 2020.
[89] A. Al Shorman, H. Faris, and I. Aljarah, "Unsupervised intelligent system based on one class support vector machine and grey wolf optimization for IoT botnet detection," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 7, pp. 2809–2825, Jul. 2020.
[90] E. N. Kalemci, S. B. I. kizler, T. Dede, and Z. Angın, "Design of reinforced concrete cantilever retaining wall using grey wolf optimization algorithm," in *Structures*, vol. 23. Elsevier, 2020, pp. 245–253.
[91] J. Hu, H. Chen, A. A. Heidari, M. Wang, X. Zhang, Y. Chen, and Z. Pan, "Orthogonal learning covariance matrix for defects of grey wolf optimizer: Insights, balance, diversity, and feature selection," *Knowl.-Based Syst.*, vol. 213, Feb. 2021, Art. no. 106684.
[92] C. L. C. Villalón, T. Stützle, and M. Dorigo, "Grey wolf, firefly and bat algorithms: Three widespread algorithms that do not contain any novelty," in *Proc. Int. Conf. Swarm Intell.* Cham, Switzerland: Springer, 2020, pp. 121–133.
[93] S. Dhargupta, M. Ghosh, S. Mirjalili, and R. Sarkar, "Selective opposition based grey wolf optimization," *Expert Syst. Appl.*, vol. 151, Aug. 2020, Art. no. 113389.
[94] H. Dong and Z. Dong, "Surrogate-assisted grey wolf optimization for high-dimensional, computationally expensive black-box problems," *Swarm Evol. Comput.*, vol. 57, Sep. 2020, Art. no. 100713.
[95] J. Carrasco, S. García, M. M. Rueda, S. Das, and F. Herrera, "Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review," *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100665.
[96] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.
[97] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.
[98] N. Covic and B. Lacevic, "Wingsuit flying search—A novel global optimization algorithm," *IEEE Access*, vol. 8, pp. 53883–53900, 2020.



ZHE XU received the B.S. degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2009, and the M.S. and Ph.D. degrees from the Faculty of Engineering, University of Toyama, Toyama, Japan, in 2012 and 2016, respectively. From 2016 to 2017, he held a postdoctoral position with the University of Toyama. From 2017 to 2018, he was a Research Fellow with the University of Toyama. He is currently a Lecturer with the Changzhou Institute of Technology, Changzhou, China. His research interests include intelligence computing, neural networks, swarm intelligent algorithms, and combinational optimization problems.

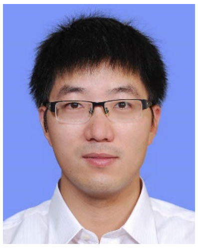
HAICHUAN YANG received the B.E. degree from the China University of Petroleum, Qingdao, China, in 2016. He is currently pursuing the M.S. degree with the University of Toyama, Toyama, Japan. His research interest includes computational intelligence.

JIAYI LI received the B.E. degree from the Beijing University of Civil Engineering and Architecture, Beijing, China, in 2020. He is currently pursuing the M.S. degree with the University of Toyama, Toyama, Japan. His research interests include computational intelligence and neural networks.

XINGYI ZHANG received the Ph.D. degree in pulmonary medicine from Shanghai Medical University, Shanghai, China, in 1995. She is currently a Chief Doctor of Shanghai General Hospital affiliated to Shanghai Jiaotong University, China. Her research interests include application of artificial intelligence technology to diagnosis of pulmonary nodules and pulmonary infectious diseases.

BO LU received the Ph.D. degree in electrical and electronic engineering from the University of Nottingham, Nottingham, U.K., in 2007.

She is currently an Associate Professor with the Faculty of Engineering, Shanghai Normal University Tianhua College, China. Her research interests include computer medical imaging, machine learning, and computer vision applications.



SHANGCE GAO (Senior Member, IEEE) received the B.S. degree from Southeast University, Nanjing, China, in 2005, and the M.S. and Ph.D. degrees from the University of Toyama, Toyama, Japan, in 2008 and 2011, respectively.

He worked as an Associate Professor with Tongji University, from 2011 to 2012, and Donghua University, from 2012 to 2014. He is currently an Associate Professor with the Faculty of Engineering, University of Toyama. His research has led to over 100 publications in top venues, such as IEEE Transactions on Neural Networks and Learning Systems (TNNLS), IEEE Transactions on Cybernetics (CYB), and IEEE Transactions on Systems, Man, and Cybernetics: Systems (TSMCS). His research interests include nature-inspired technologies, mobile computing, machine learning, and neural networks for real-world applications.

Dr. Gao was a recipient of the Best Paper Award from IEEE International Conference on Progress in Informatics and Computing, the Shanghai Rising-Star Scientist Award, the Chen-Guang Scholar of Shanghai Award, the Outstanding Academic Performance Award of IEICE, and the Outstanding Academic Achievement Award of IPSJ. He serves as a Secretary for Hokuriku Section and IEICE. He has served on the program committees for several international professional conferences. He also serves as an Associate Editor for many international journals, such as IEEE/CAA Journal of Automatica Sinica, IEEE Access, *International Journal of Bio-Inspired Computation*, *Mathematical Problems in Engineering*, and *International Journal of Automation and Control*.