IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 19, 2017, accepted July 4, 2017, date of current version May 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2724073

Dynamic Group Optimization Algorithm With Embedded Chaos

RUI TANG[1,2], SIMON FONG[2], RAYMOND K. WONG[3], AND KELVIN K. L. WONG[1,4]

[1]Faculty of Mathematics and Computer Science, Quanzhou Normal University, Fujian, Quanzhou 362000, China
[2]Department of Computer and Information Science, University of Macau, Taipa 999078, Macau
[3]School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia
[4]School of Medicine, Western Sydney University, Campbelltown, Sydney, NSW 2560, Australia

Corresponding author: Kelvin K. L. Wong (kelvin.wong@westernsydney.edu.au)

This work was supported in part by the University of Macau, RDAO, through Nature-Inspired Computing and Metaheuristics Algorithms for Optimizing Data Mining Performance under Grant MYRG2016-00069-FST, and in part by the Fujian Provincial Key Laboratory of Data-Intensive Computing, Fujian University Laboratory of Intelligent Computing and Information Processing, and Fujian Provincial Big Data Research Institute of Intelligent Manufacturing.

ABSTRACT Recently, a new algorithm named dynamic group optimization (DGO) has been proposed, which is developed to mimic the behaviors of animal and human group socializing. However, one of the major drawbacks of the DGO is the premature convergence. Therefore, in order to deal with this challenge, we introduce chaos theory into the DGO algorithm and come up with a new chaotic dynamic group optimization algorithm (CDGO) that can accelerate the convergence of DGO. Various chaotic maps are used to adjust the update of solutions in CDGO. Extensive experiments have been carried out, and the results have shown that CDGO can be a very promising tool for solving optimization algorithms. We also demonstrated good results based on real world data, where, in particular, solving multimedia data clustering problems.

INDEX TERMS Chaos, dynamic group optimization, convergence, heuristic algorithms.

I. INTRODUCTION

As optimization problems become more and more complex, the use of traditional methods to solve such complicated problems become inefficient. To cope with this problem, many researchers focus on mimicking nature, which is always a rich source of inspiration. Many kinds of heuristic optimization algorithms have recently been proposed to solve this kind of new problems. A series of different meta-heuristic optimization algorithms were proposed, and most of them are inspired by nature. Generally, all these algorithms can be considered as stochastic approach. In comparison with the deterministic approach, it does not need strict process steps and constrains. In most cases, both approaches can be found to churn out an acceptable solution. However, stochastic approaches are much more flexible and universal than deterministic approaches. Meta-heuristic optimization algorithms can be divided into three categories: 1) Evolutionary algorithm, which mimics evolution process. Genetic algorithm (GA) [1], evolutionary strategy (ES) [2], and differential evolution (DE) [3], are the most popular algorithms in this category. 2) Swarm intelligence (SI), algorithms in this category are population based. Some famous algorithms in this branch of algorithms include particle swarm optimization (PSO) [4], ant colony algorithm (ACO) [5], wolf search algorithm (WSA) [6], and cuckoo search algorithm (CS) [7]. 3) Then finally, the other algorithms neither belongs to evolutionary algorithm nor SI. One such example is the famous algorithm based on computing systems of microbial interactions and communications (COMIC) [8].

In most cases, the meta-heuristic algorithms have two phases: exploration and exploitation. Simply put, the exploration phase occurs when the algorithm discover promising search area, and the exploitation phase refers to searching the most promising solution obtained from the exploration phase as quickly as possible [9], [22], [23]. Recently, a new algorithm named the dynamic group optimization algorithm (DGO) [10] is proposed by Tang *et al.*, which mimics the behaviors of animal and human in society. There are three actions in this algorithm: 1) Intra-group cooperation, 2) Inter-group communication, and 3) Group variation. The main advantages of DGO are that it creates a new communication channel between search agents, which accelerate the convergence and enhance the ability of search. DGO is utilized swarm approach into the exploration phase, and used the evolutionary approach into exploitation phase.

Chaos theory is a novel approach that has been wide applications [11], [17], [21]. One of the famous applications is the introduction of chaos theory into optimization. Note that chaos theory is highly sensitive to initial conditions and has a unique feature of randomness [9]. Numerous algorithms can be successfully combined with chaos theory. Some popular chaotic hybrid optimization algorithm includes PSO [12], GA [13] and krill search algorithm [14]. In literature, all these hybrid algorithms outperform the original versions and demonstrate better performance. The choice of chaotic sequences is justified theoretically by their unpredictability, non-periodic, complex temporal behavior, and ergodic properties. However, it is hard to estimate how good most chaotic random number generator is in general by applying statistical tests as they do not follow a uniform distribution. Yang *et al.* [24] states that since the property of probability distribution can enhance search speed in chaos optimization, the ergodicity of chaos implies that chaotic sequences can traverse all the state of strange attractor and search a whole range of search space. As such, this basic property is utilized to search for a global optimum for chaos optimization. The larger is the chaotic sequences, the higher is the chaotic degree, and so the faster is the speed of searching for the whole search space.

Up to now, there are few works on hybridizing chaos theory into DGO for improving the search capability. One of the major drawbacks of the DGO is its premature convergence, especially while handling problems with more local optima. In this paper, sequences generated from chaotic systems substitute random numbers for the DGO parameters where it is necessary to make a random-based choice. In this way, it is intended to improve the global convergence and to prevent it from hovering about a local solution. Although, chaos cannot ensure DGO avoid local entirely because the randomness of DGO, it can enhance the optimization's ergodicity in phase space.

In this paper, we propose using chaotic dynamic search algorithm for the purpose of accelerating the convergence of DGO in order to deal with benchmark functions and multimedia data clustering problems. We integrated ten chaotic maps into this algorithm in order to extensively investigate the effectiveness of chaos theory for improving the search capability. Specifically, our aim is to compare efficiency of different one-dimensional maps as chaotic variable generator in the DGO algorithms. The performance of the proposed approach is tested on fourteen benchmark functions, which are the CEC2009 competition testing functions, which contain unimodal functions and multi modal problems. To test the performance of CDGO on solving the data clustering problems, a multimedia data processing application was carried out for validating the efficiency of our algorithm.

The organization of this paper is as follows: Section II presents the original version of DGO, overview of chaotic maps and details of proposed CDGO. The investigation of CDGO is presented in the Section III. Section IV states the conclusion of this paper, which includes contributions and future studies of CDGO.

II. METHODS

KEY NOTATIONS:

H	Head
$x_{i,j}^G$	The individual of jth member of the ith group at Gth generation.
$v_{i,j}^G$	The trial vector of jth member of the ith group at Gth generation.
$v_{i,j,k}^G$	The trial vector of kth dimension of jth member of the ith group at Gth generation.
$X(i, j)$	The individual of jth member of the ith group.
$X(r1)$	The individual of r1th population.
$f(x)$	The objective function value of x.
$Mr1$	Mutation 1 probability.
$Mr2$	Mutation 2 probability.
$Rand$	Random number generator.
b	Global best obtained so far.
b_k^G	The value of kth dimension of global best at Gth generation.

A. DYNAMIC GROUP OPTIMIZATION ALGORITHM (DGO)

DGO is a new type of meta-heuristic algorithm for solving optimization problems. This algorithm is inspired from intra- and inter- social communications in nature. In this strategy, vectors of solution are considered as members, they are divided into different groups. Each group has a head to record the group best solution. Members can leave a worse group and join a better group, where better group means the group has a better solution. Therefore, the number of members is changing dynamically as iterations/generations grow. It combines evolutionary approach and swarm approach together to accelerate convergence. Moreover, it propose a new structure of data, which divides the population into two parts, heads and members. DGO can be summarized three essential strategies by using different operators to realize a solution.

1. Intra-group cooperation
2. Inter-group communication
3. Group variation

The population of the DGO has members and each member is divided into distinct groups. A group not only has members, but also has a head which stores the best solution obtained so far from that group.

Fig. 1 illustrates the data structure of DGO, where $x_{i,1}$ is the first member of the i^{th} group. The head stores the best solution of the group, in the Fig. 1, we can see that the $x(1, 1)$ is the first member of the first group and it has n dimensional variables. *H(1)* stores the best group solution of the first group.

B. INTRA-GROUP COOPERATION

The intra-group cooperation simulates the members updating procedures. In this component, members of each group are updated by searching information obtained from global best, heads and some other members. In this action, two mutation

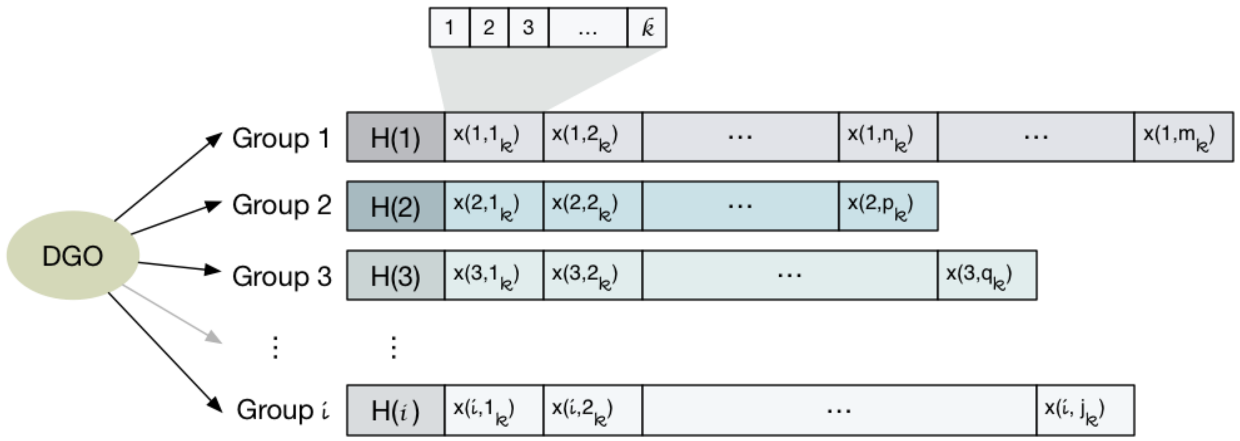

FIGURE 1. Data structure of DGO.

operators are used. The formula is as follows:

$$v_{i,j,k}^{G} = \begin{cases} H_{i,k}^{G} + \mu\left(H_{r,k}^{G} - b_{k}^{G}\right) & if\ rand\ (0,1) > Mr1 \\ x_{i,j,k}^{G} & else, \end{cases} \tag{1}$$

where the $v_{i,j,k}^{G}$ is the vector of kth dimension of jth member of the ith group, which is obtained randomly from the whole population. $H_{i,k}^{G}$ is the kth dimension value of the ith head. G is the generation. We note that b is the best solution obtained so far. Here, r is the random number with the range [0,1]. Then, $rand$ is the random number generator, and μ is drawn from normal distribution with mean 0 and standard deviation 1.

The second operator can be formulated as:

$$v_{i,j,k}^{G} = \begin{cases} b_{k}^{G} + \mu\left(x_{r1,k}^{G} - x_{r2,k}^{G}\right) & if\ rand(0,1) > Mr2 \\ x_{i,j,k}^{G} & else, \end{cases} \tag{2}$$

where r1 and r2 are indexes of two distinct individuals, which are chosen randomly (exclude heads).

C. INTER-GROUP COMMUNICATION

Heads as the groups delegate communicate with the other groups in this phase. The head can update. The movement uses the levy flight random walk. The mathematical update equation of levy flight walk is formulated by Yang and Deb [7] as follows:

$$H_i^{k+1} = H_i^k + \alpha \oplus \text{Lévy}(\lambda), \tag{3}$$

$$\alpha = \alpha_0 \times (H_i^k - b), \tag{4}$$

The $\oplus$ is the entry-wise multiplications. H_i^{k+1} and H_i^k mean the i_{th} group in the $k+1$ and k generations. Lévy(λ) is a random number, which is drawn from Lévy distribution. Here, α_0 is a scaling factor, the b is the global best solution. Equation (3) and (4) are the equations for head updating.

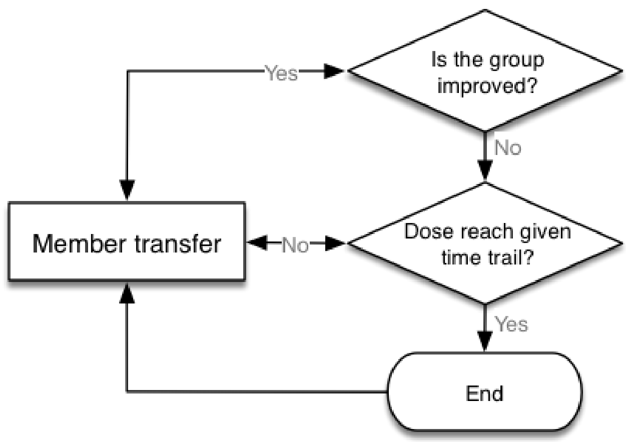

FIGURE 2. Flowchart of group variation.

Heads move towards global best by using Lévy flight walk. The exponential form of probability function is:

$$\text{Lévy} \sim \mu = t^{-\lambda}, \quad \forall 1 < \lambda \leq 3 \tag{5}$$

Mantegna R. proposed the Lévy search equation in 1992 as follow:

$$s = \frac{\emptyset \times \mu}{|v|^{1/\beta}}, \tag{6}$$

Here, $\lambda = 1 + \beta$, $\beta \in (0, 2]$. In the cuckoo search algorithm, we note that $\beta = 1.5$, which is a constant number. Then, μ and v represent the normal distribution random numbers.

$$\emptyset = \left[\frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\beta-1)/2}}\right]^{1/\beta}, \tag{7}$$

From Equations (5)-(7), we see that s is decided by two normal distribution number μ and v.

D. GROUP VARIATION

Group members are changed by ranking the fitness. The better the fitness, the more members we have. In order to

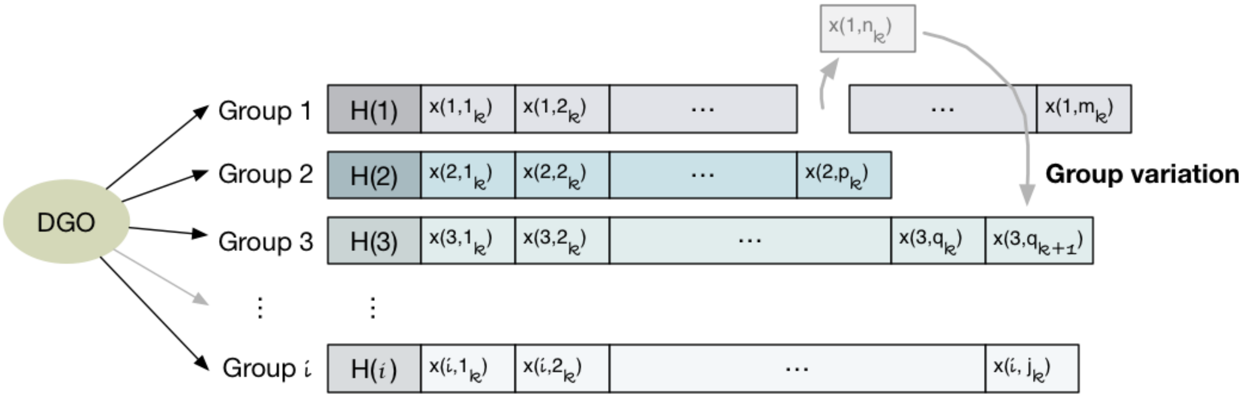

FIGURE 3. Members transfer from the worst group to the other groups.

avoid wastage of computing resources, DGO sets the group variation component so as to control the size of groups. The flow chart demonstrates what we meant:

Fig. 2 shows the procedure of group variation. Firstly, the algorithm determines whether the member of a group is improved or not. If it shows improvement, it will perform an ordinary search. Otherwise, it determines whether it reaches the given number of attempts, which is set by the user. For example, 100 runs are used here. If so, the member cannot jump out of the local optimum, so the member is transferred randomly to another group. This action helps the algorithm to avoid local optima and avoid wasting resource.

Fig. 3 illustrates a snapshot of members transfer in group variation action. We assume that *group 1* holds the worst solution and cannot improve itself within the given times attempts. Then, the group variation action is hereby triggered. The member $x(1, m_k)$ is transferred randomly from *group 1* to another group.

E. CHAOTIC MAPS

Most meta heuristic optimization algorithms belong to stochastic algorithms. The property of randomness is obtained by using probability distribution, such as uniform and Gaussian method. There is a randomness method in optimization field called chaotic optimization (CO) [19]. We note that CO has the property of dynamical, non-repetition and ergodicity. The dynamical property ensures that solutions are produced by the algorithm can be diverse to search all different modal objective search space, even on the complex multimodal landscape. Moreover, by having the ergodicity property of CO, it can perform searches at higher speeds compared to the stochastic algorithms with probability distribution. Becasue chaos theory has the feature of randomness and dynamic, it is easy to accelerate the optimization algorithm convergence and enhance the capability of diversity. In order to achieve such goal, we use one-dimensional and non-invertible maps to produce the chaotic sets. Fig. 4 shows the visualization of these ten chaotic maps. The initial value of a chaotic map has impacts on the fluctuation pattern, but in order to get an unbiased result, we set the initial point as 0.7 for all of which is suggested on [9].

The first is the Chebyshev map, which is a common chaotic map and widely used in digital communication and neural network. It can be defined as follows:

$$x_{k+1} = \cos(k cos^{-1}(x_k)) \tag{8}$$

where the range is $(-1, 1)$. Note that x_k is the *kth* chaotic number, with k denoting the iteration number.

Circle map is a simplified model for both driven mechanical rotors. Furthermore, it is a one-dimensional map which maps a circle onto itself. Circle map is presented as follows:

$$x_{k+1} = x_k + b - \left(\frac{a}{2\pi}\right) sin\,(2\pi x_k)\, mod\,(1)\,, \tag{9}$$

where $a = 0.5$ and $b = 0.2$, the range is $(0, 1)$, the parameters b and a can be regarded as strength of nonlinearity and externally applied frequency, separately. The Circle map produces much unexpected behavior with the change of parameters.

Gauss/Mouse map can be described as follows:

$$x_{k+1} = \begin{cases} 0 & x_k \\ \dfrac{1}{x_k mod(1)} & otherwise, \end{cases} \tag{10}$$

This map also generates chaotic sequences in $(0, 1)$.

Iterative map is one with infinite collapses, which can be presented as follows:

$$x_{k+1} = sin(a\pi/x_k), \tag{11}$$

where $a = 0.7$ and the chaotic sequence in $(-1, 1)$.

Logistic map can be written as follows:

$$x_{k+1} = ax_k(1 - x_k), \tag{12}$$

where $a = 4$ and the range is $(0, 1)$, it is the simplest map that appears in nonlinear dynamics of biological population evidencing chaotic behavior is logistic map.

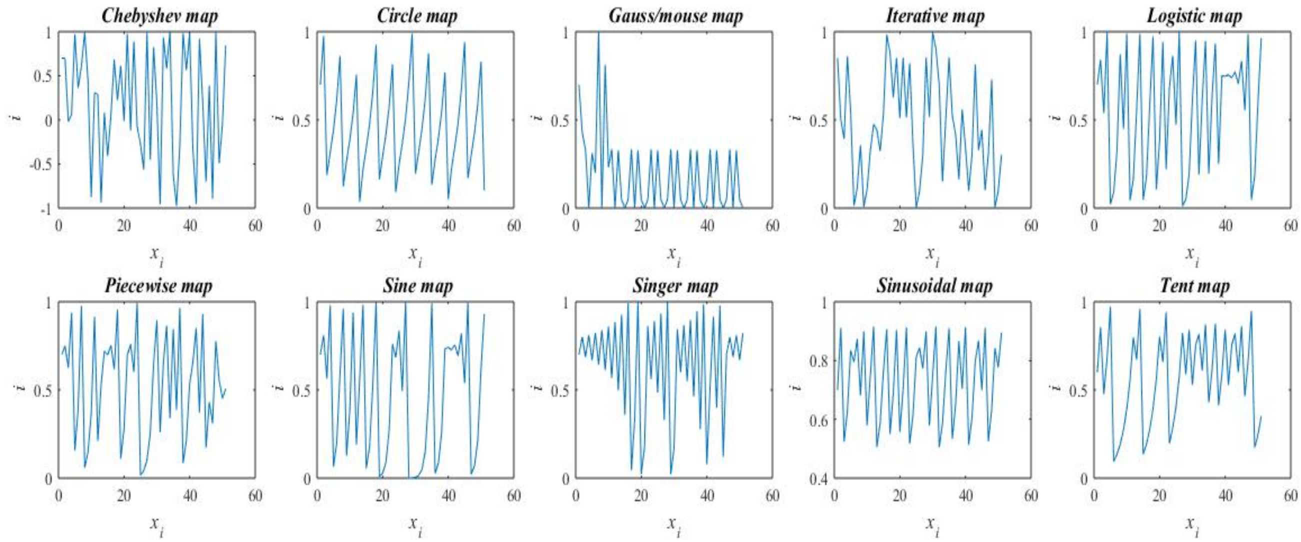

FIGURE 4. Visualization of employed ten chaotic maps on one dimensional space.

Piecewise map is governed by the following equation

$$x_{k+1} = \begin{cases} \dfrac{x_k}{P} & 0 \le x_k < P \\ \dfrac{x_k - P}{0.5 - P} & P \le x_k < 1/2 \\ \dfrac{1 - P - x_k}{0.5 - P} & \dfrac{1}{2} \le x_k < 1 - P \\ \dfrac{1 - x_k}{P} & 1 - P \le x_k < 1, \end{cases} \tag{13}$$

where $P = 0.4$ and the range is (0, 1).

The Sine map belongs to a unimodal map and is similar to the Logistic map, which can be described as follows:

$$x_{k+1} = a/4sin(\pi x_k), \tag{14}$$

where $a = 4$ and the chaotic sequence in (0, 1).

Singer map is a one-dimensional system as given below as:

$$x_{k+1} = \mu(7.86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.3x_k^4), \tag{15}$$

where $\mu = 1.07$ and the range is (0, 1).

Sinusoidal map can be defined as follows:

$$x_{k+1} = ax_k^2 sin(\pi x_k), \tag{16}$$

where $a = 2.3$ and the range is (0, 1).

Tent chaotic map is very similar to the logistic map, which displays specific chaotic effects [25]

Tent map can be described as follows:

$$x_{k+1} = \begin{cases} \dfrac{x_k}{0.7} & x_k < 0.7 \\ \dfrac{10}{3}(1 - x_k) & otherwise, \end{cases} \tag{17}$$

In order to get an unbiased results, we set the initial point is 0.7 for all chaotic maps in this work.

F. PROPOSED CHAOTIC DYNAMIC GROUP OPTIMIZATION ALGORITHM

In this section, we used the chaotic maps in two ways to combine with DGO and to improve its performance. We employed the chaotic maps for manipulating the intra-group cooperation and the inter-group communication of the DGO. Chaotic intra-group cooperation improves the exploitation whereas the chaotic inter-group enhances the capability of exploration.

1) CHAOTIC INTRA-GROUP COOPERATION

As it is presented in Eq. (1), the parameter *rand (0, 1)* is the key parameter to control the mutation operator process, the value of the parameter is vitally important to decide the convergence speed. In our work, we employed the chaotic to redefine the probability. The *rand* is replaced by chaotic map, the new chaotic equation is shown as follows:

$$v_{i,j,k}^G = \begin{cases} H_{i,k}^G + \mu\left(H_{r,k}^G - b_k^G\right) & if\ C(t) > Mr1 \\ x_{i,j,k}^G & else, \end{cases} \tag{18}$$

where $C(t)$ is the value of the chaotic map in the *tth* iteration. In the original DGO, the value is totally random. And when comparing with the chaotic method, it is lacking different exploitation patterns. It is worthwhile to mention that the range of some chaotic maps are (−1, 1), so we normalize them into (0, 1).

2) CHAOTIC INTER-GROUP COMMUNICATION

As can be seen in Eq. (6), the parameter μ is the random number draw from normal distribution, and μ is the key parameter to produce the new step size. We utilized the chaotic map

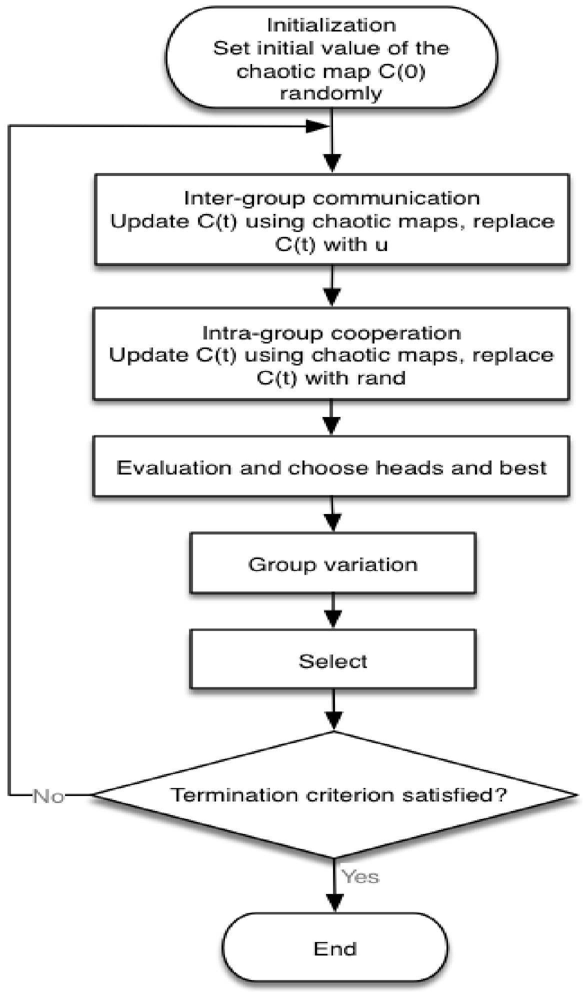

FIGURE 5. Flowchart of CDGO.

calculates the s as follows:

$$s = \frac{\emptyset \times C(t)}{|v|}^{1/\beta}, \tag{19}$$

where the $C(t)$ is the value of the chaotic map of the *t-th* iteration. Eq. (19) shows that the chaotic maps are allowed to define the chaotic inter-group communication and chaotic maps are responsible for producing the new step size of movement.

The tuning of parameters when using chaotic maps not only improves the ability of speed of convergence, but it also improves the ability of avoiding local optima. The following shows the advantage of this chaotic method. Chaotic intra-group cooperation improves CDGO to control the process of mutation, which improves the exploitation. Chaotic intra-group cooperation helps CDGO to produce new step size with a chaotic pattern, which improves the exploration. Since the chaotic maps show the chaotic behaviors, the chaotic algorithm can assist the DGO to exit the local optima. Fig. 5 shows the flowchart of CDGO.

TABLE 1. Computer environment setting.

Name	Setting
CPU	I7-3.60GHz
RAM	8.00GB
Software	MATLAB R2015a

III. EXPERIMENTAL RESULTS

To fully evaluate the performance of the DGO without a biased conclusion, we carried out different experiments. Fourteen benchmark functions were employed in our experimental studies. These functions are widely used in numerous studies [15], [16]. All experiments are performed using the same PC, and the details setting is shown in Table 1:

In our experiments, we use the average and standard deviation of the function value to compare the performance of algorithms. The maximum number of fitness evaluations that we allowed for each algorithm to minimize this value was $1000{*}D$, where D is the dimension of the problem. The fitness evaluation criteria is shown as follows:

Function value: The minimum function value of each algorithm is recorded in 50 runs, and the average and the standard deviation of the value are calculated. The average and the standard deviation are recorded in our experiment.

P-value: Wilcoxon's rank sum test. The 0.05 significant level is used to assess significance between two algorithms. P-value is calculated, which is a number in the range [0, 1] and it is the probability of observing the value under the null hypothesis.

The initial population of testing algorithms is generated uniformly at random in the search space. In all experiments, the same population size, and mutation probability is used. There are some parameters are important to DGO and CDGO: the number of groups, which decides the centroid, is 9 in our experiment. The probability of mutation pm is 0.2. Population size P is 30, and the dimension of testing function is 30. The marks—CDGO1, CDGO2,…, CDGO10 in the following paper are abbreviated for the ten corresponding chaotic maps in Section II.

A. CHAOTIC INTRA-GROUP COOPERATION

The performance of CDGOs with intra-group cooperation phase was compared with classical DGO on benchmark functions. Table 2 lists the mean, standard deviation and p-value of function values respectively, which are obtained from testing benchmark functions. The results show CDGO outperforms the original DGO in terms of the average function value at all functions except *f1*, *f4* and *f14*. In the unimodal function *f1-f7*, CDGOs outperforms DGO on 5 out of 7 functions. In multimodal function, CDGOs outperforms DGO on 6 out of 7 functions. The result of standard deviation function values is similar as the mean function value. Compared to classical DGO, we can see that most results of CDGO are a significant improvement in terms of p-value.

TABLE 2. The comparison between DGO and different CDGOs using chaotic maps at intra-groups action.

Fun.	Evaluation Metric	DGO	CDGO1	CDGO2	CDGO3	CDGO4	CDGO5	CDGO6	CDGO7	CDGO8	CDGO9	CDGO10
F1	Average function values	1.03E-69	1.65E-34	5.72E-65	1.46E-18	1.73E-18	2.58E-116	5.60E-46	8.74E-35	4.14E-28	2.98E-27	4.76E-18
	Std function values	2.31E-69	3.69E-34	1.28E-64	3.12E-18	3.87E-18	5.77E-116	1.25E-45	1.96E-34	9.25E-28	6.67E-27	1.07E-17
	p-value	-	8.41E-01	1.00E+00	1.00E+00	7.94E-03	8.41E-01	6.90E-01	4.21E-01	8.41E-01	5.48E-01	2.22E-01
F2	Average function values	2.48E-07	7.45E-21	4.18E-10	2.54E-09	8.73E-20	2.64E-51	8.80E-15	4.50E-17	6.29E-18	3.68E-25	1.15E-24
	Std function values	5.55E-07	1.67E-20	9.36E-10	3.48E-09	1.95E-19	5.90E-51	1.97E-14	1.01E-16	1.41E-17	8.23E-25	2.54E-24
	p-value	-	8.41E-01	5.48E-01	5.48E-01	1.51E-01	6.90E-01	1.00E+00	1.00E+00	8.41E-01	1.00E+00	6.90E-01
F3	Average function values	6.26E-08	1.43E-07	8.81E-08	2.91E-03	2.59E-20	4.51E-15	4.38E-07	2.78E-20	2.32E-10	5.10E-08	1.17E-08
	Std function values	1.40E-07	3.20E-07	1.97E-07	4.19E-03	5.78E-20	9.96E-15	9.79E-07	6.21E-20	5.18E-10	1.14E-07	2.13E-08
	p-value	-	4.21E-01	6.90E-01	6.90E-01	7.94E-03	3.10E-01	1.00E+00	6.90E-01	4.21E-01	4.21E-01	1.00E+00
F4	Average function values	1.65E-02	2.82E-02	2.71E-02	1.37E-01	2.40E-02	2.75E-02	1.90E-02	3.53E-02	3.28E-02	1.78E-02	2.17E-02
	Std function values	1.29E-03	1.10E-02	1.04E-02	7.29E-02	1.05E-02	7.82E-03	7.09E-03	1.02E-02	1.34E-02	8.88E-03	1.24E-02
	p-value	-	7.94E-03	3.17E-02	3.17E-02	7.94E-03	2.22E-01	7.94E-03	5.48E-01	7.94E-03	7.94E-03	6.90E-01
F5	Average function values	4.11E+00	1.13E+01	7.36E+00	4.74E+00	6.16E+00	1.08E+00	7.20E+00	7.71E-02	4.71E+00	5.28E+00	6.57E+00
	Std function values	3.83E+00	9.40E+00	7.35E+00	8.15E+00	8.48E+00	1.12E+00	5.80E+00	4.57E-02	9.78E+00	5.12E+00	7.08E+00
	p-value	-	2.22E-01	4.21E-01	4.21E-01	1.00E+00	8.41E-01	6.90E-01	5.48E-01	5.48E-01	1.00E+00	5.48E-01
F6	Average function values	2.22E-19	8.74E-22	6.01E-24	2.40E-18	3.20E-24	1.58E-22	4.81E-19	7.73E-21	5.83E-26	4.28E-20	1.91E-20
	Std function values	4.96E-19	1.25E-21	1.10E-23	4.71E-18	4.89E-24	3.50E-22	8.30E-19	1.72E-20	1.21E-25	9.57E-20	4.01E-20
	p-value	-	6.90E-01	6.90E-01	6.90E-01	9.52E-02	8.41E-01	1.00E+00	1.51E-01	1.51E-01	9.52E-02	5.48E-01
F7	Average function values	3.99E-03	4.04E-03	6.31E-03	3.49E-03	6.18E-03	6.44E-03	7.56E-03	4.84E-03	7.35E-03	6.10E-03	6.92E-03
	Std function values	2.13E-03	1.76E-03	1.98E-03	1.79E-03	3.29E-03	2.77E-03	5.37E-03	2.84E-03	8.11E-03	4.95E-03	4.02E-03
	p-value	-	8.41E-01	9.52E-02	9.52E-02	6.90E-01	4.21E-01	1.51E-01	3.10E-01	4.21E-01	6.90E-01	2.22E-01
F8	Average function values	-1.07E+04	-1.02E+04	-1.04E+04	-1.04E+04	-1.04E+04	-1.05E+04	-1.04E+04	-1.07E+04	-1.06E+04	-1.05E+04	-1.04E+04
	Std function values	2.87E+02	2.44E+02	1.35E+02	6.47E+02	5.92E+02	2.67E+02	4.66E+02	5.22E+02	2.72E+02	3.25E+02	5.52E+02
	p-value	-	9.52E-02	9.52E-02	9.52E-02	4.21E-01	4.21E-01	4.21E-01	5.48E-01	1.00E+00	8.41E-01	1.51E-01
F9	Average function values	1.99E-01	1.71E-08	3.87E-13	3.98E-01	2.66E-10	1.99E-01	1.99E-01	1.99E-01	9.30E-10	6.24E-12	3.98E-01
	Std function values	4.45E-01	3.83E-08	2.32E-13	5.45E-01	5.94E-10	4.45E-01	4.45E-01	4.45E-01	2.08E-09	1.30E-11	5.45E-01
	p-value	-	9.52E-02	2.06E-01	2.06E-01	1.51E-01	2.30E-01	8.41E-01	7.94E-01	9.68E-01	5.79E-01	7.46E-01
F10	Average function values	5.54E-12	7.87E-11	2.69E-11	9.84E-08	1.53E-11	1.04E-12	1.78E-12	2.63E-12	7.56E-09	4.06E-13	1.93E-13
	Std function values	6.82E-12	1.33E-10	5.93E-11	1.67E-07	3.13E-11	1.05E-12	2.25E-12	3.52E-12	1.69E-08	3.12E-13	1.36E-13
	p-value	-	2.22E-01	5.48E-01	5.48E-01	7.94E-03	8.41E-01	3.10E-01	4.21E-01	6.90E-01	1.00E+00	1.51E-01
F11	Average function values	1.33E-02	1.23E-02	1.18E-02	2.40E-02	6.79E-14	1.23E-02	1.57E-02	2.64E-02	1.72E-02	8.86E-03	1.13E-02
	Std function values	1.36E-02	1.25E-02	2.12E-02	3.29E-02	1.44E-13	1.79E-02	1.61E-02	3.26E-02	2.35E-02	1.32E-02	1.91E-02
	p-value	-	8.41E-01	5.87E-01	5.87E-01	7.46E-01	6.35E-02	1.00E+00	8.41E-01	9.37E-01	1.00E+00	5.48E-01

TABLE 2. ***(Continued.)*** **The comparison between DGO and different CDGOs using chaotic maps at intra-groups action.**

F12	Average function values	3.98E-26	2.07E-02	4.15E-02	3.83E-11	3.54E-19	2.07E-02	2.54E-13	2.07E-02	2.22E-21	5.67E-20	5.47E-11
	Std function values	8.51E-26	4.64E-02	5.68E-02	8.56E-11	7.91E-19	4.64E-02	5.68E-13	4.64E-02	4.96E-21	1.26E-19	1.22E-10
	p-value	-	4.21E-01	5.48E-01	5.48E-01	7.94E-03	1.59E-02	1.59E-02	1.59E-02	1.51E-01	4.21E-01	1.51E-01
F13	Average function values	5.10E-21	4.42E-24	2.22E-19	3.88E-11	4.10E-20	1.28E-19	3.36E-22	5.04E-21	7.00E-18	2.66E-18	6.63E-24
	Std function values	9.41E-21	9.71E-24	3.08E-19	8.68E-11	8.99E-20	2.87E-19	6.04E-22	1.11E-20	1.57E-17	5.83E-18	7.87E-24
	p-value	-	5.48E-01	4.21E-01	4.21E-01	3.17E-02	8.41E-01	6.90E-01	1.00E+00	5.48E-01	6.90E-01	8.41E-01
F14	Average function values	1.79E+00	1.39E+00	9.98E-01	1.79E+00	4.90E+00	1.79E+00	4.90E+00	9.98E-01	3.35E+00	5.30E+00	1.39E+00
	Std function values	1.09E+00	8.87E-01	2.48E-16	1.09E+00	5.35E+00	1.09E+00	5.35E+00	0.00E+00	4.23E+00	5.05E+00	8.87E-01
	p-value	-	1.00E+00	6.83E-01	6.83E-01	4.13E-01	5.24E-01	6.83E-01	6.83E-01	4.44E-01	4.13E-01	1.00E+00

B. CHAOTIC INTER-GROUP COMMUNICATION

Chaotic maps are used into inter-group communication to tune the exploration of DGO. Table 3 shows the result. From Table III, in terms of the average function values, CDGOs performs better than DGO on $f1$, $f2$, $f5$, $f6$, $f7$, $f9$, $f10$, $f11$ and $f12$, and they all found the global optimum. Moreover, compared to DGO. CDGOs get 8 times best result. We can know more information about Wilcoxon's rank sum test from Table 3. Our proposed CDGOs have achieved a significant improvement over the original algorithms. It can be observed that the CDGOs perform better than the classical DGO when the DGO is combined to chaotic maps of inter-groups actions.

C. CHAOTIC DGO COMBINED WITH CHAOTIC INTER AND INTRA GROUP ACTIVITIES

Results of the CDGO algorithms pertaining to both intra and inter group activities operators are provided in Table IV. It can be observed that the results of all CDGO algorithms are much better than the original DGO. All of the CDGO algorithms provide superior results compared to the DGO algorithm except $f4$ and $f9$. The p-value in Table 4 also indicates that the DGO algorithm provides the worst rates. According to Table IV, CDGO2 uses the circle map to enhance the performance of the DGO algorithm remarkably. Since CDGO2 provides the best results for the 4 tests' functions, such as unimodal and multimodal functions. It can be stated that the combination of a circle map and two operators improve the exploration and exploitation of the DGO algorithm significantly.

D. PERFORMANCE TEST AND ANALYSIS OF CHAOTIC MAPS

The unstable and chaotic sequence can produce non-repetition and dynamical search pattern, which is helpful to enhance the capability of exploration phase and capability of jumping out of local optimum. Therefore, in this section, our goal is to compare DGO and CDGO using different chaotic maps described above as Lyapunov exponent λ to measures the average exponential divergence or convergence rate. We distinguish 3 cases of λ. The first is $\lambda < 0$: A negative λ is characteristic of dissipative or non-conservative system. This kind of system exhibits asymptotic stability. The second case is $\lambda = 0$: A Lyapunov exponent of zero indicates that the system is in a steady-state mode or near the transition to chaos. The third one is $\lambda > 0$, the orbit is unstable and chaotic, in our comparison, the λ is greater the search pattern is better.

We extracted sequences from classical DGO and CDGO with different chaotic maps under the same initial condition to record the Lyapunov exponent. Moreover, we set a hypothesis testing to further check whether the sequences are chaotic or not based on a confidence level 0.5. The test hypothesis H are: null hypothesis H0: $\lambda > 0$, which indicates the presence of chaos; and alternative hypothesis H1: $\lambda < 0$, which indicates the absence of chaos. All the initial value is set to be 0.7 and the results are reported.

From Table 5, it can be shown that Lyapunov exponent λ of random variable is -0.496 and the $P = 4.35\text{e-}11$. Hence, the hypothesis H1 indicating the absence of chaos is accepted at 5% confidence level. The random variable is therefore stochastic. The other CGDO algorithm with chaotic maps had positive Lyapunov exponent λ, and the hypothesis H0 indicating the presence of chaos were accepted at 5% confidence level. The CDGO with Circle map obtained the highest Lyapunov exponent, which means that CDGO with Circle maps can produce the most unstable chaotic sequence. It also means that the Circle map is the most suitable chaotic map for applying into DGO. Chaotic sequences influence the behavior of all operators (mutation, crossover), not because new operators are introduced, but because all the existing standard operators work following the outcomes of a chaotic sequence instead of a standard random generator. The properties of chaos guarantee solutions produced by algorithms can be sufficiently diverse to search all different landscapes of search space, and the ergodicity and non-repetition enhance the speed of search.

TABLE 3. The comparison between DGO and different CDGOs using chaotic maps at inter-groups action.

Fun.	Evaluation Metric	DGO	CDGO1	CDGO2	CDGO3	CDGO4	CDGO5	CDGO6	CDGO7	CDGO8	CDGO9	CDGO10
F1	Average function values	1.55E-20	6.68E-61	7.43E-16	1.22E-103	9.62E-106	6.65E-23	2.67E-15	3.07E-55	2.62E-18	4.01E-63	2.40E-23
	Std function values	3.47E-20	1.49E-60	1.66E-15	2.73E-103	2.15E-105	1.49E-22	5.96E-15	6.87E-55	5.86E-18	8.96E-63	5.37E-23
	p-value	-	8.41E-01	6.90E-01	6.90E-01	6.90E-01	1.00E+00	1.00E+00	4.21E-01	8.41E-01	6.90E-01	5.48E-01
F2	Average function values	1.56E-55	3.27E-07	5.85E-10	1.21E-34	7.73E-56	1.33E-49	9.59E-15	2.84E-28	1.16E-20	3.00E-14	9.13E-08
	Std function values	3.50E-55	7.30E-07	1.31E-09	2.08E-34	1.73E-55	2.98E-49	2.14E-14	6.35E-28	2.58E-20	6.72E-14	2.04E-07
	p-value	-	1.51E-01	8.41E-01	8.41E-01	5.56E-02	6.90E-01	8.41E-01	6.90E-01	8.41E-01	1.51E-01	1.59E-02
F3	Average function values	5.07E-21	1.61E-08	6.03E-12	1.53E-08	5.88E-07	1.48E-13	2.21E-07	6.06E-06	5.82E-08	1.22E-09	1.76E-12
	Std function values	1.11E-20	3.61E-08	1.35E-11	3.42E-08	1.32E-06	3.31E-13	4.95E-07	1.35E-05	1.30E-07	2.72E-09	3.70E-12
	p-value	-	9.52E-02	5.48E-01	5.48E-01	3.17E-02	4.21E-01	3.10E-01	4.21E-01	6.90E-01	2.22E-01	4.21E-01
F4	Average function values	2.42E-02	2.79E-02	3.80E-02	3.53E-02	3.50E-02	2.66E-02	3.52E-02	3.55E-02	3.34E-02	3.40E-02	3.65E-02
	Std function values	9.50E-03	1.35E-02	1.47E-02	1.46E-02	1.09E-02	3.99E-03	1.05E-02	1.02E-02	1.06E-02	1.99E-02	9.90E-03
	p-value	-	1.00E+00	1.51E-01	1.51E-01	1.51E-01	9.52E-02	5.48E-01	9.52E-02	1.51E-01	1.51E-01	9.52E-02
F5	Average function values	3.45E+00	7.20E+00	9.76E+00	1.04E+01	4.46E+00	1.85E+01	8.30E-01	4.34E-01	1.05E+01	1.86E+00	6.42E+00
	Std function values	7.21E+00	9.89E+00	8.85E+00	6.68E+00	7.01E+00	3.09E+01	7.26E-01	7.30E-01	1.01E+01	3.84E+00	6.99E+00
	p-value	-	5.48E-01	5.48E-01	5.48E-01	1.51E-01	3.10E-01	1.51E-01	8.41E-01	5.48E-01	5.48E-01	5.48E-01
F6	Average function values	5.02E-20	6.72E-15	6.48E-20	8.38E-22	3.15E-19	3.17E-13	5.27E-18	8.10E-21	4.46E-21	6.82E-17	2.09E-20
	Std function values	1.12E-19	1.50E-14	8.92E-20	1.34E-21	7.05E-19	7.08E-13	1.16E-17	1.77E-20	9.47E-21	1.52E-16	4.67E-20
	p-value	-	6.90E-01	6.90E-01	6.90E-01	8.41E-01	4.21E-01	8.41E-01	1.00E+00	8.41E-01	4.21E-01	6.90E-01
F7	Average function values	6.54E-03	5.54E-03	7.32E-03	9.57E-03	6.02E-03	6.92E-03	6.46E-03	5.67E-03	6.39E-03	4.58E-03	6.64E-03
	Std function values	2.57E-03	1.87E-03	3.95E-03	9.58E-03	2.88E-03	5.53E-03	3.02E-03	1.97E-03	2.48E-03	2.23E-03	4.45E-03
	p-value	-	5.48E-01	8.41E-01	8.41E-01	1.00E+00	6.90E-01	6.90E-01	8.41E-01	6.90E-01	1.00E+00	1.00E+00
F8	Average function values	-1.06E+04	-1.05E+04	-1.02E+04	-1.04E+04	-1.03E+04	-1.03E+04	-9.89E+03	-1.05E+04	-1.05E+04	-1.05E+04	-1.01E+04
	Std function values	4.94E+02	1.08E+02	2.56E+02	3.11E+02	2.22E+02	3.09E+02	6.08E+02	3.73E+02	2.86E+02	2.65E+02	5.30E+02
	p-value	-	3.10E-01	1.51E-01	1.51E-01	8.41E-01	3.10E-01	5.48E-01	9.52E-02	6.90E-01	8.41E-01	2.22E-01
F9	Average function values	3.98E-01	5.47E-12	3.98E-01	1.99E-01	6.93E-13	3.98E-01	1.99E-01	1.99E-01	5.97E-01	3.35E-12	5.97E-01
	Std function values	5.45E-01	8.45E-12	5.45E-01	4.45E-01	1.08E-12	5.45E-01	4.45E-01	4.45E-01	8.90E-01	5.96E-12	8.90E-01
	p-value	-	1.98E-01	7.30E-01	7.30E-01	5.48E-01	1.59E-01	4.13E-01	5.24E-01	1.35E-01	9.52E-01	1.00E+00
F10	Average function values	5.25E-13	3.41E-12	3.18E-12	1.44E-12	3.38E-12	9.56E-13	6.36E-13	1.41E-12	2.72E-11	4.70E-08	1.48E-12
	Std function values	4.30E-13	3.47E-12	3.94E-12	2.56E-12	4.36E-12	8.29E-13	1.07E-12	2.13E-12	5.75E-11	1.04E-07	2.59E-12
	p-value	-	5.56E-02	1.51E-01	1.51E-01	1.00E+00	3.97E-02	4.21E-01	4.52E-01	6.90E-01	4.21E-01	8.89E-01
F11	Average function values	1.71E-02	1.41E-02	3.43E-02	3.94E-03	2.40E-02	1.38E-02	3.23E-02	1.48E-03	3.03E-02	3.53E-02	7.87E-03
	Std function values	3.17E-02	3.16E-02	3.73E-02	8.80E-03	2.70E-02	1.37E-02	5.50E-02	3.31E-03	4.72E-02	2.24E-02	1.38E-02
	p-value	-	1.00E+00	4.21E-01	4.21E-01	1.00E+00	6.90E-01	8.41E-01	1.00E+00	8.41E-01	6.90E-01	1.00E+00

TABLE 3. ***(Continued.)*** **The comparison between DGO and different CDGOs using chaotic maps at inter-groups action.**

F12	Average function values	1.39E-19	1.33E-18	4.07E-22	7.87E-15	2.09E-17	3.75E-15	7.10E-24	1.43E-22	3.85E-24	2.07E-02	1.68E-13
	Std function values	1.94E-19	2.97E-18	9.11E-22	1.76E-14	4.67E-17	8.38E-15	1.13E-23	3.11E-22	8.35E-24	4.64E-02	3.76E-13
	p-value	-	6.90E-01	4.21E-01	4.21E-01	1.00E+00	6.90E-01	5.48E-01	5.48E-01	8.41E-01	4.21E-01	6.90E-01
F13	Average function values	1.09E-19	4.95E-17	5.85E-17	1.03E-15	2.33E-21	2.04E-18	9.98E-20	5.48E-22	2.80E-20	6.07E-22	9.22E-19
	Std function values	1.56E-19	1.11E-16	1.31E-16	2.31E-15	5.20E-21	4.57E-18	2.23E-19	1.22E-21	5.30E-20	1.34E-21	2.06E-18
	p-value	-	1.51E-01	8.41E-01	8.41E-01	1.00E+00	1.51E-01	1.51E-01	6.90E-01	5.56E-02	1.00E+00	1.51E-01
F14	Average function values	9.98E-01	1.79E+00	1.79E+00	9.98E-01	2.95E+00	5.29E+00	1.39E+00	2.95E+00	3.33E+00	1.39E+00	6.48E+00
	Std function values	1.11E-16	1.09E+00	1.09E+00	2.22E-16	4.37E+00	5.91E+00	8.87E-01	4.37E+00	5.22E+00	8.87E-01	4.83E+00
	p-value	-	4.44E-01	4.44E-01	4.44E-01	7.22E-01	7.22E-01	1.27E-01	4.44E-01	1.00E+00	1.00E+00	7.94E-03

In order to fully investigate the performance of our proposed CDGO, we also carried out a comparison between CDGO and two optimization algorithms and two well-known swarm intelligence optimization algorithms, which includes CPSO [11], CKH [14], WSA [6] and PSO [4]. We choose CDGO2 as our representative of CDGO.

The results are reported in Table 6, which clearly shows that the CDGO algorithm overall yields better results than the other algorithms on all functions. For example, on function f1, CDGO2 reached 1e-16, but the other algorithms merely obtained solution at 1e-5. From Table 6, we can draw the similar conclusion that our proposed CDGO is useful tool for solving the optimization problems.

E. CHAOTIC DGO SOLVING REAL WORLD CLUSTERING PROBLEM

In order to fully investigate the performance of CDGO, we employed CDGO to solve real world multimedia data set clustering problems. The multimedia data are, unstructured, loose and dynamical, and they are difficult to cluster. We note that k-means is a simple but powerful algorithm for clustering. Objects that are similar in the same group, but different in the other groups is a characteristics of k-means. It starts with K initial cluster centroids and assigns each object to the nearest centroid conveniently. The core of k-means algorithm is updating centroids and reassignment of group objects. The mathematical equations are shown as follows:

$$clamt = \min_{k \in K}\{||x_i - cen_k||\}, \tag{20}$$

Here, x represents objects $(x_1, x_2, \ldots, x_n)$, where each object is a D-dimensional real vector. cen_k is the mean of points in the kth cluster. Now, k-means is computationally difficult (which makes it a NP-hard problem). Many studies show it converges quickly to a local optimum that does not achieve the best clustering result. Therefore, it quite natural to apply the heuristic optimization algorithm to k-means for the aid of searching the global optimum in each computational iteration. Wolf search clustering (Cwolf) [6], cuckoo search clustering (Ccuckoo) [18] and PSO clustering (CPSO) [18] are well-known algorithms in hybrid k-means algorithms. In order to test the performance of CDGO on multimedia data clustering, we also hybridized k-means with CDGO (CCDGO). There are three actions in CCDGO, initialization, exploration and cluster assignment. Each search agent in CCDGO holds a set of centroids $(cen_1, cen_2, \ldots, cen_K)$ with $K \times D$ dimension, which is computed iteratively. In the initialization, each search agent selects K objects randomly from whole data set to take as the initial centroids, then the rest objects are assigned to the nearest cluster based on Eq. (20). The second action is exploration. CDGO plays an important role here. Due to the search agent during each iteration is always tending to update a better solution so as to find the optimal combination of centroids, which is the goal of each search agent in CCDGO. The update centroid equations are shown as follows:

$$w_{i,j} = \begin{cases} 1, & x_i \in cluter_j \\ 0, & x_i \notin cluter_j, \end{cases} \tag{21}$$

$$cen_{j,v} = \frac{\sum_{i=1}^{S} w_{i,j} x_{i,v}}{\sum_{i=1}^{S} w_{i,j}}, \quad j = ..K, \ v = 1..K * D, \tag{22}$$

where S is the solution space has several x_i solutions, i is the index of the solution. $cen_{j,v}$ is the centroids at the jth cluster and the vth attribute. The objective function can be described as follows:

$$F(cen) = \sum_{j=1}^{K} \sum_{i=1}^{S} w_{i,j} \sum_{v=1}^{K*D} (x_{i,v} - cen_{j,v})^2, \tag{23}$$

To measure the distance between each x and the centroid, the calculation process loops $K \times D$ times to consider the values of all the attributes of x in every dimension v. It is worthwhile to mention that each centroid cen is required to be split into K segments from every search agent. The last action is the cluster assignment. CCDGO ranks search agents based on (23), which finds the best solution as the centroids and reassigns all objects to the nearest cluster. The flowchart of CCDGO shows in the Fig. 6.

TABLE 4. The comparison between DGO and different CDGOs using chaotic maps at intra and inter-groups action.

Fun.	Evaluation Metric	DGO	CDGO1	CDGO2	CDGO3	CDGO4	CDGO5	CDGO6	CDGO7	CDGO8	CDGO9	CDGO10
F1	Average function values	2.26E-86	1.45E-24	1.72E-97	1.57E-09	2.16E-50	1.46E-20	4.62E-36	4.30E-30	4.00E-71	4.66E-21	3.50E-38
	Std function values	5.06E-86	3.24E-24	3.85E-97	2.31E-09	4.83E-50	3.26E-20	1.03E-35	7.67E-30	8.95E-71	1.04E-20	7.82E-38
	p-value	-	9.52E-02	4.21E-01	4.21E-01	7.94E-03	4.21E-01	5.56E-02	2.22E-01	3.17E-02	4.21E-01	6.90E-01
F2	Average function values	2.14E-15	1.33E-09	1.53E-08	1.97E-05	6.22E-24	7.39E-11	6.81E-09	1.24E-33	8.42E-07	1.60E-10	2.19E-15
	Std function values	4.79E-15	2.98E-09	2.39E-08	2.80E-05	1.39E-23	1.65E-10	1.38E-08	2.78E-33	1.88E-06	3.58E-10	4.89E-15
	p-value	-	1.51E-01	3.10E-01	3.10E-01	7.94E-03	9.52E-02	1.00E+00	8.41E-01	5.56E-02	9.52E-02	5.48E-01
F3	Average function values	3.12E-13	3.92E-22	5.60E-21	3.93E-02	6.82E-17	1.40E-08	1.25E-06	4.53E-06	1.95E-18	1.10E-13	4.55E-06
	Std function values	6.97E-13	8.10E-22	1.25E-20	6.33E-02	1.51E-16	3.13E-08	2.80E-06	9.87E-06	4.33E-18	2.43E-13	1.02E-05
	p-value	-	3.10E-01	4.21E-01	4.21E-01	7.94E-03	1.00E+00	1.00E+00	6.90E-01	5.48E-01	1.00E+00	6.90E-01
F4	Average function values	2.22E-02	3.30E-02	4.99E-02	1.98E-01	2.75E-02	3.38E-02	3.05E-02	3.01E-02	2.83E-02	3.77E-02	3.78E-02
	Std function values	7.09E-03	6.00E-03	1.80E-02	7.76E-02	6.18E-03	9.27E-03	6.63E-03	7.80E-03	3.24E-03	1.56E-02	1.81E-02
	p-value	-	3.17E-02	1.59E-02	1.59E-02	7.94E-03	4.21E-01	9.52E-02	5.56E-02	1.51E-01	1.51E-01	2.22E-01
F5	Average function values	4.41E+00	6.92E+00	9.70E+00	7.01E+00	2.07E+00	4.45E+00	7.38E+00	8.21E+00	4.23E+00	4.43E+00	1.02E+01
	Std function values	6.05E+00	8.08E+00	8.31E+00	7.82E+00	2.93E+00	5.88E+00	1.04E+01	9.07E+00	7.54E+00	6.84E+00	9.18E+00
	p-value	-	6.90E-01	4.21E-01	4.21E-01	5.48E-01	8.41E-01	1.00E+00	1.00E+00	5.48E-01	8.41E-01	2.22E-01
F6	Average function values	3.72E-19	1.53E-21	4.21E-18	8.36E-07	5.42E-20	9.26E-20	5.14E-25	5.63E-21	4.95E-17	1.38E-21	1.84E-21
	Std function values	8.32E-19	3.37E-21	9.41E-18	1.55E-06	1.03E-19	2.07E-19	1.09E-24	1.25E-20	1.11E-16	2.07E-21	2.96E-21
	p-value	-	8.41E-01	6.90E-01	6.90E-01	7.94E-03	6.90E-01	1.00E+00	4.21E-01	1.00E+00	6.90E-01	8.41E-01
F7	Average function values	5.00E-03	6.22E-03	3.94E-03	4.28E-03	8.02E-03	7.84E-03	5.06E-03	5.41E-03	1.26E-02	1.03E-02	4.24E-03
	Std function values	2.30E-03	2.74E-03	1.46E-03	2.73E-03	7.73E-03	4.86E-03	1.91E-03	1.58E-03	1.40E-02	3.17E-03	1.49E-03
	p-value	-	8.41E-01	4.21E-01	4.21E-01	6.90E-01	8.41E-01	5.48E-01	1.00E+00	1.00E+00	3.10E-01	6.90E-01
F8	Average function values	-1.05E+04	-1.04E+04	-1.08E+04	-1.05E+04	-1.05E+04	-1.04E+04	-1.05E+04	-1.03E+04	-1.04E+04	-1.04E+04	-1.01E+04
	Std function values	2.17E+02	2.22E+02	5.27E+02	4.96E+02	6.45E+02	1.60E+02	2.79E+02	3.89E+02	4.66E+02	3.25E+02	2.01E+02
	p-value	-	5.48E-01	3.10E-01	3.10E-01	5.48E-01	1.00E+00	4.21E-01	6.90E-01	4.21E-01	8.41E-01	5.56E-02
F9	Average function values	3.52E-13	1.16E-10	5.97E-01	5.97E-01	3.98E-01	3.98E-01	1.99E-01	1.05E-12	4.21E-13	1.99E-01	1.99E-01
	Std function values	2.14E-13	2.57E-10	5.45E-01	8.90E-01	5.45E-01	5.45E-01	4.45E-01	1.38E-12	2.50E-13	4.45E-01	4.45E-01
	p-value	-	1.51E-01	1.03E-01	1.03E-01	7.94E-03	6.35E-02	1.67E-01	5.71E-01	4.13E-01	6.51E-01	2.06E-01
F10	Average function values	1.23E-05	1.16E-08	1.55E-09	5.78E-05	6.33E-13	6.01E-08	3.76E-13	2.61E-12	2.07E-13	8.50E-08	4.93E-08
	Std function values	2.74E-05	2.59E-08	3.47E-09	7.52E-05	6.67E-13	1.34E-07	4.91E-13	4.45E-12	1.63E-13	1.90E-07	9.19E-08
	p-value	-	5.48E-01	8.41E-01	8.41E-01	9.52E-02	9.37E-01	1.51E-01	2.38E-01	8.41E-01	5.48E-01	5.48E-01
F11	Average function values	2.02E-02	3.68E-02	2.02E-02	1.08E-02	4.60E-02	1.28E-02	2.17E-11	3.92E-02	1.72E-02	7.37E-03	1.47E-02
	Std function values	1.22E-02	2.99E-02	2.41E-02	2.41E-02	3.64E-02	2.46E-02	4.85E-11	4.37E-02	3.20E-02	1.28E-02	2.66E-02
	p-value	-	5.48E-01	8.41E-01	8.41E-01	1.51E-01	4.21E-01	1.51E-01	7.94E-03	1.00E+00	4.21E-01	4.21E-01

TABLE 4. ***(Continued.)*** **The comparison between DGO and different CDGOs using chaotic maps at intra and inter-groups action.**

F12	Average function values	2.10E-21	2.07E-02	4.23E-15	1.15E-09	1.96E-21	1.12E-18	3.82E-24	1.08E-17	2.07E-02	4.15E-02	5.50E-18
	Std function values	4.40E-21	4.64E-02	9.46E-15	1.36E-09	4.39E-21	2.51E-18	6.24E-24	2.40E-17	4.64E-02	5.68E-02	1.23E-17
	p-value	-	1.51E-01	4.21E-01	4.21E-01	7.94E-03	6.90E-01	8.41E-01	5.48E-01	1.00E+00	2.22E-01	1.00E+00
F13	Average function values	6.41E-24	3.83E-18	3.26E-25	7.03E-11	1.33E-26	3.43E-17	4.53E-23	1.56E-18	2.61E-18	1.15E-16	1.47E-19
	Std function values	8.12E-24	5.32E-18	6.25E-25	1.07E-10	2.53E-26	5.07E-17	9.85E-23	3.48E-18	5.83E-18	2.58E-16	3.26E-19
	p-value	-	5.48E-01	1.51E-01	1.51E-01	7.94E-03	9.52E-02	3.10E-01	8.41E-01	3.10E-01	9.52E-02	8.41E-01
F14	Average function values	2.95E+00	2.95E+00	3.35E+00	1.79E+00	9.98E-01	1.39E+00	2.19E+00	3.33E+00	1.79E+00	1.39E+00	2.95E+00
	Std function values	4.37E+00	4.37E+00	4.23E+00	1.09E+00	1.11E-16	8.87E-01	1.09E+00	5.22E+00	1.09E+00	8.87E-01	4.37E+00
	p-value	-	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	2.86E-01	7.22E-01	1.00E+00	1.00E+00

TABLE 5. **The comparison results between DGO and different CDGOs using chaotic maps.**

	random variable	Chebyshev	Circle	Gauss	Iterative	Logistic	Piecewise	Sine	Singer	Sinusoidal	Tent
Lyapunov exponent λ	-0.496	2.473206	2.813982	2.249289	1.831603	2.419504	2.120411	1.327458	2.640097	0.925197	2.706031
p-value	4.35e-11,	0.996592	1	1	0.999658	1	1	1	0.978982	1	0.999876
H	1	0	0	0	0	0	0	0	0	0	0

We used CCDGO, k-means and an EA-based algorithm-Cwolf to compare the performance. The parameter setting of DGO is the same as the Section III, k-means and Cwolf are set as suggested in [17]. We ran the experiments 50 times to obtain an average result of each algorithm. In order to fully investigate the performance of CDGO on real world data clustering problems, we employed the evaluation metrics: accuracy, precision, recall and F-measure. Three multimedia datasets are used in our comparison. They all have the property of high-dimensional space, noise and polluted data.

Parkinson Speech dataset with multiple types of sound recordings data set is employed in this experiment. This data set is a well-known multimedia data set. It records the voices of 40 people in the real world. Types of voice include vowels, numbers, words and sentences. The data set has 28 attributes and 1040 instances, which is very hard to cluster. Numerous experts choose this dataset to test clustering performance of methods on this data set.

Web Page dataset is multimedia content, were separated from graphic areas of the web page. There are 5473 instances and 10 attributes, which come from 54 different documents. One of the dataset's application is to cluster the data objects to separate groups of text, graphics and pictures depending on the attributes values.

Libra Movement dataset is extracted 45 frames from videos to cluster the hand movement. It contains 360 instances and 91 attributes.

Fig. 7 shows the result of comparison on three datasets, with the numerical result listed in the Table 7. We can clearly find that the CCDGO outperforms the other two methods. For the Parkinson Speech dataset, although the accuracy of CCDGO is lower than k-means, the F-measure is significantly better than K-means, which indicates the overall performance of CCDGO is the best. Precision of CCDGO is significantly better than both k-means and Cwolf. For Web Page dataset, CCDGO obtains the best result on all evaluation metrics. The recall value of CCDGO is significantly higher than k-means, which means CCDGO finds much more correct clustering objects. For Libra Movement dataset, CCDGOC exhibits its global optimum finding ability in the task of clustering. In general, the results demonstrate that our proposed CCDGO is promising to solve multimedia data clustering problems in comparisons to k-means and Cwolf.

F. DISCUSSION

As mentioned in Section I, there are many methods to enhance search capability. Although many of them show a great contribution to improving optimization algorithm, but chaotic optimization is the relatively suitable method to hybridize.

Technically, a metaheuristic algorithm has three components in the stochastic search process. The first is global exploration. It explores the entire search space to scout the vicinity of the promising solution. The second is local exploitation, which is the convergence towards the most promising solutions in the area obtained from the global exploration phase. The third is mutation or crossover. The current solution is transformed into a new solution and varied

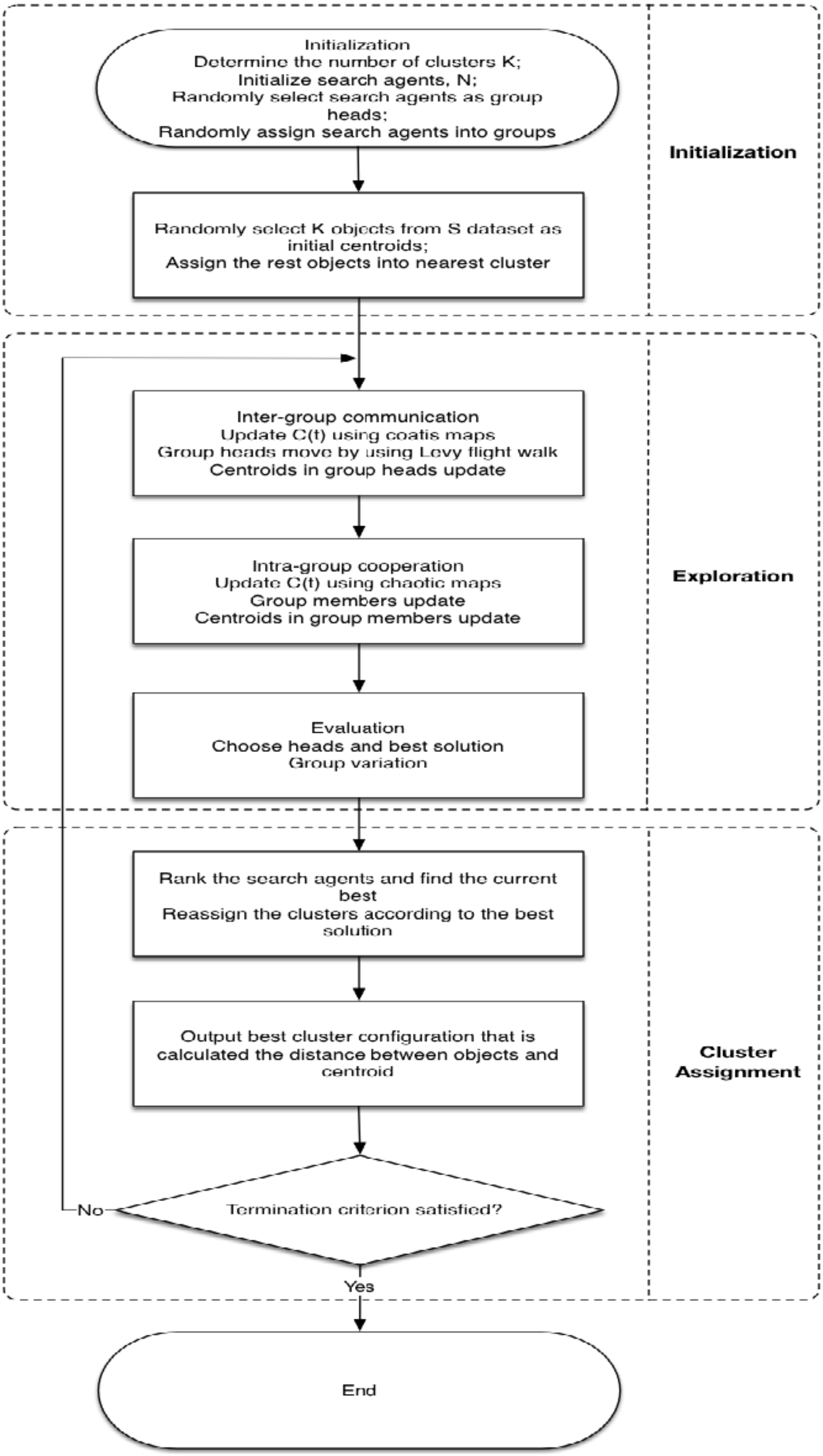

FIGURE 6. Flowchart of CCDG.

slightly. Many approaches are used to improve these components, such as sub-swarm and hybrid populations. However, balancing global exploration and local exploitation is still difficult, better global exploration capability is usually accompanied by worse local exploitation, and vice versa. DGO lends itself strongly to exploration and exploitation.

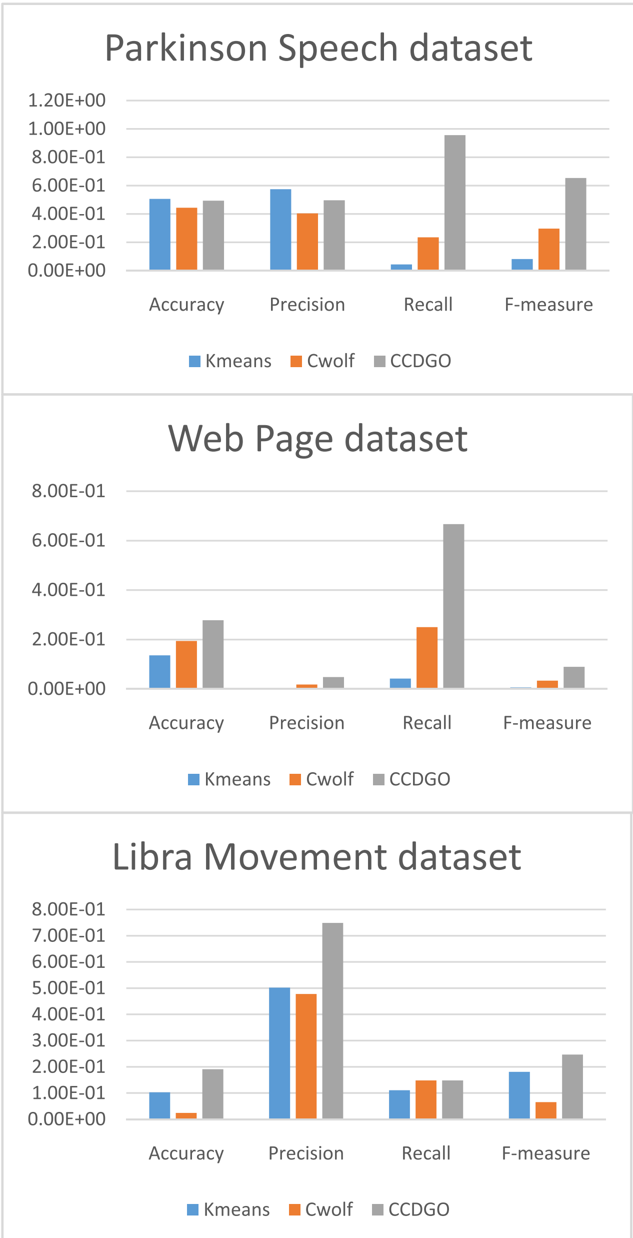

FIGURE 7. Comparison of CCDGO,Cwolf and K-means.

The search strategy employed in CDGO is mainly based on random walks, which includes **Lévy** flight in innergroup cooperation and uniform distribution in intergroup communication. These two randomness have distinct advantages and disadvantages. For **Lévy** flight, the step size is obeyed the heavy-tailed distance distribution, it expanses the scope of search, but the new solution tends to accumulate along the previous movement. For uniform distribution, it has the simple structure and easy to use, but it may produce duplicate solution. Introducing chaos is the most suitable approach to solve those problems. It has the property of the non-repetition, ergodicity and dynamic. The dynamic property ensures the solutions produced by algorithms can be diverse to search all different landscapes of search space, and the ergodicity and non-repetition enhance the speed of search. Therefore, it is natural to employ the chaos into DGO to enhance the search ability of DGO.

TABLE 6. The comparison results between DGO and different CDGOs using chaotic maps.

Function	CPSO	CKH	WSA	PSO	CDGO2
F1	2.60E-02	2.20E-03	6.53E-01	1.71E-05	7.43E-16
F2	3.05E+01	2.92E+01	8.55E-01	2.59E+01	5.85E-10
F3	2.85E-01	6.38E+00	7.45E-01	2.06E+00	6.03E-12
F4	2.60E-03	1.30E-03	8.82E-01	1.01E-06	3.80E-02
F5	2.43E+01	1.91E+02	8.75E-01	3.98E+01	9.76E+00
F6	6.98E-01	4.41E+00	7.34E-01	2.70E-03	6.48E-20
F7	7.12E-01	1.43E+01	9.28E-01	1.24E+00	7.32E-03
F8	9.86E-01	9.22E+00	8.29E-01	1.31E-02	-1.02E+04
F9	4.15E-08	1.10E-07	6.82E-12	7.70E-03	3.98E-01
F10	1.26E-07	9.14E-08	2.11E-12	4.53E-02	3.18E-12
F11	1.12E-06	6.79E-08	8.17E-12	1.65E-01	3.43E-02
F12	2.74E-04	1.62E-04	7.45E-08	4.84E-07	4.07E-22
F13	4.44E-08	8.22E-08	9.12E-08	8.74E-11	5.85E-17
F14	4.80E-02	1.50E-02	1.66E-07	1.65E+00	1.79E+00

TABLE 7. The comparison of CCDGO, Cwolf and K-means.

	Function	Accuracy	Precision	Recall	*F*measure
Parkinson Speech dataset	Kmeans	5.06E-01	5.75E-01	4.42E-02	8.21E-02
	Cwolf	4.44E-01	4.04E-01	2.35E-01	2.97E-01
	CCDGO	4.94E-01	4.97E-01	9.56E-01	6.54E-01
Web Page dataset	Kmeans	1.36E-01	3.50E-03	4.17E-02	6.40E-03
	Cwolf	1.94E-01	1.79E-02	2.50E-01	3.33E-02
	CCDGO	2.78E-01	4.79E-02	6.67E-01	8.94E-02
Libra Movement dataset	Kmeans	1.03E-01	5.02E-01	1.11E-01	1.81E-01
	Cwolf	2.45E-02	4.78E-01	1.48E-01	6.54E-02
	CCDGO	1.91E-01	7.49E-01	1.48E-01	2.47E-01

The convergence properties of DGO are strongly related to its stochastic nature and DGO uses a random sequence for its parameters during a run. Generating random sequences with a long period and good uniformity are very important for easy simulating complex phenomena, sampling, numerical analysis, decision making and especially in heuristic optimization. Its quality determines the reduction of storage and computation time to achieve a desired accuracy [25]. Chaos has properties of randomness, non-repetition and ergodicity, it is perfectly matched the stochastic feature of meta heuristic optimization algorithms. Chaotic optimization not only accelerates the speed of algorithm, but can also enhance the variety of movement pattern. From the experiments conducted in Section III on the benchmarks, it is demonstrated that CDGO performs better than the classical DGO. Moreover, it is also interesting to compare the capability between CCDGO and the EA-based clustering algorithms on multimedia clustering problems. Multimedia data are loose, unstructured and dynamical, which are the tough problems to solve are. The comparison between CCDGO and other algorithms shows that CCSGO could generate better results on real world data clustering problems.

Although benchmark functions is the most wildly used way to evaluate the performance of the heuristic algorithms, it still not a perfect approach. Most heuristic optimization algorithms have randomness and parameters. And therefore, the different tuning parameters may result in a significant difference in their performance. In our work, we used same experiment environment to obtain the unbiased results. However, our benchmark evaluation may generate different results if we change the population size or termination condition. In spite of these caveats, the benchmark results show that CDGO is promising optimization tool.

IV. CONCLUSION

Chaos has been widely observed in various applications. In this paper, we used chaos theory combined with the latest algorithm DGO to perform two activities: improvement of intra-group cooperation and inter-group communication. The first advantage of CDGO is using fewer chaotic maps to enhance the searching capability. Secondly, chaotic optimization performs search at higher speed compared to the stochastic searches rely on probability [20]. Moreover, CDGO is a simple structure and easy to implement.

In order to investigate the performance of CDGO in this study, fourteen benchmark functions are utilized into comparison. The experimental results showed that our proposed algorithm outperforms the other algorithms in different testing conditions. Furthermore, the CDGO2 which utilizes the circle map enhances the performance of the DGO algorithm remarkably.

The main contributions of this paper are as follows. (1) Because the chaotic maps used into intergroup phase, the convergence in local exploitation is accelerated. (2) The intergroup communication enhances the global exploration ability by using chaotic sequences. It provides variety in the population. (3) Local exploitation and global exploration are both improved in the CDGO algorithm, so the balance between global exploration and local exploitation is coordinated by itself in a self-adaptive way. (4) We use the sequence generator at a variable (bit) level, rather than on a vector (individual) level. (5) The algorithm is easy to hybrid into the other data mining tools.

From the comparison among CCDGO and the other two well-known algorithms on multimedia data clustering problems, it can be shown that the applicability of CCDGO for multimedia data clustering problem solving is feasible. For future studies, it may be worthwhile to employ CDGO algorithms for solving real-world engineering problems. In addition, other chaotic maps are also worth applying onto DGO [26]–[28].

REFERENCES

[1] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Mach. Learn.*, vol. 3, nos. 2–3, pp. 95–99, 1988.
[2] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evol. Comput.*, vol. 9, no. 2, pp. 159–195, Jun. 2001.
[3] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[4] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*. New York, NY, USA: Springer, 2011, pp. 760–766.
[5] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, no. 6791, pp. 39–42, 2000.
[6] R. Tang, S. Fong, X.-S. Yang, and S. Deb, "Integrating nature-inspired optimization algorithms to K-means clustering," in *Proc. 7th Int. Conf. Digit. Inf. Manage. (ICDIM)*, Aug. 2012, pp. 116–123.
[7] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biol. Inspired Comput. (NaBIC)*, Dec. 2009, pp. 210–214.
[8] R. Gregory, R. Paton, J. Saunders, and Q. H. Wu, "Parallelising a model of bacterial interaction and evolution," *BioSystems*, vol. 76, nos. 1–3, pp. 121–131, 2004.
[9] S. Saremi, S. Mirjalili, and A. Lewis, "Biogeography-based optimisation with chaos," *Neural Comput. Appl.*, vol. 25, no. 5, pp. 1077–1097, 2014.
[10] R. Tang, S. Fong, S. Deb, and R. Wong, "Dynamic group search algorithm," in *Proc. 4th Int. Symp. Comput. Bus. Intell. (ISCBI)*, Olten, Switzerland, Sep. 2016, pp. 159–164.
[11] Z. Assarzadeh and A. R. Naghsh-Nilchi, "Chaotic particle swarm optimization with mutation for classification," *J. Med. Signals Sensors*, vol. 5, no. 1, pp. 12–20, 2015.
[12] 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, 2013.
[13] X. Han and X. Chang, "An intelligent noise reduction method for chaotic signals based on genetic algorithms and lifting wavelet transforms," *Inf. Sci.*, vol. 218, pp. 103–118, Jan. 2013.
[14] G.-G. Wang, L. Guo, A. H. Gandomi, G.-S. Hao, and H. Wang, "Chaotic Krill Herd algorithm," *Inf. Sci.*, vol. 274, pp. 17–34, Aug. 2014.
[15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[16] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.
[17] X. Wu, B. Zhu, Y. Hu, and Y. Ran, "A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps," *IEEE Access*, vol. 5, pp. 6429–6436, 2017.
[18] T. Rui, S. Fong, X.-S. Yang, and S. Deb, "Nature-inspired clustering algorithms for Web intelligence data," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol. (WI-IAT)*, Dec. 2012, pp. 147–153.
[19] L. D. S. Coelho and V. C. Mariani, "Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization," *Expert Syst. Appl.*, vol. 34, no. 3, pp. 1905–1913, 2008.
[20] A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, "Firefly algorithm with chaos," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 1, pp. 89–98, 2013.
[21] A. Allison and D. Abbott, "Control systems with stochastic feedback," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 11, no. 3, pp. 715–724, 2001.
[22] M. B. Calva-Yáñez, P. A. Niño-Suarez, E. A. Portilla-Flores, J. A. Aponte-Rodríguez, and E. Santiago-Valentn, "Reconfigurable mechanical system design for tracking an ankle trajectory using an evolutionary optimization algorithm," *IEEE Access*, vol. 5, pp. 5480–5493, 2017.
[23] M. E. Lotfy, T. Senjyu, M. A. Farahat, A. F. Abdel-Gawad, and A. Yona, "Enhancement of a small power system performance using multi-objective optimization," *IEEE Access*, vol. 5, pp. 6212–6224, 2017.
[24] D. Yang, Z. Liu, and J. Zhou, "Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 4, pp. 1229–1246, 2014.
[25] B. Alatas, E. Akin, and A. B. Ozer, "Chaos embedded particle swarm optimization algorithms," *Chaos, Solitons Fract.*, vol. 40, no. 4, pp. 1715–1734, 2009.

[26] 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.

[27] M. S. Tavazoei and M. Haeri, "Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms," *Appl. Math. Comput.*, vol. 187, no. 2, pp. 1076–1085, 2007.

[28] M. Ghasemi, S. Ghavidel, J. Aghaei, M. Gitizadeh, and H. Falah, "Application of chaos-based chaotic invasive weed optimization techniques for environmental OPF problems in the power system," *Chaos, Solitons Fract.*, vol. 69, pp. 271–284, Dec. 2014.

RUI TANG received the B.S. degree in software engineering from Nanchang University, Jiangxi, China, in 2010, and the M.S. degree in software engineering from the University of Macau, Taipa, Macau, in 2013, where he is currently pursuing the Ph.D. degree with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau. He received the full scholarship from the University of Macau to support his Ph.D. study. He has authored or coauthored more than a dozen international conference and peer-reviewed journal papers in the area of computer science and data mining. His research interests are data mining, swarm intelligence algorithms and its optimization, data pre-processing, including feature selection and imbalanced class dataset rebalancing, and text mining: sentiment analysis.

SIMON FONG received the B.Eng. degree (Hons.) in computer systems and the Ph.D. degree in computer science from La Trobe University, Australia, in 1993 and 1998, respectively. He was an Assistant Professor with the School of Computer Engineering, Nanyang Technological University, Singapore. Prior to his academic career, he took up various managerial and technical posts, such as a Systems Engineer, an IT Consultant, and an e-Commerce Director in Melbourne, Hong Kong, and Singapore. Some companies that he worked before include Hong Kong Telecom, Singapore Network Services, AES Pro-Data and United Oversea Bank, Singapore. He is currently an Associate Professor with the Computer and Information Science Department, University of Macau. He is also one of the Founding Member of the Data Analytics and Collaborative Computing Research Group, Faculty of Science and Technology. He has published over 200 international conference and peer-reviewed journal papers, mostly in the area of e-commerce technology, business intelligence and data mining.

RAYMOND K. WONG received the B.Sc. degree from The Australian National University, and the M.Phil. and Ph.D. degrees from The Hong Kong University of Science and Technology. From 2005 to 2011, he founded and led the Database Research Group, NICTA (the largest ICT organization in Australia). He was the CTO/Lead Developer of several technology startup companies. He is currently an Associate Professor with the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia. He has published over 150 research publications and two patents in these areas, and supervised 13 Ph.D. students to completion. His research expertise lies in database systems and data mining.

KELVIN K. L. WONG received the B.Eng. degree (Hons.) in mechanical and production engineering from Nanyang Technological University in 2001, the M.A.I.T. degree in applied information technology from The University of Sydney in 2003, and the Ph.D. degree in electrical and electronic engineering from The University of Adelaide in 2009. From 2006 to 2009, he was doing research work on medical imaging and cardiac flow analysis. He has over ten years of medical imaging, computational hemodynamics, and simulation modeling experience. Since 2009, he has been involved in computational fluid and solid dynamics. In 2013, he was involved in the discrete element method with applications in biological systems with The University of Western Australia.

He was the originator of the spatial game moment concept, in which multi-objective multi-constraint combinatorial optimization problems can be treated as decision-making problems in the game theoretical sense, and solved with high efficiency. In addition, he is the first author of the book *Methods in Research and Development of Biomedical Devices* (World Scientific, 2013), and a co-author of a second book *Computational Hemodynamics—Theory, Modelling and Applications* (Springer, 2015). His publications now span a diverse range of topics in the medical science and engineering field. He has served as associate editors and guest editors for journals in the area of biomedical engineering.