IEICE TRANS. INF. & SYST., VOL.E100–D, NO.4 APRIL 2017

PAPER

Multiple Chaos Embedded Gravitational Search Algorithm

Zhenyu SONG[†], ***Nonmember***, **Shangce GAO**[†a)], ***Member***, **Yang YU**[†], **Jian SUN**[†,††], ***and*** **Yuki TODO**[†††], ***Nonmembers***

SUMMARY This paper proposes a novel multiple chaos embedded gravitational search algorithm (MCGSA) that simultaneously utilizes multiple different chaotic maps with a manner of local search. The embedded chaotic local search can exploit a small region to refine solutions obtained by the canonical gravitational search algorithm (GSA) due to its inherent local exploitation ability. Meanwhile it also has a chance to explore a huge search space by taking advantages of the ergodicity of chaos. To fully utilize the dynamic properties of chaos, we propose three kinds of embedding strategies. The multiple chaotic maps are randomly, parallelly, or memory-selectively incorporated into GSA, respectively. To evaluate the effectiveness and efficiency of the proposed MCGSA, we compare it with GSA and twelve variants of chaotic GSA which use only a certain chaotic map on a set of 48 benchmark optimization functions. Experimental results show that MCGSA performs better than its competitors in terms of convergence speed and solution accuracy. In addition, statistical analysis based on Friedman test indicates that the parallelly embedding strategy is the most effective for improving the performance of GSA.

key words: *chaos, gravitational search algorithm, local search, optimization, meta-heuristics*

1. Introduction

Meta-heuristics have been successfully and widely used for solving various optimization problems in the past decades [1]. A considerable number of meta-heuristics have been proposed based on metaphors of natural evolution, swarm mechanisms or man-made processes [2]. These meta-heuristics include evolutionary computation [3], particle swarm optimizations [4], ant colony algorithms [5], artificial immune systems [6], etc. Among them, gravitational search algorithm (GSA) which is inspired from the Newton's law of gravity and motion [7] has demonstrated to be a powerful optimization tool when applied to function optimization problems and many real-world problems [7]–[10].

Like other nature-inspired meta-heuristic algorithms, GSA is a population-based adaptive search technique. In GSA, a population of candidate solutions are modeled as a swarm of objects. At each iteration, the objects update their position by moving stochastically towards regions previously visited by the other objects. The object with heavier mass has a larger effective attraction radius and hence a greater intensity of attraction. By lapse of time, the objects tend to move towards the heaviest object. In comparison with other well-known optimization algorithms, such as the particle swarm optimization, GSA has been confirmed higher performance in searching ability [7]. However, GSA still has some inherent disadvantages, such as it usually sticks on local optimal solutions, which indicates that it is unable to improve the solutions' quality in the latter search phases [11], [12].

Many attempts have been made to alleviate the inherent local minima trapping problem and further improve the search performance of GSA [12]. Li and Zhou [13] modified the velocity updating rule by utilizing the memory and social information of agents, aiming to accelerate GSA's convergence speed. An opposition-based learning rule was proposed for population initialization and generation jumping in GSA [14]. A niching GSA was proposed by dividing the main swarm of masses into several small sub-swarms to maintain the diversity of population [15], thus improving the performance of GSA for multimodel optimization problems.

Considerable effort has been devoted to incorporating chaos into meta-heuristics in recent years. Chaos is a universal phenomenon of nonlinear dynamic systems and it is apparently an irregular motion, seemingly unpredictable random behavior exhibited by a deterministic nonlinear system under deterministic conditions. Due to the ergodicity and dynamic properties of chaos, chaotic maps can help meta-heuristic optimization algorithms to enhance the diversity among individuals and avoid premature convergence. In the literature, chaotic maps have been incorporated into evolutionary algorithms [16], particle swarm optimization [17], biogeography-based optimisation [18], water cycle algorithm [19], fruit fly optimization [20], Krill Herd algorithm [21], bat algorithm [22], differential evolution algorithm [23], harmony search algorithm [24], firefly algorithm [25], ant swarm optimization [26], imperialist competitive algorithm [27], and others.

In our previous work [28], we proposed two kinds of chaos-based GSAs. One used chaotic sequences to substitute random numbers for different parameters, and the other used chaotic variables to perform a chaotic local search. Both embedding strategies of chaos were found to be benefit for improving GSA's search ability, and the latter seemed to be more efficient. The work [28] has been extended by con-

Manuscript received January 1, 2017.
Manuscript publicized January 13, 2017.
[†]The authors are with Faculty of Engineering, University of Toyama, Toyama-shi, 930–8555 Japan.
[††]The author is with College of Computer Science and Technology, Taizhou University, Taizhou, 225300, China.
[†††]The author is with School of Electrical and Computer Engineering, Kanazawa University, Kanazawa-shi, 920–1192 Japan.
a) E-mail: gaosc@eng.u-toyama.ac.jp
DOI: 10.1587/transinf.2016EDP7512

sidering five different chaotic maps. Preliminary experimental results in [29] empirically showed that all introduced five chaotic maps generally exhibited effectiveness of improving the performance of GSA. Nevertheless, there is no specific chaotic maps can enable GSA to achieve the best solution for all optimization problems, suggesting that the performance of chaotic GSAs are related not only to the search capacity of the algorithm, but also to the landscape of the solved problems.

In this paper, we investigate the capability of chaotic local search using different chaotic maps and different incorporation strategies for improving the search performance of GSA. The motivation of this study comes from the following aspects. First, the effectiveness of the incorporation of chaos into GSA needs to be verified via extensive experiments. Second, as a number of chaotic maps is available, it is required to find out which one is the most appropriate for GSA. Third, a well-established embedding strategy which can fully utilize the search dynamics of chaos needs to be designed. Based on these considerations, we propose a multiple chaos embedded gravitational search algorithm (MCGSA) in this paper.

In all prior chaotic meta-heuristics [16]–[29], only a single certain chaotic map is embedded into the meta-heuristic algorithm to perform the chaotic search. Few research studies the integration of multiple chaotic systems which simultaneously perform the search. The prior chaotic GSA in [28] only utilized the well-known Logistic maps to realize the chaotic search, and the extended one [29] compared the performance difference during five chaotic maps. Obviously, the search capacity of such single chaos embedded GSA is limited. Multiple chaos can be expected to provide more dynamic properties for alleviating the local problem trapping problem of GSA. To realize these, we first construct twelve variants of single chaos embedded GSA using twelve different chaotic maps. Then, three novel multiple chaos embedded GSAs (MCGSA) are proposed, i.e., the twelve chaotic maps are (1) randomly, (2) parallelly, and (3) memory-selectively incorporated into GSA. The resultant chaotic GSAs are called CGSA-R, CGSA-P, and CGSA-M, respectively. Extensive experiments are conducted based on 48 widely used benchmark numerical optimization functions. Experimental results and statistical analysis verified that MCGSA can perform better than the traditional GSA and those single chaos embedded GSAs.

The remainder of this paper is organized as follows. Section 2 gives a brief description of the traditional GSA. Section 3 summarizes the twelve chaotic maps used in this study. In Sect. 4, we first introduce the single chaos embedded GSA. Then three kinds of MCGSA are presented in details. Section 5 provides experimental results. Section 6 concludes the paper.

2. Brief Description of Traditional GSA

GSA is a population based meta-heuristic algorithm inspired by the law of gravity among objects. Each agent in the population of GSA is considered as objects and its performance is measured by its mass. The position of agent corresponds to a solution of the optimization problem needed to be solved. Moving the position of agent can result in an improvement of the solution's quality.

Formally, every agent $X_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^D)$, $(i = 1, 2, \ldots, N)$ attracts each other by gravitational forces in a D-dimensional search space, where x_i^d represents the position of i-th agent in the d-th dimension. The corresponding velocity of agent X_i is expressed by $V_i = (v_i^1, \ldots, v_i^d, \ldots, v_i^D)$. The mass of each agent in iteration t, denoted by $M_i(t)$, is calculated via the map of its fitness as follows:

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

where $\text{fit}(X_i(t))$ represents the fitness of agent X_i by calculating the objective function. For a minimization problem, $\text{best}(t)$ and $\text{worst}(t)$ are defined as

$$\text{best}(t) = \min_{j=1,2,\ldots,N} \text{fit}(X_j(t)) \tag{2}$$

$$\text{worst}(t) = \max_{j=1,2,\ldots,N} \text{fit}(X_j(t)) \tag{3}$$

The force acting on the i-th agent from the j-th agent is defined as:

$$F_{ij}^d(t) = G(t)\frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon}(x_j^d(t) - x_i^d(t)) \tag{4}$$

where $R_{ij}(t) = \|x_i(t), x_j(t)\|_2$ is the Euclidean distance between two agents, and ε is a small constant, preventing the denominator in Eq. (4) from being zero. In addition, $G(t)$ is the gravitational constant at time t, defined by

$$G(t) = G_0 \exp\left(-\alpha \frac{t}{t_{max}}\right) \tag{5}$$

where G_0 is the initial value, α is a shrinking constant, t_{max} is the maximum number of iterations. For the i-th agent, the overall force that acts on it is a randomly weighted sum of the forces exerted from the surrounding agents.

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i} \text{rand}_j F_{ij}^d(t) \tag{6}$$

where $Kbest$ is the set of first K agents with the best fitness and biggest mass, rand_j is a random number uniformly generated in the interval $[0, 1]$. Furthermore,

$$K = \left\lfloor \left(\beta + \left(1 - \frac{t}{t_{max}}\right)(1 - \beta)\right) N \right\rfloor \tag{7}$$

It is clear that K is initially set to N and is decreased linearly, which is controlled by a constant β. The operation $\lfloor \cdot \rfloor$ is the floor function. Based on the law of motion, the acceleration of the i-th agent is calculated by:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \tag{8}$$

Then, the next velocity of an agent is considered as a fraction of its current velocity added to its acceleration. Therefore, its position and its velocity could be updated as follows:

$$v_i^d(t+1) = \text{rand}_i v_i^d(t) + a_i^d(t) \tag{9}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{10}$$

where rand_i is a random variable in $[0, 1]$. It should be noted that both rand_i and rand_j are uniformly generated random numbers, and they generally differ from each other. In fact they are an attempt of giving randomized characteristics to the search.

3. Chaotic Maps

One dimensional non-invertible maps are the simplest systems with the capability of generating chaotic motion. In this study, twelve well-known one-dimensional chaotic maps widely used in related researches [16]–[27] are considered.

(1) Logistic map: this classic logistic map appears in nonlinear dynamics of biological population evidencing chaotic behavior, and can be written as in the following.

$$z_{k+1} = \mu z_k(1 - z_k) \tag{11}$$

where z_k is the kth chaotic number. Obviously, $z_k \in (0, 1)$ under the conditions that the initial $z_0 \in (0, 1)$ and that $z_0 \notin \{0.0, 0.025, 0.5, 0.75, 1.0\}$. In our experiment, we set $\mu = 4$ and the initial number $z_0 = 0.152$.

(2) Piecewise linear chaotic map (PWLCM): it has been known as ergodic and has uniform invariant density function on their definition intervals. The simplest PWLCM is governed by the following equation.

$$z_{k+1} = \begin{cases} z_k/p, & z_k \in (0, p) \\ (1 - z_k)(1 - p), & z_k \in [p, 1) \end{cases} \tag{12}$$

In the experiment, p is set to be 0.7 and $z_0 = 0.002$.

(3) Singer map: it is a one-dimensional system as given in the following.

$$z_{k+1} = \mu(7.86z_k - 23.31z_k^2 + 28.75z_k^3 - 13.302875z_k^4) \tag{13}$$

Singer map exhibits chaotic behaviors when the parameter μ is set as a value between 0.9 and 1.08. In this study, we set $\mu = 1.073$ and $z_0 = 0.152$.

(4) Sine map: it belongs to a unimodal map which is similar to the Logistic map, can it is written as the following equation.

$$z_{k+1} = \frac{a}{4}\sin(\pi z_k) \tag{14}$$

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

(5) Sinusoidal map: this iterator can be defined as

$$z_{k+1} = a z_k^2 \sin(\pi z_k) \tag{15}$$

where $a = 2.3$ and we set the initial number of this chaotic system as $z_0 = 0.74$.

(6) Tent map: this map is similar to the well-known Logistic map, and displays some specific chaotic effects. These two maps can be converted to each other, and there is a relationship of topological conjugacy between them. Tent map can be defined by the following equation.

$$z_{k+1} = \begin{cases} z_k/\beta, & 0 < z_k \le \beta \\ (1 - z_k)/(1 - \beta), & \beta < z_k \le 1 \end{cases} \tag{16}$$

We set $\beta = 0.4$ and $z_0 = 0.152$.

(7) Bernoulli shift map: this map belongs to the class of piecewise linear maps similar to the Tent map. It is formulated as follows

$$z_{k+1} = \begin{cases} z_k/(1 - \lambda), & 0 < z_k \le 1 - \lambda \\ (z_k - 1 + \lambda)/\lambda, & 1 - \lambda < z_k < 1 \end{cases} \tag{17}$$

We set $\lambda = 0.4$ and $z_0 = 0.152$.

(8) Chebyshev map: it is a common chaotic map, and has wide application in the neural network, digital communication and security. Its equation is expressed as

$$z_{k+1} = \cos(\phi \cos^{-1} z_k) \tag{18}$$

where the parameter ϕ is set to be 5 and the initial chaotic number $z_0 = 0.152$.

(9) Circle map: this map is a simplified model for both driven mechanical rotors and the phase locked loop in electronics. It is a one-dimensional map which maps a circle onto itself. It is represented by the following equation.

$$z_{k+1} = z_k + a - \frac{b}{2\pi}\sin(2\pi z_k) \mod (1) \tag{19}$$

For $a = 0.5$ and $b = 2.2$, it can generate chaotic sequence in $(0, 1)$. Also, we set $z_0 = 0.152$ in the experiment.

(10) Cubic map: it is one of the most commonly used maps in generating chaotic sequence in various applications like cryptography. It can be formally defined by

$$z_{k+1} = \rho z_k(1 - z_k^2) \tag{20}$$

We set $\rho = 2.59$ and $z_0 = 0.242$.

(11) Gaussian map: it is represented using the following equation

$$z_{k+1} = \begin{cases} 0, & z_k = 0 \\ (\mu/z_k) \mod (1) & z_k \ne 0 \end{cases} \tag{21}$$

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

(12) Iterative chaotic map with infinite collapses (ICMIC): this map has infinite fixed points, and can be defined using

$$z_{k+1} = \sin(a/z_k) \tag{22}$$

where $a \in (0, +\infty)$ is an adjustable parameter, and we set $a = 70$ in our experiment. It is clear that ICMIC generates chaotic sequence $z \in [-1, 0) \cup (0, 1]$.

Algorithm 1:

```
for all agent i (i = 1, 2, ..., N) do
    initialize position X_i randomly in search space [L, U]
end-for
while termination criteria not satisfied do
    for all agent i do
        compute overall force F_i^d(t) according to Eqs. (1)–(7)
        compute acceleration a_i^d(t) according to Eq. (8)
        update velocity according to Eq. (9)
        update position according to Eq. (10)
    end-for
    find out the global best agent X_g
    implement the chaotic local search approach
    decrease the chaotic search radius
end-while
```

4. Chaotic Gravitational Search Algorithm (CGSA)

Chaos is a kind of a characteristic of nonlinear dynamic system which exhibits bounded dynamic unstable, pseudo random, ergodic, non-period behavior depended on initial value and control parameters [30]. Due to its ergodicity and randomicity, a chaotic system changes randomly, but eventually goes through every state if the time duration is long enough. This characteristic of chaotic systems can be utilized to build up a search operator for optimizing objective functions. Nevertheless, chaos optimization works well in a small search space but generates unacceptable optimization time in a large search space [31]. Therefore, chaotic search is often incorporated into other global optimizers such as evolutionary algorithms to enhance their search ability [16]–[27].

Compared with the methodology which uses chaotic sequences to substitute random values of the controlling parameters in GSA, chaotic local search has been demonstrated to be more effective for improving the performance of GSA [28]. As a matter of fact, chaotic local search is often adopted in related researches [17]–[27]. Thus, we employ the chaotic local search in this study.

The framework of CGSA is illustrated in Algorithm 1 and each variant of CGSA differs from each other by specifying the chaotic local search procedure.

4.1 Single Chaos Embedded CGSA

The chaotic local search that utilizes only a single chaotic map is defined as follows.

$$X_{g'}(t) = X_g(t) + r(t)(U - L)(z(t) - 0.5) \tag{23}$$

where $X_g(t)$ denotes the position of the current global best agent in the population at the t-th iteration number. $X_{g'}(t)$ is indicated as the new agent generated by the chaotic local search. U and L are the upper bound and lower bound of the search space, respectively. $z(t)$ is a chaotic variable generated from one of the considered chaotic maps. $r(t) \in (0, 1)$ is a chaotic search radius which is used to control the exploitation range of the search. It is worth pointing out that Eq. (23) actually denotes a batch local search manner. In a generation, the same chaotic variable $z(t)$ is used to update all components of the vector X_g (i.e., for all D dimensions).

Without the loss of generality, we suppose the optimization problem is a minimization one. After the local search is performed, an agent updating procedure is carried out according to the following equations.

$$X_g(t+1) = \begin{cases} X_{g'}(t) & \text{If } \ \mathrm{fit}(X_{g'}(t)) \le \mathrm{fit}(X_g(t)) \\ X_g(t) & \text{Otherwise} \end{cases} \tag{24}$$

$$X_i(t+1) = X_i(t) \ \text{ For } \ i = 1, 2, \ldots, N \ \text{ And } \ i \ne g \tag{25}$$

The newly generated solution $X_{g'}$ will replace the current global best agent if the fitness is improved, while the others survive to enter into the next iteration.

Regarding the single chaos embedded CGSA, some remarks are given in the following.

1. The local search is performed on the global best agent, not only aiming to improve the search performance of GSA, but also being able to save computational time when compared to the scheme that applies the local search to all agents;
2. Once the acquired values of $X_{g'}$ in Eq. (23) locate out of the search bound, these values will be reset to the closest boundary value; and
3. Considering the fact that chaotic search is efficient in small range, a shrinking scheme is used to narrow the search neighborhood by lapse of iteration using $r(t+1) = 0.988 \times r(t)$.

The variants of single chaos embedded CGSA using the chaotic map in Eqs. (11)–(22) are called CGSA-1–CGSA-12, respectively.

4.2 Multiple Chaos Embedded CGSA

Different chaotic maps exhibit different and distinct dynamic properties [32], [33]. Multiple chaos are supposed to provide more opportunities for a meta-heuristic to help it jump out of the local minima via the ergodicity and randomicity of chaos. The method of incorporating multiple chaos into meta-heuristics remains challenging and fascinating. In the prior researches no sophisticated scheme has been proposed. Thus, we innovatively propose three novel multiple chaos embedding schemes in this paper. The twelve chaotic maps in Eqs. (11)–(22) are (1) randomly, (2) parallelly, and (3) memory-selectively incorporated into GSA, respectively.

4.2.1 CGSA-R

The chaotic local search that randomly makes use of multiple chaos is defined in the following.

$$X_{g'}(t) = X_g(t) + r(t)(U - L)(z^j(t) - 0.5) \tag{26}$$

where $z^j(t)$ is a chaotic variable generated from the j-th chaotic map, and j is an uniformly distributed number generated from the set $\{1, 2, \ldots, 12\}$. In each iteration, only a single selected chaotic map is used. All twelve chaotic maps are used during the whole iterations and each one is implemented for approximately $t_{max}/12$ times. Thereafter, the updating procedure shown in Eqs. (24) and (25) is performed.

4.2.2 CGSA-P

The chaotic local search that parallelly uses multiple chaos can be defined as follows.

$$X^j_{g'}(t) = X_g(t) + r(t)(U - L)(z^j(t) - 0.5) \tag{27}$$

where $X^j_{g'}$, $j = 1, 2, \ldots, 12$ presents a candidate solution temporarily generated by the chaotic local search and it indicates that twelve candidate solutions are simultaneously generated using twelve different chaotic maps. Thereafter, the best one among the twelve candidate solutions is taken to compare with the current global best solution $X_g(t)$. If the fitness can be improved, then replace the original one; otherwise remain the same. The updating rule can be formally expressed as:

$$X_g(t+1) = \begin{cases} X^{j_{min}}_{g'}(t) & \text{If } \ \mathrm{fit}(X^{j_{min}}_{g'}(t)) \leq \mathrm{fit}(X_g(t)) \\ X_g(t) & \text{Otherwise} \end{cases} \tag{28}$$

$$j_{min} = j \in \{1, 2, \ldots, 12\} \quad \text{s.t.} \min_{j=1,2,\ldots,12} \mathrm{fit}(X^j_{g'}(t)) \tag{29}$$

4.2.3 CGSA-M

The basic idea of CGSA-M is derived from the adaptive trail vector generation strategy for differential evolution [34]. Similarly, we use this memory-based strategy for adaptively selecting different chaotic maps, and hereby named memory-selectively incorporation scheme. The implementation of the memory-selectively incorporation scheme can be described in the following.

In CGSA-M, with respect to each current global best agent X_g, one chaotic map is selected from twelve chaotic maps according to the probability learned from the success rate and failure rate in generating improved solutions within a certain number (i.e. LP) of previous iterations. The selected strategy is applied to the current global best agent X_g, to generate a new agent X'_g for comparing the fitness after utilizing the j-th chaotic map with X'_g to decide whether X_g would be replaced by X'_g, as shown in Eq. (23).

Initially, the probability of selecting each chaotic map is set to be 1/12, suggesting that all chaotic maps have the equal probability to be selected. With the lapse of iteration, the selection probabilities are updated according to the following rules.

$$p_{j,t} = \frac{S_{j,t}}{\sum_{j=1}^{12} S_{j,t}} \tag{30}$$

$$S_{j,t} = \frac{\sum_{g=t-LP}^{t-1} ns_{j,g}}{\sum_{g=t-LP}^{t-1} ns_{j,g} + \sum_{g=t-LP}^{t-1} nf_{j,g}} + \phi, \tag{31}$$
$$(j = 1, 2, \ldots, 12;\ t > LP)$$

where $p_{j,t}$ denotes the probability of selecting the j-th chaotic map at the t-th iteration. $ns_{j,t}$ indicates the number of new individuals generated by the j-th chaotic map and successfully entering the next iteration within the previous LP iterations with respect to generation t, and $nf_{j,t}$ denotes the number of these new individuals which failed to enter into the next iteration. $S_{j,t}$ represents the success rate, and $\phi = 0.01$ is set to avoid the null success rate. It is apparent that the larger the success rate for the j-th chaotic map, the higher the probability of applying it to generate new individual at the current iteration.

5. Experimental Results

To evaluate the performance of the proposed multiple chaos embedded gravitational search algorithms, i.e., CGSA-R, CGSA-P, CGSA-M, we make a comparison with the original gravitational search algorithm [7] and twelve single chaos embedded gravitational search algorithms using different chaotic maps (i.e., CGSA-1–CGSA-12). The experiment is conducted using Matlab on a personal PC.

In order to make a statistical analysis, all compared algorithms are implemented 30 times based on a total number of 48 benchmark functions. These benchmark functions are taken from [35], [36]. F1–F23 are the most commonly used benchmark numerical functions [35], where F1 and F5 are unimodal functions; F6 is a step function which has only one minimum and is discontinuous; F7 is a noisy quartic function; F8–F13 are multimodal functions with plenty of local minima and the number of the local minima in these functions increase exponentially with the dimension of the function; F14–F23 are low dimensional functions which only have a few local minima. These functions can successfully test the searching capacity of algorithms in terms of convergence speed and global exploration ability. In other words, unimodal functions are able to reflect the convergence speed of the algorithm in a direct manner, and multimodal ones are likely to estimate the algorithms' ability of escaping from local minima. Nevertheless, these traditional 23 benchmark functions suffer from two problems: (1) global minima lie at the center of the search range (usually at $\overrightarrow{0}^D$), which might be easily utilized as a prior knowledge; (2) local minima lie along the coordinate axes or no linkage among the variables exists [34]. Shifted or rotated functions proposed in CEC '05 [36] can solve these problems in traditional 23 benchmarks functions. F24–F48 are CEC '05 functions, where F24–F28 are shifted unimodal functions; F29–F37 are shifted multimodal functions; F38–F48 are rotated hybrid composition functions. Figure 1 illustrates the characteristics of the unimodal function F5 and the shifted rotated Griewank's function F30 respectively, in terms of the two-dimensional sketch and the contour.

The user-defined parameters in GSA and CGSAs are set as follows. The population size N is 50. The maximum iteration number t_{max} is 1000. ε in Eq. (4) is set to be 1.0E−100 to make sure that it exerts little influence on the gravitational force. The shrinking constant α in Eq. (5)

Table 1 Experimental results of benchmark functions (F1–F6) using traditional GSA, CGSA with 12 different chaos, CGSA-R, CGSA-P, and CGSA-M.

Algorithm	F1	F2	F3	F4	F5	F6
GSA	2.04E−17 ± 6.21E−18	2.31E−08 ± 3.49E−09	2.46E+02 ± 9.31E+01	3.23E−09 ± 7.46E−10	2.61E+01 ± 2.63E−01	0.00E+00 ± 0.00E+00
CGSA-1	1.16E−17 ± 3.11E−18	1.60E−08 ± 1.87E−09	2.25E+02 ± 9.03E+01	**2.73E−09 ± 6.76E−10**	2.82E+01 ± 1.12E+01	0.00E+00 ± 0.00E+00
CGSA-2	9.99E−18 ± 2.93E−18	1.59E−08 ± 2.35E−09	2.07E+02 ± 8.51E+01	3.07E−09 ± 6.73E−10	2.81E+01 ± 1.16E+01	0.00E+00 ± 0.00E+00
CGSA-3	1.08E−17 ± 2.86E−18	1.64E−08 ± 2.18E−09	2.24E+02 ± 7.96E+01	3.05E−09 ± 5.90E−10	3.29E+01 ± 2.58E+01	0.00E+00 ± 0.00E+00
CGSA-4	1.09E−17 ± 2.77E−18	1.67E−08 ± 2.73E−09	2.19E+02 ± 7.41E+01	1.04E−02 ± 5.72E−02	2.83E+01 ± 1.23E+01	0.00E+00 ± 0.00E+00
CGSA-5	1.21E−17 ± 3.11E−18	1.58E−08 ± 2.26E−09	1.97E+02 ± 6.53E+01	5.98E−04 ± 3.27E−03	2.93E+01 ± 1.39E+01	0.00E+00 ± 0.00E+00
CGSA-6	1.10E−17 ± 3.38E−18	1.65E−08 ± 2.25E−09	2.16E+02 ± 8.26E+01	3.63E−03 ± 1.99E−02	2.87E+01 ± 1.42E+01	0.00E+00 ± 0.00E+00
CGSA-7	1.07E−17 ± 2.78E−18	**1.54E−08 ± 1.82E−09**	2.16E+02 ± 8.32E+01	2.85E−09 ± 4.93E−10	2.60E+01 ± 2.00E−01	0.00E+00 ± 0.00E+00
CGSA-8	1.12E−17 ± 3.21E−18	1.63E−08 ± 3.16E−09	**1.92E+02 ± 7.68E+01**	4.57E−03 ± 2.50E−02	2.85E+01 ± 1.30E+01	0.00E+00 ± 0.00E+00
CGSA-9	1.12E−17 ± 2.88E−18	1.60E−08 ± 2.23E−09	2.19E+02 ± 9.91E+01	5.68E−03 ± 3.11E−02	2.95E+01 ± 1.93E+01	0.00E+00 ± 0.00E+00
CGSA-10	9.98E−18 ± 3.00E−18	1.62E−08 ± 2.03E−09	2.33E+02 ± 9.46E+01	4.10E−03 ± 2.24E−02	3.06E+01 ± 1.54E+01	0.00E+00 ± 0.00E+00
CGSA-11	1.26E−17 ± 5.33E−18	1.66E−08 ± 2.74E−09	2.48E+02 ± 1.05E+02	2.80E−09 ± 5.74E−10	2.61E+01 ± 2.86E−01	0.00E+00 ± 0.00E+00
CGSA-12	1.11E−17 ± 3.24E−18	1.59E−08 ± 2.78E−09	2.12E+02 ± 8.33E+01	3.11E−02 ± 1.70E−01	3.71E+01 ± 5.06E+01	0.00E+00 ± 0.00E+00
CGSA-R	1.17E−17 ± 4.12E−18	1.66E−08 ± 2.50E−09	2.11E+02 ± 8.92E+01	2.85E−09 ± 5.10E−10	8.64E+00 ± 1.18E+01	0.00E+00 ± 0.00E+00
CGSA-P	**9.10E−18 ± 2.87E−18**	1.66E−08 ± 2.50E−09	2.11E+02 ± 7.30E+01	2.91E−09 ± 7.14E−10	**7.02E+00 ± 1.75E+01**	0.00E+00 ± 0.00E+00
CGSA-M	1.19E−17 ± 3.82E−18	1.58E−08 ± 2.48E−09	2.32E+02 ± 7.41E+01	2.85E−09 ± 5.00E−10	1.46E+01 ± 3.55E+01	0.00E+00 ± 0.00E+00

Table 2 Experimental results of benchmark functions (F7–F12) using traditional GSA, CGSA with 12 different chaos, CGSA-R, CGSA-P, and CGSA-M.

Algorithm	F7	F8	F9	F10	F11	F12
GSA	1.97E−02 ± 1.23E−02	−2.66E+03 ± 4.17E+02	1.53E+01 ± 4.67E+00	3.60E−09 ± 4.55E−10	4.15E+00 ± 2.29E+00	3.76E−02 ± 1.32E−01
CGSA-1	1.16E−02 ± 4.62E−03	−2.89E+03 ± 4.95E+02	1.98E+01 ± 4.45E+00	2.68E−09 ± 4.81E−10	4.28E+00 ± 1.75E+00	4.33E−02 ± 9.29E−02
CGSA-2	1.11E−02 ± 5.43E−03	−2.81E+03 ± 4.97E+02	1.89E+01 ± 4.21E+00	2.63E−09 ± 4.26E−10	3.70E+00 ± 1.25E+00	3.44E−02 ± 9.33E−02
CGSA-3	1.12E−02 ± 6.63E−03	−2.98E+03 ± 4.87E+02	2.12E+01 ± 6.12E+00	2.69E−09 ± 3.44E−10	**3.61E+00 ± 1.38E+00**	1.07E−01 ± 3.79E−01
CGSA-4	1.34E−02 ± 4.72E−03	−2.96E+03 ± 5.55E+02	2.22E+01 ± 5.43E+00	2.74E−09 ± 3.00E−10	4.14E+00 ± 1.71E+00	3.91E−02 ± 7.32E−02
CGSA-5	1.18E−02 ± 6.83E−03	−2.82E+03 ± 5.33E+02	1.99E+01 ± 5.10E+00	2.65E−09 ± 3.60E−10	3.82E+00 ± 1.26E+00	3.45E−02 ± 6.76E−02
CGSA-6	1.26E−02 ± 4.55E−03	−2.90E+03 ± 4.23E+02	2.05E+01 ± 4.86E+00	2.67E−09 ± 3.44E−10	4.32E+00 ± 1.25E+00	1.62E−02 ± 4.61E−02
CGSA-7	1.21E−02 ± 6.41E−03	−3.01E+03 ± 3.85E+02	2.19E+01 ± 5.51E+00	2.67E−09 ± 4.13E−10	3.77E+00 ± 1.57E+00	6.28E−02 ± 2.14E−01
CGSA-8	1.10E−02 ± 5.58E−03	−2.68E+03 ± 3.60E+02	2.26E+01 ± 5.34E+00	2.65E−09 ± 3.76E−10	4.03E+00 ± 1.25E+00	4.30E−02 ± 9.36E−02
CGSA-9	**1.07E−02 ± 4.40E−03**	−2.74E+03 ± 4.43E+02	2.17E+01 ± 7.08E+00	2.75E−09 ± 4.76E−10	3.95E+00 ± 1.17E+00	4.15E−02 ± 9.66E−02
CGSA-10	1.08E−02 ± 4.58E−03	−2.94E+03 ± 4.91E+02	2.06E+01 ± 5.40E+00	2.65E−09 ± 3.47E−10	3.98E+00 ± 1.42E+00	3.99E−02 ± 9.56E−02
CGSA-11	1.20E−02 ± 5.24E−03	−2.89E+03 ± 4.98E+02	2.14E+01 ± 6.89E+00	2.60E−09 ± 3.36E−10	4.41E+00 ± 1.80E+00	5.09E−02 ± 1.15E−01
CGSA-12	1.10E−02 ± 4.52E−03	−2.72E+03 ± 4.58E+02	2.18E+01 ± 6.90E+00	2.88E−09 ± 4.76E−10	4.27E+10 ± 1.81E+00	1.73E−02 ± 3.93E−02
CGSA-R	1.13E−02 ± 4.65E−03	−3.06E+03 ± 6.34E+02	**1.46E+01 ± 3.91E+00**	2.68E−09 ± 3.14E−10	3.70E+00 ± 1.61E+00	4.09E−02 ± 8.16E−02
CGSA-P	1.18E−02 ± 3.80E−03	−3.07E+03 ± 5.83E+02	1.53E+01 ± 3.54E+00	2.63E−09 ± 3.90E−10	4.15E+00 ± 1.50E+00	**1.08E−02 ± 3.29E−02**
CGSA-M	1.10E−02 ± 4.31E−03	**−3.38E+03 ± 6.92E+02**	2.17E+01 ± 6.00E+00	**2.56E−09 ± 3.64E−10**	3.72E+00 ± 1.40E+00	3.85E−02 ± 1.20E−01

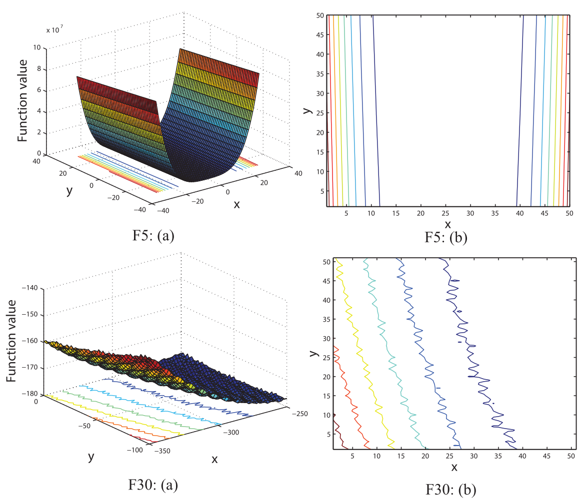

Fig. 1 The 2-dimensional sketch (a) and the contour (b) for the unimodal function F5 and the shifted rotated Griewank's function F30, respectively.

controls the decrease speed of $G(t)$ and is set to be $0.02t_{max}$. The initial value of the gravitational parameter $G_0 = 100$. The attraction scope parameter β in Eq. (7) is set to be 2%. As suggested in [34], we adopt $LP = 50$ for CGSA-M.

Tables 1–8 summarize the experimental results of GSA, CGSA with 12 different chaos, CGSA-R, CGSA-P and CGSA-M for 48 tested benchmark functions. The recorded results are shown in the form of *Ave.* ± *Dev.*, where *Ave.* denotes the average of the optimization error (final best-so-far solution) of 30 independent runs for each algorithm, and *Dev.* represents its standard deviation. The best result among the compared 16 algorithms is shown in bold. From Tables 1–8, we can find that

1. The best results are always obtained by one of the variants of chaotic GSA rather than GSA, which suggests that the chaotic local search definitely improves the search performance of GSA.
2. The proposed MCGSA (including CGSA-R, CGSA-P and CGSA-M) can acquire the best solutions for 31 out of 48 benchmark functions.
3. On the other hand, all twelve variants of single chaos embedded GSA can perform the best for only 18 benchmark functions.
4. Thus, it can be stated that the multiple chaos incorporation scheme is generally better than the single one for improving the performance of GSA.

To give some insights into the search performance of compared algorithms, Figs. 2 and 3 depict the convergence graphs and distributions of the final solutions for functions F5 and F30 respectively. Two kinds of convergence graphs are utilized: one is the average best-so-far solutions versus the iteration number, and the other is the ratio of the best-so-far solution versus the iteration number. In Figs. 2 (a) and 3 (a), the horizontal axis in a linear scale indicates the generation (i.e., iteration number) of the algorithm, while the

Table 3 Experimental results of benchmark functions (F13–F18) using traditional GSA, CGSA with 12 different chaos, CGSA-R, CGSA-P, and CGSA-M.

Algorithm	F13	F14	F15	F16	F17	F18
GSA	7.32E−04 ± 2.79E−03	3.61E+00 ± 2.86E+00	1.94E−03 ± 3.48E−04	−1.03E+00 ± 0.00E+00	3.98E−01 ± 1.13E−16	3.00E+00 ± 5.08E−15
CGSA-1	1.23E−03 ± 3.81E−03	1.20E+00 ± 5.46E−01	1.45E−03 ± 9.07E−04	−1.03E+00 ± 0.00E+00	3.98E−01 ± 1.13E−16	3.00E+00 ± 4.29E−15
CGSA-2	7.32E−04 ± 2.79E−03	1.31E+00 ± 6.55E−01	1.22E−03 ± 5.66E−04	−1.03E+00 ± 5.45E−16	3.98E−01 ± 0.00E+00	3.00E+00 ± 2.33E−15
CGSA-3	1.37E−03 ± 4.50E−03	1.84E+00 ± 1.38E+00	1.19E−03 ± 5.53E−04	−1.03E+00 ± 5.68E−16	3.98E−01 ± 0.00E+00	3.00E+00 ± 2.09E−15
CGSA-4	1.09E−03 ± 3.36E−03	1.43E+00 ± 7.65E−01	1.21E−03 ± 6.07E−04	−1.03E+00 ± 6.05E−16	3.98E−01 ± 0.00E+00	3.00E+00 ± 2.39E−15
CGSA-5	**1.12E−18 ± 2.98E−19**	1.30E+00 ± 6.71E+00	1.51E−03 ± 8.29E−04	−1.03E+00 ± 5.53E−16	3.98E−01 ± 0.00E+00	3.00E+00 ± 2.23E−15
CGSA-6	1.12E−18 ± 3.59E−19	1.43E+00 ± 8.90E−01	1.26E−03 ± 6.88E−04	−1.03E+00 ± 6.12E−16	3.98E−01 ± 0.00E+00	3.00E+00 ± 2.17E−15
CGSA-7	1.97E−03 ± 4.53E−03	1.37E+00 ± 6.17E−01	1.24E−03 ± 6.20E−04	−1.03E+00 ± 0.00E+00	3.98E−01 ± 1.13E−16	3.00E+00 ± 4.42E−15
CGSA-8	1.37E−03 ± 3.59E−03	1.61E+00 ± 1.17E+00	1.15E−03 ± 6.32E−04	−1.03E+00 ± 0.00E+00	3.98E−01 ± 1.13E−16	3.00E+00 ± 4.81E−15
CGSA-9	1.61E−03 ± 4.29E−03	1.51E+00 ± 9.80E−01	1.22E−03 ± 4.47E−04	−1.03E+00 ± 5.83E−16	3.98E−01 ± 2.93E−04	3.00E+00 ± 2.22E−15
CGSA-10	9.56E−04 ± 4.10E−03	1.80E+00 ± 9.53E−01	1.36E−03 ± 6.59E−04	−1.03E+00 ± 5.90E−16	3.98E−01 ± 4.66E−06	3.00E+00 ± 1.47E−15
CGSA-11	3.66E−04 ± 2.01E−03	1.45E+00 ± 7.64E−01	1.29E−03 ± 9.15E−04	−1.03E+00 ± 5.68E−16	3.98E−01 ± 0.00E+00	3.00E+00 ± 2.26E−15
CGSA-12	2.60E−04 ± 1.42E−03	1.39E+00 ± 6.67E−01	1.02E−03 ± 4.64E−04	−1.03E+00 ± 5.68E−16	3.98E−01 ± 9.43E−05	3.00E+00 ± 2.02E−15
CGSA-R	3.10E−04 ± 1.70E−03	1.59E+00 ± 1.23E+00	1.30E−03 ± 6.95E−04	**−1.03E+00 ± 0.00E+00**	**3.98E−01 ± 1.13E−16**	**3.00E+00 ± 4.93E−15**
CGSA-P	1.01E−03 ± 3.12E−03	**1.20E+00 ± 4.80E−01**	1.25E−03 ± 5.48E−04	**−1.03E+00 ± 0.00E+00**	**3.98E−01 ± 1.13E−16**	**3.00E+00 ± 5.50E−15**
CGSA-M	3.66E−04 ± 2.01E−03	1.37E+00 ± 1.14E+00	**1.02E−03 ± 3.63E−04**	**−1.03E+00 ± 0.00E+00**	**3.98E−01 ± 1.73E−05**	**3.00E+00 ± 4.97E−15**

Table 4 Experimental results of benchmark functions (F19–F24) using traditional GSA, CGSA with 12 different chaos, CGSA-R, CGSA-P, and CGSA-M.

Algorithm	F19	F20	F21	F22	F23	F24
GSA	−3.86E+00 ± 2.71E−15	−3.32E+00 ± 1.36E−15	−7.35E+00 ± 3.44E+00	−1.04E+01 ± 0.00E+00	−1.05E+01 ± 9.03E−15	1.28E+03 ± 5.03E+02
CGSA-1	−3.86E+00 ± 2.71E−15	−3.28E+00 ± 5.70E−02	−7.39E+00 ± 3.11E+00	−9.97E+00 ± 1.67E+00	−1.02E+01 ± 1.37E+00	1.27E+03 ± 6.94E+02
CGSA-2	−3.86E+00 ± 2.48E−15	−3.29E+00 ± 5.54E−02	−7.31E+00 ± 3.20E+00	−9.27E+00 ± 2.34E+00	−1.00E+01 ± 1.64E+00	1.57E+03 ± 8.15E+02
CGSA-3	−3.86E+00 ± 2.54E−15	−3.30E+00 ± 4.51E−02	−7.89E+00 ± 3.12E+00	−9.52E+00 ± 2.01E+00	−9.92E+00 ± 1.90E+00	1.14E+03 ± 7.77E+02
CGSA-4	−3.86E+00 ± 2.45E−15	−3.29E+00 ± 5.54E−02	−8.65E+00 ± 2.84E+00	−9.62E+00 ± 2.06E+00	−1.05E+01 ± 1.78E−15	1.49E+03 ± 9.77E+02
CGSA-5	−3.86E+00 ± 2.36E−15	−3.28E+00 ± 5.83E−02	−7.72E+00 ± 3.12E+00	−9.19E+00 ± 2.52E+00	−9.74E+00 ± 2.10E+00	1.15E+03 ± 5.86E+02
CGSA-6	−3.86E+00 ± 2.37E−15	−3.30E+00 ± 4.84E−02	−7.13E+00 ± 3.19E+00	−9.52E+00 ± 2.01E+00	−1.01E+00 ± 1.68E+00	1.25E+03 ± 7.38E+02
CGSA-7	−3.86E+00 ± 2.71E−15	−3.31E+00 ± 4.11E−02	−8.15E+00 ± 3.18E+00	−9.27E+00 ± 2.34E+00	−9.67E+00 ± 2.30E+00	1.44E+03 ± 7.28E+02
CGSA-8	−3.86E+00 ± 2.71E−15	−3.28E+00 ± 5.70E−02	−6.98E+00 ± 3.53E+00	−9.44E+00 ± 2.21E+00	−1.02E+00 ± 1.36E+00	1.26E+03 ± 8.00E+02
CGSA-9	−3.86E+00 ± 2.45E−15	−3.30E+00 ± 4.51E−02	−8.06E+00 ± 3.10E+00	−9.54E+00 ± 2.28E+00	−9.92E+00 ± 1.90E+00	1.47E+03 ± 7.29E+02
CGSA-10	−3.86E+00 ± 2.46E−15	−3.30E+00 ± 4.51E−02	**−8.89E+00 ± 2.37E+00**	−9.62E+00 ± 2.07E+00	−9.83E+00 ± 2.18E+00	1.45E+03 ± 9.86E+02
CGSA-11	−3.86E+00 ± 2.49E−15	−3.26E+00 ± 6.05E−02	−7.64E+00 ± 3.22E+00	−9.52E+00 ± 2.00E+00	−1.02E+00 ± 1.37E+00	1.49E+03 ± 7.68E+02
CGSA-12	−3.86E+00 ± 2.40E−15	−3.30E+00 ± 4.51E−02	−6.99E+00 ± 3.52E+00	−9.89E+00 ± 1.94E+00	−1.00E+00 ± 1.64E+00	1.33E+03 ± 7.20E+02
CGSA-R	**−3.86E+00 ± 2.71E−15**	**−3.32E+00 ± 1.36E−15**	−8.22E+00 ± 2.62E+00	**−1.04E+01 ± 0.00E+00**	**−1.05E+01 ± 9.03E−15**	**1.13E+03 ± 7.35E+02**
CGSA-P	**−3.86E+00 ± 2.71E−15**	**−3.32E+00 ± 1.36E−15**	−7.64E+00 ± 3.02E+00	−9.87E+00 ± 1.61E+00	**−1.05E+01 ± 9.03E−15**	1.20E+03 ± 7.27E+02
CGSA-M	**−3.86E+00 ± 2.71E−15**	−3.29E+00 ± 5.54E−02	−7.88E +00 ± 2.91E+00	−9.87E+00 ± 1.61E+00	**−1.05E+01 ± 9.03E−15**	1.15E+03 ± 4.78E+02

Table 5 Experimental results of benchmark functions (F25–F30) using traditional GSA, CGSA with 12 different chaos, CGSA-R, CGSA-P, and CGSA-M.

Algorithm	F25	F26	F27	F28	F29	F30
GSA	2.00E+04 ± 2.05E+03	5.09E+07 ± 7.46E+07	6.79E+04 ± 1.34E+04	2.28E+04 ± 2.12E+03	1.26E+08 ± 8.42E+07	1.21E+04 ± 2.97E+02
CGSA-1	1.97E+04 ± 1.98E+03	4.69E+07 ± 4.64E+07	5.36E+04 ± 1.07E+04	2.29E+04 ± 3.02E+03	1.10E+08 ± 6.11E+07	1.15E+04 ± 5.33E+02
CGSA-2	1.95E+04 ± 3.08E+03	4.36E+07 ± 4.97E+07	5.61E+04 ± 1.12E+04	2.32E+04 ± 3.61E+03	1.09E+08 ± 4.28E+07	1.16E+04 ± 4.40E+02
CGSA-3	**1.93E+04 ± 2.73E+03**	4.33E+07 ± 3.22E+07	5.04E+04 ± 8.65E+03	2.27E+04 ± 2.55E+03	1.03E+08 ± 4.56E+07	1.14E+04 ± 6.32E+02
CGSA-4	1.98E+04 ± 2.16E+03	3.81E+07 ± 3.92E+07	5.62E+04 ± 9.62E+03	**2.16E+04 ± 2.44E+03**	1.15E+08 ± 7.45E+07	1.16E+04 ± 4.98E+02
CGSA-5	1.97E+04 ± 2.07E+03	4.18E+07 ± 5.02E+07	5.14E+04 ± 9.68E+03	2.30E+04 ± 2.89E+03	1.13E+08 ± 5.72E+07	1.16E+04 ± 4.34E+02
CGSA-6	1.96E+04 ± 2.32E+03	3.77E+07 ± 2.93E+07	5.55E+04 ± 1.45E+04	2.30E+04 ± 2.61E+03	1.14E+08 ± 5.68E+07	1.13E+04 ± 5.01E+02
CGSA-7	2.03E+04 ± 2.26E+03	3.96E+07 ± 3.64E+07	5.50E+04 ± 1.26E+04	2.30E+04 ± 2.67E+03	1.23E+08 ± 6.93E+07	1.14E+04 ± 5.48E+02
CGSA-8	1.98E+04 ± 2.68E+03	4.94E+07 ± 4.80E+07	5.76E+04 ± 1.57E+04	2.19E+04 ± 2.76E+03	1.09E+08 ± 5.74E+07	1.15E+04 ± 5.05E+02
CGSA-9	2.01E+04 ± 2.02E+03	4.52E+07 ± 5.25E+07	5.35E+04 ± 9.18E+03	2.24E+04 ± 2.99E+03	1.23E+08 ± 6.99E+07	1.15E+04 ± 5.23E+02
CGSA-10	2.07E+04 ± 2.22E+03	3.50E+07 ± 3.50E+07	**4.98E+04 ± 9.69E+03**	2.24E+04 ± 2.75E+03	1.36E+08 ± 9.57E+07	1.14E+04 ± 4.86E+02
CGSA-11	1.98E+04 ± 2.35E+03	4.38E+07 ± 3.35E+07	5.77E+04 ± 1.34E+04	2.29E+04 ± 2.57E+03	1.33E+08 ± 7.86E+07	1.15E+04 ± 5.05E+02
CGSA-12	1.96E+04 ± 2.36E+03	4.32E+07 ± 4.17E+07	5.58E+04 ± 1.33E+04	2.22E+04 ± 2.71E+03	1.22E+08 ± 6.04E+07	1.16E+04 ± 4.12E+02
CGSA-R	1.95E+04 ± 2.22E+03	3.33E+07 ± 2.73E+07	5.93E+04 ± 1.68E+04	2.23E+04 ± 2.57E+03	1.15E+08 ± 7.58E+07	1.09E+04 ± 4.99E+02
CGSA-P	2.01E+04 ± 2.74E+03	**3.18E+07 ± 2.41E+07**	5.53E+04 ± 1.07E+04	2.28E+04 ± 3.18E+03	**1.00E+08 ± 5.17E+07**	**8.70E+03 ± 6.70E+02**
CGSA-M	1.96E+04 ± 2.69E+03	4.05E+07 ± 5.30E+07	5.44E+04 ± 1.15E+04	2.21E+04 ± 2.22E+03	1.04E+08 ± 7.30E+07	1.12E+04 ± 4.59E+02

vertical axis in a logarithmic scale represents the average fitness of the best-so-far solutions generated by the algorithm. From these two sub-figures, it can be found that all algorithms converge quickly in early phases of the iteration, and trapped into local minima at the later phases. The best final solutions are obtained by CGSA-P for both F5 and F30.

On the other hand, the ratio of the best-so-far solutions found by chaotic GSAs to those found by GSA is depicted in Figs. 2 (b) and 3 (b), aiming to verify the effects of chaotic local search on the GSA. From Fig. 2 (b), we observe that all chaotic GSAs can find better solutions than GSA in earlier phases, suggesting that the chaotic local search is able to improve GSA in an exploitation manner. However, some of chaotic GSAs generate worse solutions than GSA in the latter search phase and cannot improve solutions' qualities any further. It reveals that the chaotic local search cannot always improve the performance of GSA for all the time. Its effects strongly depend on the used chaotic map, which is usually the common case in chaotic meta-heuristics. In this figure, it is clear that the proposed MCGSA (including CGSA-R, CGSA-P and CGSA-M) can generate better solutions than the other compared algorithms. A same phenomenon that MCGSA performs better than the others can also be observed in Fig. 3 (b).

Due to the stochastic feature of all compared 16 algorithms, a box-and-whisker diagram is used to depict the dis-

Table 6 Experimental results of benchmark functions (F31–F36) using traditional GSA, CGSA with 12 different chaos, CGSA-R, CGSA-P, and CGSA-M.

Algorithm	F31	F32	F33	F34	F35	F36
GSA	2.03E+01 ± 9.60E−02	4.47E+01 ± 8.85E+00	3.58E+01 ± 7.50E+00	1.99E+00 ± 1.35E+00	5.28E+03 ± 6.81E+03	6.04E+00 ± 1.10E+00
CGSA-1	2.03E+01 ± 1.14E−01	5.48E+01 ± 1.14E+01	4.55E+01 ± 1.21E+01	4.16E+00 ± 2.55E+00	1.58E+03 ± 2.80E+03	5.16E+00 ± 1.10E+00
CGSA-2	2.02E+01 ± 9.34E−02	5.51E+01 ± 1.16E+01	5.22E+01 ± 1.20E+01	3.76E+00 ± 2.04E+00	2.53E+03 ± 3.97E+03	5.15E+00 ± 1.23E+00
CGSA-3	**2.02E+01 ± 7.44E−02**	5.21E+01 ± 9.32E+00	4.93E+01 ± 9.41E+00	4.57E+00 ± 3.28E+00	1.74E+03 ± 2.31E+03	5.06E+00 ± 1.19E+00
CGSA-4	2.03E+01 ± 7.26E−02	4.88E+01 ± 9.36E+00	4.88E+01 ± 1.30E+01	4.17E+00 ± 2.52E+00	2.10E+03 ± 2.98E+03	5.27E+00 ± 1.30E+00
CGSA-5	2.02E+01 ± 7.29E−02	5.35E+01 ± 1.30E+01	5.01E+01 ± 1.40E+01	3.98E+00 ± 2.09E+00	2.31E+03 ± 3.95E+03	5.36E+00 ± 1.12E+00
CGSA-6	2.03E+01 ± 9.75E−02	5.08E+01 ± 9.70E+00	5.34E+01 ± 1.16E+01	3.68E+00 ± 2.19E+00	1.67E+03 ± 2.16E+03	5.16E+00 ± 1.21E+00
CGSA-7	2.03E+01 ± 7.66E−02	5.07E+01 ± 1.12E+01	4.79E+01 ± 1.11E+01	3.56E+00 ± 2.34E+00	2.85E+03 ± 3.75E+03	5.28E+00 ± 1.27E+00
CGSA-8	2.03E+01 ± 1.04E−01	5.24E+01 ± 9.35E+00	4.79E+01 ± 9.65E+00	4.19E+00 ± 2.29E+00	3.25E+03 ± 9.69E+03	5.41E+00 ± 1.03E+00
CGSA-9	2.03E+01 ± 9.20E−02	5.59E+01 ± 1.14E+01	5.18E+01 ± 1.37E+01	3.96E+00 ± 2.48E+00	2.63E+03 ± 5.25E+03	5.29E+00 ± 1.02E+00
CGSA-10	2.03E+01 ± 8.85E−02	5.44E+01 ± 9.68E+00	4.78E+01 ± 1.20E+01	3.47E+00 ± 1.98E+00	2.21E+03 ± 2.35E+03	5.18E+00 ± 1.21E+00
CGSA-11	2.02E+01 ± 9.50E−02	5.21E+01 ± 1.12E+01	4.66E+01 ± 1.13E+01	4.04E+00 ± 2.07E+00	2.29E+03 ± 3.07E+03	5.16E+00 ± 1.32E+00
CGSA-12	2.03E+01 ± 1.02E−01	5.12E+01 ± 9.90E+00	5.17E+01 ± 1.34E+01	3.59E+00 ± 2.29E+00	**1.46E+03 ± 1.62E+03**	5.02E+00 ± 1.25E+00
CGSA-R	2.03E+01 ± 8.38E−02	4.19E+01 ± 7.18E+00	3.50E+01 ± 7.43E+00	**1.80E+00 ± 1.43E+00**	1.62E+03 ± 2.54E+03	**4.88E+00 ± 9.47E−01**
CGSA-P	2.03E+01 ± 8.82E−02	**4.05E+01 ± 7.41E+00**	**3.41E+01 ± 8.11E+00**	1.98E+00 ± 1.56E+00	2.92E+03 ± 4.00E+03	4.95E+00 ± 1.27E+00
CGSA-M	2.03E+01 ± 1.03E−01	5.79E+01 ± 1.34E+01	5.10E+01 ± 1.03E+01	4.04E+00 ± 2.85E+00	2.26E+03 ± 4.35E+03	5.28E+00 ± 1.21E+00

Table 7 Experimental results of benchmark functions (F37–F42) using traditional GSA, CGSA with 12 different chaos, CGSA-R, CGSA-P, and CGSA-M.

Algorithm	F37	F38	F39	F40	F41	F42
GSA	1.43E+01 ± 1.30E−01	3.23E+02 ± 4.85E+01	2.40E+02 ± 2.17E+02	3.51E+02 ± 2.60E+02	9.66E+02 ± 3.40E+01	9.55E+02 ± 5.03E+01
CGSA-1	1.38E+01 ± 2.57E−01	3.45E+02 ± 5.86E+01	2.07E+02 ± 1.69E+02	2.16E+02 ± 1.79E+02	9.70E+02 ± 4.99E+01	9.69E+02 ± 3.73E+01
CGSA-2	1.38E+01 ± 3.06E−01	3.50E+02 ± 6.74E+01	1.70E+02 ± 1.54E+02	2.82E+02 ± 2.11E+02	9.72E+02 ± 3.77E+01	9.64E+02 ± 5.77E+01
CGSA-3	1.38E+01 ± 2.28E−01	3.66E+02 ± 8.37E+01	2.55E+02 ± 1.93E+02	3.04E+02 ± 2.13E+02	9.69E+02 ± 4.84E+01	9.70E+02 ± 4.81E+01
CGSA-4	1.38E+01 ± 3.32E−01	3.55E+02 ± 8.21E+01	1.93E+02 ± 1.68E+02	**1.63E+02 ± 1.36E+02**	9.71E+02 ± 3.83E+01	9.59E+02 ± 6.24E+01
CGSA-5	1.39E+01 ± 2.79E−01	3.44E+02 ± 6.06E+01	2.21E+02 ± 1.77E+02	2.37E+02 ± 1.83E+02	9.73E+02 ± 3.56E+01	**9.44E+02 ± 6.90E+01**
CGSA-6	1.37E+01 ± 3.13E−01	3.72E+02 ± 7.22E+01	2.44E+02 ± 1.83E+02	2.71E+02 ± 2.02E+02	**9.41E+02 ± 7.80E+01**	9.66E+02 ± 4.73E+01
CGSA-7	1.38E+01 ± 2.59E−01	3.55E+02 ± 6.21E+01	1.26E+02 ± 9.50E+01	2.36E+02 ± 1.92E+02	9.74E+02 ± 5.02E+01	9.78E+02 ± 1.78E+01
CGSA-8	1.38E+01 ± 3.12E−01	3.31E+02 ± 8.11E+01	**1.14E+02 ± 9.44E+01**	2.72E+02 ± 2.17E+02	9.75E+02 ± 3.62E+01	9.64E+02 ± 4.85E+01
CGSA-9	1.38E+01 ± 2.71E−01	3.45E+02 ± 6.12E+01	1.30E+02 ± 1.19E+02	2.27E+02 ± 1.92E+02	9.70E+02 ± 4.86E+01	9.64E+02 ± 5.79E+01
CGSA-10	1.37E+01 ± 3.25E−01	3.27E+02 ± 4.61E+01	1.59E+02 ± 1.57E+02	2.51E+02 ± 1.91E+02	9.71E+02 ± 5.00E+01	9.53E+02 ± 6.40E+01
CGSA-11	1.38E+01 ± 3.06E−01	3.45E+02 ± 7.11E+01	1.33E+02 ± 1.26E+02	2.25E+02 ± 1.87E+02	9.55E+02 ± 6.39E+01	9.68E+02 ± 5.92E+01
CGSA-12	1.38E+01 ± 2.35E−01	3.44E+02 ± 7.20E+01	1.69E+02 ± 1.48E+02	3.06E+02 ± 2.09E+02	9.46E+02 ± 7.45E+01	9.73E+02 ± 4.77E+01
CGSA-R	1.38E+01 ± 3.02E−01	3.43E+02 ± 6.88E+01	2.29E+02 ± 1.67E+02	2.49E+02 ± 2.00E+02	9.76E+02 ± 2.42E+01	9.68E+02 ± 4.76E+01
CGSA-P	1.38E+01 ± 2.48E−01	**3.02E+02 ± 9.91E+00**	1.77E+02 ± 1.38E+02	2.82E+02 ± 1.97E+02	9.51E+02 ± 5.77E+01	9.50E+02 ± 6.09E+01
CGSA-M	**1.37E+01 ± 2.81E−01**	3.56E+02 ± 7.79E+01	2.20E+02 ± 1.90E+02	2.55E+02 ± 1.95E+02	9.83E+02 ± 2.00E+01	9.76E+02 ± 3.78E+01

Table 8 Experimental results of benchmark functions (F43–F48) using traditional GSA, CGSA with 12 different chaos, CGSA-R, CGSA-P, and CGSA-M.

Algorithm	F43	F44	F45	F46	F47	F48
GSA	9.66E+02 ± 3.78E+01	6.73E+02 ± 2.92E+02	9.31E+02 ± 1.71E+01	7.58E+02 ± 2.82E+02	2.55E+02 ± 2.11E+02	1.68E+03 ± 1.29E+01
CGSA-1	9.63E+02 ± 5.66E+01	7.83E+02 ± 3.29E+02	9.62E+02 ± 2.61E+01	8.10E+02 ± 3.10E+02	4.94E+02 ± 4.57E+02	1.73E+03 ± 2.59E+01
CGSA-2	9.68E+02 ± 4.90E+01	6.96E+02 ± 3.04E+02	9.63E+02 ± 2.28E+01	8.27E+02 ± 3.01E+02	7.36E+02 ± 4.79E+02	1.72E+03 ± 2.35E+01
CGSA-3	**9.55E+02 ± 7.27E+01**	7.61E+02 ± 3.25E+02	9.65E+02 ± 2.49E+01	8.84E+02 ± 2.91E+02	4.97E+02 ± 4.32E+02	1.72E+03 ± 2.19E+01
CGSA-4	9.64E+02 ± 5.75E+01	7.61E+02 ± 3.26E+02	9.61E+02 ± 2.91E+01	8.79E+02 ± 3.02E+02	3.93E+02 ± 3.94E+02	1.72E+03 ± 1.82E+01
CGSA-5	9.74E+02 ± 3.67E+01	7.62E+02 ± 3.26E+02	9.56E+02 ± 1.88E+01	8.18E+02 ± 2.92E+02	5.39E+02 ± 4.70E+02	1.73E+03 ± 2.85E+01
CGSA-6	9.73E+02 ± 3.76E+01	7.82E+02 ± 3.23E+02	9.57E+02 ± 2.79E+01	7.63E+02 ± 2.94E+02	5.14E+02 ± 4.53E+02	1.73E+03 ± 1.79E+01
CGSA-7	9.71E+02 ± 3.63E+01	8.28E+02 ± 3.34E+02	9.72E+02 ± 2.77E+01	8.17E+02 ± 3.19E+02	4.18E+02 ± 4.05E+02	1.73E+03 ± 2.81E+01
CGSA-8	9.65E+02 ± 4.69E+01	7.84E+02 ± 3.30E+02	9.59E+02 ± 2.63E+01	7.83E+02 ± 3.03E+02	5.51E+02 ± 4.60E+02	1.73E+03 ± 3.08E+01
CGSA-9	9.68E+02 ± 4.73E+01	7.40E+02 ± 3.20E+02	9.52E+02 ± 2.52E+01	8.21E+02 ± 3.02E+02	3.88E+02 ± 3.83E+02	1.73E+03 ± 2.64E+01
CGSA-10	9.57E+02 ± 5.46E+01	9.13E+02 ± 3.20E+02	9.59E+02 ± 2.14E+01	8.58E+02 ± 2.99E+02	4.52E+02 ± 4.26E+02	1.73E+03 ± 2.23E+01
CGSA-11	9.80E+02 ± 3.54E+01	6.52E+02 ± 2.80E+02	9.66E+02 ± 2.31E+01	7.56E+02 ± 2.89E+02	5.76E+02 ± 4.71E+02	1.73E+03 ± 2.54E+01
CGSA-12	9.71E+02 ± 4.89E+01	8.07E+02 ± 3.29E+02	9.70E+02 ± 3.09E+01	8.55E+02 ± 3.10E+02	5.79E+02 ± 4.74E+02	1.73E+03 ± 2.08E+01
CGSA-R	9.63E+02 ± 4.72E+01	7.83E+02 ± 3.30E+02	9.76E+02 ± 2.29E+01	**7.29E+02 ± 2.74E+02**	4.52E+02 ± 4.30E+02	**1.66E+03 ± 1.42E+01**
CGSA-P	9.83E+02 ± 1.87E+01	**6.52E+02 ± 2.80E+02**	**9.29E+02 ± 1.76E+01**	8.03E+02 ± 3.04E+02	**2.37E+02 ± 1.76E+02**	1.75E+03 ± 2.76E+01
CGSA-M	9.80E+02 ± 3.83E+01	8.26E+02 ± 3.32E+02	9.79E+02 ± 3.76E+01	7.74E+02 ± 2.95E+02	5.52E+02 ± 4.72E+02	1.73E+03 ± 2.86E+01

tribution of final obtained best-so-far solutions in Figs. 2 (c) and 3(c). In these figures, five characteristic values including the smallest observation, lower quartile, median, upper quartile, and the largest observation are illustrated. Symbol + indicates outliers. From these sub-figures, it is quite clear that MCGSA outperforms its competitive algorithms. Especially, CGSA-P performs the best among all compared algorithms.

To further demonstrate the effectiveness and robustness of the proposed MCGSA, the average rankings of the algorithms obtained by the Friedman test [37], [38] on all tested 48 benchmark optimization functions are summarized in Tables 9–11. The Friedman test is a nonparametric statistical test which applies the *post hoc* method of Iman-Davenport [38]. It can rank the algorithms for each problem separately. The best performing algorithm among all compared algorithms should have rank 1, the second best rank 2, and so on. From these results, it can be found that the best performing algorithm usually changes for a certain optimization function. It is difficult to find such an algorithm which can perform the best for all tested problems (also known as the No Fee Lunch Theorem [39]). Thus, we confirm that the performance of an algorithm not only depends on its searching capacity, but also relies on the fitness structure (shown as in Fig. 1) of the solved function.

Furthermore, the values in the last column of Table 11

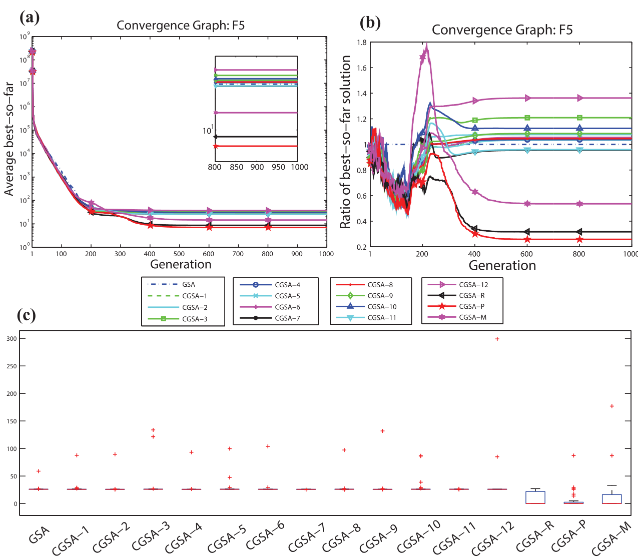

Fig. 2 Search performance of the algorithms for comparison on F5.

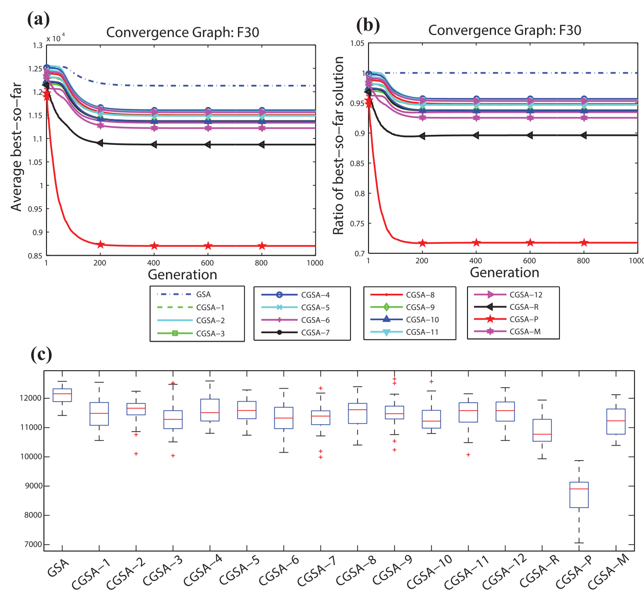

Fig. 3 Search performance of the algorithms for comparison on F30.

Table 9 Rankings of 16 variants of GSAs based on Friedman test for benchmark optimization function F1–F16.

Algorithm	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16
GSA	14.8	15.4	10.4	10.7	10.6	8.5	12.2	11.0	5.0	15.4	8.0	8.6	12.8	13.6	13.8	8.5
CGSA-1	8.8	8.2	8.7	6.5	9.2	8.5	8.5	8.6	8.6	8.0	9.0	9.1	8.3	6.9	9.1	8.5
CGSA-2	6.8	8.0	8.1	10.0	9.1	8.5	7.6	9.2	8.0	7.7	7.6	8.2	8.5	7.1	8.4	8.5
CGSA-3	7.7	8.7	9.2	9.6	10.9	8.5	7.2	7.3	9.2	8.2	7.1	8.5	8.8	10.4	7.8	8.5
CGSA-4	8.1	8.8	9.0	8.3	9.0	8.5	9.8	8.0	10.6	9.0	8.8	8.9	8.6	8.3	8.2	8.5
CGSA-5	9.8	7.4	7.4	9.7	9.3	8.5	8.2	9.6	9.0	7.8	8.2	8.7	7.7	8.1	10.2	8.5
CGSA-6	7.7	8.5	8.2	8.4	9.4	8.5	9.2	8.5	9.3	8.0	9.9	7.9	7.9	7.8	8.0	8.5
CGSA-7	7.7	7.1	8.4	8.2	8.4	8.5	8.6	7.0	10.2	8.0	7.7	8.6	8.7	8.8	8.4	8.5
CGSA-8	8.4	7.8	6.5	9.0	10.3	8.5	7.7	11.1	10.5	7.7	9.1	8.6	9.3	8.7	7.3	8.5
CGSA-9	8.4	7.9	8.2	8.9	8.5	8.5	7.5	9.9	9.8	8.5	8.7	8.3	8.4	8.7	8.4	8.5
CGSA-10	6.8	7.6	9.2	7.7	10.0	8.5	7.9	8.1	9.0	8.1	8.5	8.8	8.0	9.3	8.8	8.5
CGSA-11	9.3	8.8	9.7	7.6	10.0	8.5	8.5	8.6	9.5	7.0	9.5	9.3	7.7	8.9	7.7	8.5
CGSA-12	8.1	8.0	8.0	7.3	9.9	8.5	8.3	10.2	9.0	9.9	9.0	7.8	7.5	7.9	6.5	8.5
CGSA-R	8.8	8.6	7.6	8.1	4.0	8.5	8.2	7.1	3.7	8.4	7.8	9.0	8.0	7.6	8.4	8.5
CGSA-P	5.7	8.4	8.1	8.2	3.1	8.5	8.9	6.7	4.6	7.5	9.2	7.4	8.3	6.7	8.4	8.5
CGSA-M	9.1	7.0	9.2	7.9	3.8	8.5	7.7	5.0	10.1	6.7	8.0	8.3	7.5	7.4	6.7	8.5

Table 10 Rankings of 16 variants of GSAs based on Friedman test for benchmark optimization function F17–F32.

Algorithm	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30	F31	F32
GSA	8.4	8.5	8.5	6.8	8.8	7.5	8.0	8.8	8.9	8.2	12.7	9.3	9.1	14.5	9.8	5.8
CGSA-1	8.4	8.5	8.5	9.5	8.8	8.1	8.5	8.1	7.9	8.7	7.9	9.5	8.2	9.2	8.4	9.7
CGSA-2	8.4	8.5	8.5	9.2	9.0	9.1	8.8	10.4	7.2	9.0	9.0	9.2	8.8	10.3	7.4	10.2
CGSA-3	8.4	8.5	8.5	8.1	8.4	8.9	8.8	7.1	7.6	9.3	6.5	8.7	7.8	8.0	7.2	9.1
CGSA-4	8.4	8.5	8.5	9.2	7.5	8.6	8.0	9.5	8.2	8.4	9.0	6.9	8.3	9.6	8.1	7.9
CGSA-5	8.4	8.5	8.5	9.7	8.6	9.2	9.0	7.5	8.4	7.5	7.0	9.0	8.8	9.9	7.6	9.4
CGSA-6	8.4	8.5	8.5	8.4	9.4	8.8	8.5	7.8	7.9	9.2	8.0	9.3	8.9	7.9	8.0	9.0
CGSA-7	8.4	8.5	8.5	7.9	8.2	9.1	9.1	9.3	9.3	8.8	8.0	9.2	8.8	8.0	9.8	8.3
CGSA-8	8.4	8.5	8.5	9.5	9.3	8.8	8.5	8.3	8.6	8.8	9.1	7.6	8.3	9.5	9.3	9.4
CGSA-9	8.7	8.5	8.5	8.1	8.3	8.6	8.8	9.7	9.3	8.1	8.1	8.1	9.4	9.0	9.8	10.4
CGSA-10	8.7	8.5	8.5	8.1	7.0	8.6	8.8	8.7	10.4	7.8	6.4	8.5	8.7	8.1	8.3	10.2
CGSA-11	8.4	8.5	8.5	10.8	8.7	8.8	8.5	9.9	8.6	9.2	9.6	8.6	9.4	9.7	7.9	8.5
CGSA-12	8.7	8.5	8.5	8.1	9.3	8.1	8.8	8.6	8.2	9.1	8.1	7.7	9.1	9.9	7.7	8.4
CGSA-R	8.4	8.5	8.5	6.9	7.8	7.5	8.0	7.0	8.3	7.8	9.7	7.9	7.9	4.9	8.7	4.4
CGSA-P	8.4	8.5	8.5	6.9	8.6	8.3	8.0	7.4	8.9	7.7	8.6	8.9	7.5	1.0	10.6	4.2
CGSA-M	9.2	8.5	8.5	9.2	8.5	8.3	8.0	8.1	8.1	8.3	8.1	7.6	7.0	6.6	7.5	11.1

record the average ranking of 48 functions for all compared 16 algorithms. CGSA-P gets the smallest value of 7.47, which means that it averagely performs the best for all functions. The second smallest value 7.58 is acquired by CGSA-R, while CGSA-M gets the third one. It is worth pointing out that GSA gets the largest ranking value, indicating that all chaotic GSAs performs better than GSA. In addition, it is mostly desired that a general well-performing algorithm should be designed. From this practical problem-solving perspective, we can conclude that the proposed mul-

Table 11 Rankings of 16 variants of GSAs based on Friedman test for benchmark optimization function F33–F48.

Algorithm	F33	F34	F35	F36	F37	F38	F39	F40	F41	F42	F43	F44	F45	F46	F47	F48	Total Average
GSA	3.9	5.4	10.7	11.6	15.0	9.2	7.3	8.9	6.8	6.7	7.2	7.1	3.6	8.1	6.5	2.2	**9.23**
CGSA-1	7.5	9.3	6.8	8.4	9.1	8.3	9.6	8.0	8.9	8.4	8.5	8.5	9.3	8.4	8.9	10.3	**8.55**
CGSA-2	10.5	9.0	9.3	7.9	8.3	8.8	8.3	9.0	8.6	9.0	8.2	7.7	9.4	8.3	10.8	9.0	**8.68**
CGSA-3	9.5	10.2	8.1	7.2	8.3	9.6	10.6	10.1	9.1	8.9	9.3	8.6	9.6	9.8	8.6	7.7	**8.62**
CGSA-4	9.1	9.9	7.6	8.6	8.1	8.7	9.0	6.7	8.2	8.3	8.8	8.5	8.6	9.9	7.6	7.7	**8.55**
CGSA-5	9.7	9.7	8.9	9.0	9.4	9.4	9.1	8.0	8.6	7.2	8.2	8.5	8.1	8.5	9.2	8.5	**8.65**
CGSA-6	11.2	8.7	8.2	8.1	6.6	10.3	9.9	8.2	7.8	8.6	8.8	8.9	7.8	8.1	8.7	10.2	**8.59**
CGSA-7	9.1	8.8	9.3	8.4	7.7	8.7	7.7	7.6	10.0	8.9	8.2	9.4	10.3	8.4	8.2	9.6	**8.59**
CGSA-8	9.5	10.0	8.3	9.8	7.7	7.1	7.1	8.8	9.0	8.2	6.9	8.8	8.5	8.2	9.0	9.6	**8.66**
CGSA-9	10.0	9.0	8.4	9.0	8.3	8.8	6.9	7.7	9.1	9.2	8.6	8.2	7.3	8.8	7.6	9.7	**8.65**
CGSA-10	9.0	8.5	9.6	8.1	7.5	6.6	7.4	8.5	8.9	7.7	7.0	10.3	8.2	8.8	8.1	9.5	**8.41**
CGSA-11	8.4	9.8	9.0	8.6	8.9	8.5	7.3	7.9	8.2	10.3	10.4	7.1	9.7	7.9	9.3	9.9	**8.83**
CGSA-12	10.5	8.4	7.8	7.8	8.2	8.0	8.2	9.7	8.0	9.5	9.4	9.1	9.7	8.7	9.2	9.5	**8.60**
CGSA-R	4.0	4.5	7.4	7.1	8.0	8.5	9.6	7.8	8.7	9.3	7.0	8.9	11.4	7.4	8.5	1.1	**7.58**
CGSA-P	3.6	5.5	9.1	7.4	8.2	5.5	9.0	10.1	6.4	6.3	9.5	7.0	3.3	8.6	6.5	12.4	**7.47**
CGSA-M	10.6	9.3	7.5	8.9	6.9	10.0	9.0	8.9	10.0	9.7	10.1	9.4	11.0	8.1	9.3	9.1	**8.37**

tiple chaos incorporation scheme is effective for improving the performance of GSA.

6. Discussions

As discussed in detail in Sect. 5, our proposed multiple chaotic embedded CGSA (MCGSA) can perform better than traditional GSA and 12 single chaos embedded CGSAs under the condition of the same maximum number of iterations. That is to say, under this condition MCGSA outperforms its competitive algorithms in terms of solution accuracy and the capacity of jumping out of local optima. However, as the computational cost in each iteration for the compared algorithms is different, it is necessary to compare these algorithms under the same computational cost. To realize this, we set the termination criteria to be the maximum number of function evaluations (i.e., D*10000) for all algorithms.

On the other hand, to further verify the effect of the usage of chaos, we perform a contrast analysis to answer the following question: why the chaos is effective for perturbation force of the local search. We conduct two variants of CGSA by replacing the $z(t)$ in Eq. (23) to be random numbers with a uniform distribution in [0, 1] or a standard normal distribution with mean 0 and variance 1. We designate the newly conducted algorithms to be GSA-UD (i.e., the GSA using a uniform distribution random sequence embedded local search) and GSA-ND (i.e., the GSA using a normal distribution random sequence embedded local search), respectively.

Table 12 summarizes the results for GSA, CGSA-R, CGSA-P, CGSA-M, GSA-UD, and GSA-ND for the all benchmark functions over 30 independent runs. Figures 4 and 5 depict the convergence graphs (average best-so-far versus number of function evaluation) for two typical functions: F5 and F30. For F13, CGSA-P and CGSA-R perform significantly better than the others. For F19, F30 and F31, although almost the same average values are obtained by CGSA-P and GSA-UD, CGSA-P possesses a smaller deviation, which suggests that CGSA-P is more robust to generate promising solutions. Similar observation can be found for F18, F20 and F47, which indicates that CGSA-R

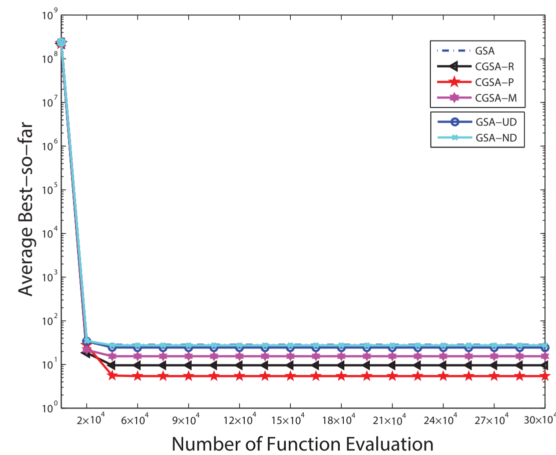

Fig. 4 Convergence graph of F5: solution along with the number of function evaluation.

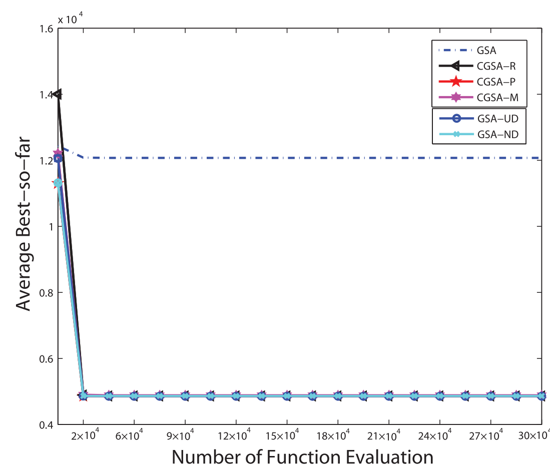

Fig. 5 Convergence graph of F30: solution along with the number of function evaluation.

is the most promising, and for F22 and F38, which shows CGSA-M is the most competitive. Exceptions can be found for F28 and F43, which shows MCGSA performs slightly worse than GSA-UD or GSA-ND. It can be concluded that MCGSA generally outperforms the compared algorithms in

Table 12 Experimental results of benchmark functions using traditional GSA, CGSA-R, CGSA-P, CGSA-M, GSA-UD and GSA-ND under the same maximum number of function evaluations.

Algorithm	F1	F2	F3	F4	F5	F6
GSA	1.30E−17 ± 4.33E−18	2.32E−08 ± 3.07E−09	2.70E+02 ± 9.52E+01	3.45E−09 ± 6.91E−10	2.83E+01 ± 1.14E+01	0.00E+00 ± 0.00E+00
CGSA-R	**1.92E−18 ± 4.68E−19**	**2.30E−08 ± 4.62E−09**	**2.19E+02 ± 8.92E+01**	1.21E−01 ± 4.80E−01	9.59E+00 ± 1.15E+01	**0.00E+00 ± 0.00E+00**
CGSA-P	2.31E−18 ± 8.88E−19	2.40E−08 ± 4.20E−09	2.29E+02 ± 8.30E+01	**3.30E−09 ± 6.37E−10**	**5.42E+00 ± 2.15E+01**	**0.00E+00 ± 0.00E+00**
CGSA-M	1.97E−18 ± 5.70E−19	2.34E−08 ± 3.51E−09	2.40E+02 ± 8.47E+01	2.56E−03 ± 1.40E−02	1.55E+01 ± 1.78E+01	**0.00E+00 ± 0.00E+00**
GSA-UD	6.84E−18 ± 2.05E−18	2.35E−08 ± 3.22E−09	2.55E+02 ± 7.11E+01	5.01E−02 ± 2.75E−01	2.45E+01 ± 4.29E+01	0.00E+00 ± 0.00E+00
GSA-ND	5.87E−18 ± 2.34E−18	2.32E−08 ± 3.31E−09	2.55E+02 ± 9.08E+01	7.03E−03 ± 3.03E−02	2.75E+01 ± 4.08E+01	0.00E+00 ± 0.00E+00
Algorithm	F7	F8	F9	F10	F11	F12
GSA	2.18E−02 ± 1.04E−02	−2.89E+03 ± 4.49E+02	1.60E+01 ± 3.84E+00	2.36E−09 ± 2.35E−10	4.46E+00 ± 2.07E+00	4.58E−02 ± 1.17E−01
CGSA-R	1.08E−02 ± 4.95E−03	**−3.23E+03 ± 5.73E+02**	**1.49E+01 ± 3.36E+00**	**1.08E−09 ± 1.36E−10**	3.67E+00 ± 1.44E+00	4.65E−02 ± 1.14E−01
CGSA-P	**1.05E−02 ± 4.81E−03**	−2.93E+03 ± 4.65E+02	1.52E+01 ± 3.56E+00	1.91E−09 ± 2.93E−10	4.05E+00 ± 1.75E+00	**3.13E−02 ± 6.20E−02**
CGSA-M	1.18E−02 ± 4.64E−03	−2.90E+03 ± 4.97E+02	1.52E+01 ± 3.99E+00	1.95E−09 ± 4.05E−10	**2.77E+00 ± 8.92E−01**	6.04E−02 ± 9.90E−02
GSA-UD	1.35E−02 ± 5.93E−03	−2.95E+03 ± 4.47E+02	1.66E+01 ± 3.82E+00	1.98E−09 ± 3.15E−10	3.91E+00 ± 1.79E+00	4.84E−02 ± 8.49E−02
GSA-ND	1.22E−02 ± 5.01E−03	−3.11E+03 ± 5.54E+02	1.65E+01 ± 3.52E+00	1.94E−09 ± 3.12E−10	4.08E+00 ± 1.63E+00	6.87E−02 ± 1.09E−01
Algorithm	F13	F14	F15	F16	F17	F18
GSA	3.66E−04 ± 2.01E−03	6.32E+00 ± 3.10E+00	6.96E−03 ± 4.75E−03	−1.03E+00 ± 4.97E−16	3.98E−01 ± 0.00E+00	3.00E+00 ± 5.89E−15
CGSA-R	6.75E−19 ± 1.86E−19	4.80E+00 ± 3.24E+00	**4.09E−03 ± 2.47E−03**	**−1.03E+00 ± 4.70E−16**	**3.98E−01 ± 0.00E+00**	**3.00E+00 ± 3.39E−15**
CGSA-P	**6.39E−19 ± 2.53E−19**	5.35E+00 ± 3.74E+00	5.11E−03 ± 3.32E−03	−1.03E+00 ± 4.79E−16	**3.98E−01 ± 0.00E+00**	3.00E+00 ± 4.07E−15
CGSA-M	6.90E−04 ± 2.63E−03	**4.40E+00 ± 3.30E+00**	4.63E−03 ± 2.62E−03	**−1.03E+00 ± 4.70E−16**	**3.98E−01 ± 0.00E+00**	3.00E+00 ± 3.59E−15
GSA-UD	3.66E−04 ± 2.01E−03	5.00E+00 ± 3.45E+00	5.42E−03 ± 2.72E−03	−1.03E+00 ± 4.70E−16	3.98E−01 ± 0.00E+00	3.00E+00 ± 4.06E−15
GSA-ND	3.66E−04 ± 2.01E−03	4.42E+00 ± 3.70E+00	4.76E−03 ± 3.27E−03	−1.03E+00 ± 4.79E−16	3.98E−01 ± 0.00E+00	3.00E+00 ± 4.29E−15
Algorithm	F19	F20	F21	F22	F23	F24
GSA	−3.86E+00 ± 3.17E−03	−3.32E+00 ± 1.49E−15	−6.42E+00 ± 3.80E+00	−1.01E+01 ± 1.14E−15	−1.05E+01 ± 2.68E−15	1.69E+03 ± 8.03E+02
CGSA-R	−3.86E+00 ± 2.47E−03	**−3.32E+00 ± 1.49E−15**	−6.03E+00 ± 3.71E+00	−1.01E+01 ± 1.39E+00	**−1.06E+01 ± 1.32E+00**	**1.10E+03 ± 6.69E+02**
CGSA-P	**−3.86E+00 ± 1.03E−03**	−3.32E+00 ± 1.54E−15	−6.75E+00 ± 3.72E+00	−1.02E+01 ± 1.22E+00	−1.00E+01 ± 1.61E+00	1.30E+03 ± 8.96E+02
CGSA-M	−3.86E+00 ± 3.24E−03	−3.32E+00 ± 2.57E−02	**−7.93E+00 ± 3.46E+00**	**−1.02E+01 ± 7.49E−01**	−9.87E+00 ± 2.05E+00	1.67E+03 ± 9.01E+02
GSA-UD	−3.86E+00 ± 2.30E−03	−3.32E+00 ± 1.57E−15	−7.30E+00 ± 3.58E+00	−1.02E+01 ± 7.71E−01	−1.03E+01 ± 1.48E+00	1.26E+03 ± 7.12E+02
GSA-ND	−3.86E+00 ± 2.59E−03	−3.32E+00 ± 1.57E−15	−7.01E+00 ± 2.96E+00	−1.02E+01 ± 7.92E−01	−1.00E+01 ± 1.67E+00	1.43E+03 ± 6.54E+02
Algorithm	F25	F26	F27	F28	F29	F30
GSA	2.02E+04 ± 2.14E+03	4.54E+07 ± 5.13E+07	6.60E+04 ± 1.39E+04	2.29E+04 ± 2.80E+03	1.13E+08 ± 5.02E+07	1.21E+04 ± 2.51E+02
CGSA-R	1.99E+04 ± 2.19E+03	3.09E+07 ± 2.72E+07	6.29E+04 ± 9.66E+03	2.25E+04 ± 2.40E+03	1.11E+08 ± 6.50E+07	4.93E+03 ± 9.86E+01
CGSA-P	**1.97E+04 ± 2.05E+03**	4.63E+07 ± 4.96E+07	**6.22E+04 ± 1.20E+04**	2.28E+04 ± 2.46E+03	**1.08E+08 ± 5.79E+07**	**4.86E+03 ± 1.17E+01**
CGSA-M	2.04E+04 ± 2.16E+03	**2.92E+07 ± 2.08E+07**	6.29E+04 ± 1.09E+04	2.24E+04 ± 2.16E+03	1.36E+08 ± 6.82E+07	4.88E+03 ± 5.36E+01
GSA-UD	2.03E+04 ± 2.48E+03	4.84E+07 ± 4.60E+07	6.56E+04 ± 1.19E+04	**2.23E+04 ± 2.11E+03**	1.13E+08 ± 5.00E+07	4.86E+03 ± 2.84E+01
GSA-ND	2.01E+04 ± 1.78E+03	3.49E+07 ± 2.26E+07	7.02E+04 ± 1.42E+04	2.35E+04 ± 2.12E+03	1.25E+08 ± 7.60E+07	4.86E+03 ± 5.81E+01
Algorithm	F31	F32	F33	F34	F35	F36
GSA	2.03E+01 ± 9.36E−02	3.92E+01 ± 6.07E+00	3.53E+01 ± 7.74E+00	1.86E+00 ± 1.38E+00	3.01E+03 ± 3.94E+03	5.55E+00 ± 1.23E+00
CGSA-R	2.03E+01 ± 1.06E−01	4.15E+01 ± 8.41E+00	**3.26E+01 ± 7.98E+00**	1.30E+00 ± 1.22E+00	2.46E+03 ± 2.62E+03	5.39E+00 ± 1.40E+00
CGSA-P	**2.03E+01 ± 7.79E−02**	**4.05E+01 ± 8.96E+00**	3.56E+01 ± 5.81E+00	1.86E+00 ± 1.17E+00	5.15E+03 ± 6.84E+03	5.75E+00 ± 1.30E+00
CGSA-M	2.03E+01 ± 9.36E−02	4.25E+01 ± 7.67E+00	3.57E+01 ± 5.67E+00	**1.22E+00 ± 1.31E+00**	**2.04E+03 ± 3.00E+03**	**5.28E+00 ± 1.58E+00**
GSA-UD	2.03E+01 ± 9.15E−02	4.14E+01 ± 6.20E+00	3.39E+01 ± 6.42E+00	1.58E+00 ± 1.40E+00	2.48E+03 ± 3.36E+03	6.14E+00 ± 1.34E+00
GSA-ND	2.03E+01 ± 8.02E−02	4.10E+01 ± 6.62E+00	3.33E+01 ± 6.13E+00	2.38E+00 ± 1.14E+00	5.31E+03 ± 8.08E+03	5.61E+00 ± 1.32E+00
Algorithm	F37	F38	F39	F40	F41	F42
GSA	1.42E+01 ± 1.29E−01	3.09E+02 ± 3.74E+01	2.94E+02 ± 2.12E+02	3.35E+02 ± 2.58E+02	9.68E+02 ± 6.15E+01	9.59E+02 ± 4.44E+01
CGSA-R	1.41E+01 ± 1.59E−01	3.08E+02 ± 2.57E+01	2.31E+02 ± 2.08E+02	2.58E+02 ± 2.06E+02	9.61E+02 ± 4.55E+01	**9.58E+02 ± 4.58E+01**
CGSA-P	1.41E+01 ± 1.99E−01	3.07E+02 ± 2.54E+01	2.65E+02 ± 2.12E+02	**2.08E+02 ± 1.75E+02**	9.63E+02 ± 5.68E+01	9.59E+02 ± 1.42E+07
CGSA-M	**1.40E+01 ± 2.71E−01**	**3.04E+02 ± 1.83E+01**	**1.76E+02 ± 2.16E+02**	2.21E+02 ± 1.62E+02	**9.52E+02 ± 6.22E+01**	9.61E+02 ± 4.65E+01
GSA-UD	1.41E+01 ± 2.26E−01	3.04E+02 ± 1.83E+01	2.69E+02 ± 1.86E+02	2.38E+02 ± 1.95E+02	9.59E+02 ± 6.89E+01	9.62E+02 ± 4.32E+01
GSA-ND	1.41E+01 ± 2.19E−01	3.09E+02 ± 3.76E+01	2.95E+02 ± 2.17E+02	2.42E+02 ± 1.87E+02	9.78E+02 ± 1.65E+01	9.67E+02 ± 1.71E+01
Algorithm	F43	F44	F45	F46	F47	F48
GSA	9.67E+02 ± 3.48E+01	8.24E+02 ± 3.29E+02	9.32E+02 ± 2.21E+01	8.38E+02 ± 3.09E+02	2.00E+02 ± 7.96E−13	1.68E+03 ± 8.44E+00
CGSA-R	9.65E+02 ± 3.60E+01	7.17E+02 ± 3.12E+02	**9.29E+02 ± 1.91E+01**	8.61E+02 ± 2.96E+02	**2.00E+02 ± 1.16E+00**	1.68E+03 ± 1.10E+01
CGSA-P	9.66E+02 ± 3.35E+01	**6.74E+02 ± 2.94E+02**	9.36E+02 ± 1.88E+01	8.56E+02 ± 2.95E+02	3.09E+02 ± 2.85E+02	**1.67E+03 ± 9.35E+00**
CGSA-M	9.61E+02 ± 5.61E+01	7.38E+02 ± 3.18E+02	9.31E+02 ± 1.95E+01	**6.97E+02 ± 2.70E+02**	2.56E+02 ± 2.13E+02	1.68E+03 ± 8.30E+00
GSA-UD	9.55E+02 ± 5.41E+01	7.17E+02 ± 3.12E+02	9.29E+02 ± 1.41E+01	8.00E+02 ± 2.93E+02	2.00E+02 ± 5.73E−13	1.68E+03 ± 6.75E+00
GSA-ND	**9.51E+02 ± 5.24E+01**	8.05E+02 ± 3.25E+02	9.29E+02 ± 1.90E+01	7.94E+02 ± 2.94E+02	2.63E+02 ± 2.40E+02	1.68E+03 ± 1.12E+01

terms of solution accuracy under almost the same computational burden for most tested benchmark functions (46 out of 48 functions). In addition, it should be pointed out that the solutions in Table 12 are generally better than those in Tables 1–8 because more iteration numbers are implemented in this complementary experiment (e.g. 6000 iterations are carried out for GSA). All in all, we can conclude that: (1) the local search induced by chaos is more efficient than that by random or normal distribution numbers; (2) multiple chaos embedded local search generally performs better than single chaos embedded one; and (3) the parallelly embedding strategy is the most effective for improving the performance of GSA.

7. Conclusions

In this paper, taking into account the abundant searching dynamics of different chaos we innovatively propose a multiple chaos embedded gravitational search algorithm (MCGSA). To further improve the searching performance of GSA, three kinds of incorporation schemes are investigated. Multiple chaotic maps are randomly, parallelly, or memory-selectively incorporated into GSA, respectively. Experimental results based on a set of 48 benchmark optimization functions verify the effectiveness and robustness of the proposed MCGSA. Especially, the parallelly embedding scheme for GSA is demonstrated to be the most effective based on the Friedman test. This study opens the door to the following future researches:

1. MCGSA should be verified on other practical problems, especially engineering optimization problems.
2. The effectiveness of the proposed multiple chaos incorporation scheme should be applied on other metaheuristics to further reveal its effects.
3. Through our experimental results, we find that a certain chaotic map is effective for some specific optimization functions. The influence of the distinct chaotic search dynamics on the algorithm should be further studied.
4. As each chaotic map has an inherent Lyapunov exponent which reflects its chaotic degree, a Lyapunov exponent based adaptive multiple chaos incorporation scheme should also be designed.

Acknowledgment

This research was partially supported by JSPS KAKENHI Grant No. 15K00332 (Japan).

References

[1] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," ACM Computing Surveys (CSUR), vol.35, no.3, pp.268–308, 2003.
[2] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," Information Sciences, vol.237, pp.82–117, 2013.
[3] X. Yao and Y. Xu, "Recent advances in evolutionary computation," Journal of Computer Science and Technology, vol.21, no.1, pp.1–18, 2006.
[4] K. Kameyama, "Particle swarm optimization — A survey," IEICE Trans. Inf. & Syst., vol.E92-D, no.7, pp.1354–1361, July 2009.
[5] S. Gao, Y. Wang, J. Cheng, Y. Inazumi, and Z. Tang, "Ant colony optimization with clustering for solving the dynamic location routing problem," Applied Mathematics and Computation, vol.285, pp.149–173, 2016.
[6] S. Wang, S. Gao, Aorigele, Y. Todo, and Z. Tang, "A multi-learning immune algorithm for numerical optimization," IEICE Trans. Fundamentals, vol.E98-A, no.1, pp.362–377, Jan. 2015.
[7] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A gravitational search algorithm," Information Sciences, vol.179, no.13, pp.2232–2248, 2009.
[8] W. Zhang, P. Niu, G. Li, and P. Li, "Forecasting of turbine heat rate with online least squares support vector machine based on gravitational search algorithm," Knowledge-Based Systems, vol.39, pp.34–44, 2013.
[9] B. Ji, X. Yuan, Z. Chen, and H. Tian, "Improved gravitational search algorithm for unit commitment considering uncertainty of wind power," Energy, vol.67, pp.52–62, 2014.
[10] C. Lopez-Molina, H. Bustince, J. Fernandez, P. Couto, and B. De Baets, "A gravitational approach to edge detection based on triangular norms," Pattern Recognit., vol.43, no.11, pp.3730–3741, 2010.
[11] C. Li, J. Zhou, B. Fu, P. Kou, and J. Xiao, "T–s fuzzy model identification with a gravitational search-based hyperplane clustering algorithm," IEEE Trans. Fuzzy Syst., vol.20, no.2, pp.305–317, 2012.
[12] N.M. Sabri, M. Puteh, and M.R. Mahmood, "A review of gravitational search algorithm," Int. J. Advance. Soft Comput. Appl., vol.5, no.3, pp.1–39, 2013.
[13] C. Li and J. Zhou, "Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm," Energy Conversion and Management, vol.52, no.1, pp.374–381, 2011.
[14] B. Shaw, V. Mukherjee, and S. Ghoshal, "A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems," International Journal of Electrical Power & Energy Systems, vol.35, no.1, pp.21–33, 2012.
[15] S. Yazdani, H. Nezamabadi-pour, and S. Kamyab, "A gravitational search algorithm for multimodal optimization," Swarm and Evolutionary Computation, vol.14, pp.1–14, 2014.
[16] 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, 2003.
[17] T. Xiang, X. Liao, and K.-w. Wong, "An improved particle swarm optimization algorithm combined with piecewise linear chaotic map," Applied Mathematics and Computation, vol.190, no.2, pp.1637–1645, 2007.
[18] S. Saremi, S. Mirjalili, and A. Lewis, "Biogeography-based optimisation with chaos," Neural Computing and Applications, vol.25, no.5, pp.1077–1097, 2014.
[19] A.A. Heidari, R.A. Abbaspour, and A.R. Jordehi, "An efficient chaotic water cycle algorithm for optimization tasks," Neural Computing and Applications, vol.28, no.1, pp.57–85, 2017.
[20] M. Mitić, N. Vuković, M. Petrović, and Z. Miljković, "Chaotic fruit fly optimization algorithm," Knowledge-Based Systems, vol.89, pp.446–458, 2015.
[21] G.-G. Wang, L. Guo, A.H. Gandomi, G.S. Hao, and H. Wang, "Chaotic Krill Herd algorithm," Information Sciences, vol.274, pp.17–34, 2014.
[22] A.H. Gandomi and X.-S. Yang, "Chaotic bat algorithm," Journal of Computational Science, vol.5, no.2, pp.224–232, 2014.
[23] D. Jia, G. Zheng, and M.K. Khan, "An effective memetic differential evolution algorithm based on chaotic local search," Information Sciences, vol.181, no.15, pp.3175–3187, 2011.
[24] R. Arul, G. Ravi, and S. Velusami, "Chaotic self-adaptive differential harmony search algorithm based dynamic economic dispatch," International Journal of Electrical Power & Energy Systems, vol.50, pp.85–96, 2013.
[25] A. Kazem, E. Sharifi, F.K. Hussain, M. Saberi, and O.K. Hussain, "Support vector regression with chaos-based firefly algorithm for stock market price forecasting," Applied Soft Computing, vol.13, no.2, pp.947–958, 2013.
[26] Y. Li, Q. Wen, L. Li, and H. Peng, "Hybrid chaotic ant swarm optimization," Chaos, Solitons & Fractals, vol.42, no.2, pp.880–889, 2009.
[27] S. Talatahari, B.F. Azar, R. Sheikholeslami, and A.H. Gandomi, "Imperialist competitive algorithm combined with chaos for global optimization," Communications in Nonlinear Science and Numerical Simulation, vol.17, no.3, pp.1312–1319, 2012.
[28] S. Gao, C. Vairappan, Y. Wang, Q. Cao, and Z. Tang, "Gravitational search algorithm combined with chaos for unconstrained numerical optimization," Applied Mathematics and Computation, vol.231, pp.48–62, 2014.
[29] D. Shen, T. Jiang, W. Chen, Q. Shi, and S. Gao, "Improved chaotic gravitational search algorithms for global optimization," IEEE Congress on Evolutionary Computation (CEC), pp.1220–1226, 2015.
[30] J. Gleick, Chaos: Making a New Science (Enhanced Edition), Open Road Media, 2011.
[31] M.S. Tavazoei and M. Haeri, "Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms," Applied Mathematics and Computation, vol.187, no.2, pp.1076–1085, 2007.
[32] Y.Y. He, J.Z. Zhou, X.Q. Xiang, H. Chen, and H. Qin, "Comparison of different chaotic maps in particle swarm optimization algorithm for long-term cascaded hydroelectric system scheduling," Chaos, Solitons & Fractals, vol.42, no.5, pp.3169–3176, 2009.
[33] D. Yang, Z. Liu, and J. Zhou, "Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization," Communications in Nonlinear Science and Numerical Simulation, vol.19, no.4, pp.1229–1246, 2014.
[34] A.K. Qin, V.L. Huang, and P.N. Suganthan, "Differential evolution

algorithm with strategy adaptation for global numerical optimization," IEEE Trans. Evol. Comput., vol.13, no.2, pp.398–417, 2009.
[35] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," IEEE Trans. Evol. Comput., vol.3, no.2, pp.82–102, 1999.
[36] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," KanGAL report, vol.2005005, p.2005, 2005.
[37] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," Information Sciences, vol.180, no.10, pp.2044–2064, 2010.
[38] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, and F. Herrera, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," Soft Computing, vol.13, no.3, pp.307–318, 2009.
[39] D.H. Wolpert and W.G. Macready, "No free lunch theorems for optimization," IEEE Trans. Evol. Comput., vol.1, no.1, pp.67–82, 1997.

Zhenyu Song received his B.S. degree in Nanjing University of Information Science & Technology, Nanjing, China in 2009. Now, he is working toward the D.E. degree at University of Toyama, Toyama, Japan. His main research interests are evolutionary algorithms, artificial neural networks and multi-objective optimization problems.

Shangce Gao received the Ph.D. degree in innovative life science from University of Toyama, Toyama, Japan, in 2011. He is currently an Associate Professor with the Faculty of Engineering, University of Toyama, Toyama, Japan. From 2011 to 2012, he was an Associate Research Fellow with the Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai, China. From 2012 to 2014, he was an Associate Professor with the College of Information Sciences and Technology, Donghua University, Shanghai, China. His main research interests include mobile computing, computational intelligence, nature-inspired technologies, swarm intelligence, and neural networks for optimization and real-world applications. He was a recipient of the Shanghai Rising-Star Scientist award, the Chen-Guang Scholar of Shanghai award, the Outstanding Academic Performance Award of IEICE, the Outstanding Self-financed Students Abroad Award of Chinese Government, and the Outstanding Academic Achievement Award of IPSJ. He is a Senior Member of IEEE.

Yang Yu received the B.E. degree in Civil Engineering from the Yancheng Institute of Technology, Yancheng, China, in 2014. He is currently pursuing the M.S. degree at University of Toyama, Toyama, Japan. His main research interests are data mining, optimization problems and nature-inspired algorithms.

Jian Sun received the B.S. degree from Nanjing Normal University, Nanjing, China in 2000 and an M.S. degree from Yangzhou University, Yangzhou, China, in 2010. Now she is working toward the D.E. degree at University of Toyama, Toyama, Japan. Her main research interests are computational intelligence, multiple-valued logic and artificial neural networks.

Yuki Todo received the B.S. degree from Zhejiang University, Zhejiang, China, the M.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, and the D.E. degree from Kanazawa University, Kanazawa, Japan, in 1983, 1986, and 2005, respectively. From 1987 to 1989, she was an Assistant Professor with the Institute of Microelectronics, Shanghai Jiaotong University, Shanghai, China. From 1989 to 1990, she was a research student at Nagoya University, Nagoya, Japan. From 1990 to 2000, she was a Senior Engineer with Sanwa Newtech Inc., Miyazaki, Japan. From 2000 to 2011, she worked with Tateyama Systems Institute, Toyama, Japan. In 2012, she joined Kanazawa University, where she is now an Associate Professor with the Faculty of Electrical and Computer Engineering. Her current research interests include multiple-valued logic, neural networks, and optimization.