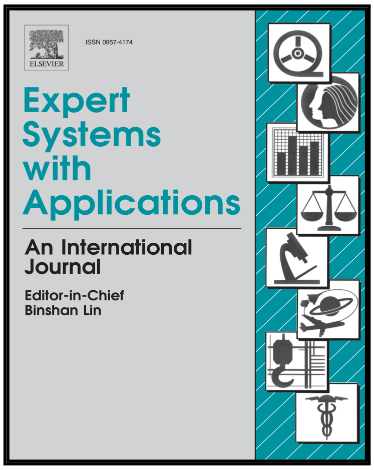
Journal Pre-proof



SCGSA: A Sine Chaotic Gravitational Search Algorithm for Continuous Optimization Problems

Jianhua Jiang, Ran Jiang, Xianqiu Meng, Keqin Li

PII: S0957-4174(19)30835-8
DOI: https://doi.org/10.1016/j.eswa.2019.113118
Reference: ESWA 113118

To appear in: *Expert Systems With Applications*

Received date: 29 July 2019
Revised date: 30 November 2019
Accepted date: 2 December 2019

Please cite this article as: Jianhua Jiang, Ran Jiang, Xianqiu Meng, Keqin Li, SCGSA: A Sine Chaotic Gravitational Search Algorithm for Continuous Optimization Problems, *Expert Systems With Applications* (2019), doi: https://doi.org/10.1016/j.eswa.2019.113118

Highlights

- SCGSA is inspired by SCA(sine cosine algorithm).
- Sine moving pattern and k are designed to balance exploration and exploitation.
- SCGSA solves the problem that CGSA is prone to suffer from local optima.
- SCGSA performs well in high dimension.
- The result is superior to various well-known algorithms in optimization domain.

SCGSA: A Sine Chaotic Gravitational Search Algorithm for Continuous Optimization Problems☆

Jianhua Jiang[a,b,*], Ran Jiang[a], Xianqiu Meng[a], Keqin Li[c]

[a]*Department of Data Science, Jilin University of Finance and Economics,Changchun 130117, P. R. China*
[b]*Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun, 130012, P. R. China*
[c]*Department of Computer Science, State University of New York, New Paltz, NY 12561, USA*

Abstract

Gravitational search algorithm (GSA), as one of the novel meta-heuristic optimization algorithms inspired by the law of gravity and mass interactions, is however prone to local optima stagnation due to heavier gravity. Hence, an enhanced version, chaotic gravitational constants for the gravitational search algorithm (CGSA), was proposed to improve the exploration ability through various chaotic maps. In this paper, with insightful utilization of sine cosine algorithm, we put forward *sine chaotic gravitational search algorithm* (SCGSA) as a further step of CGSA to escape from its local optima stagnation. The experiments show remarkable results in both the speed of convergence and the ability of finding global optima in 30 benchmark functions (CEC 2014), thus proving a better balance between exploration and exploitation in SCGSA compared with CGSA.

Keywords: Gravitational search algorithm, chaotic maps, sine cosine algorithm, continuous optimization problem

☆The authors are grateful to the financial support by the National Natural Science Foundation of China (no. 61572225), Natural Science Foundation of the Science and Technology Department of Jilin Province, China (no. 20180101044JC), the Social Science Foundation of Jilin Province, China (nos. 2019B68, 2017BS28), the Foundation of the Education Department of Jilin Province, China (nos. JJKH20180465KJ, JJKH20190111KJ) and the Foundation of Jilin University of Finance and Economics (no. 2018Z05).

*corresponding author.
Email address: jianhuajiang@yahoo.com(Jianhua Jiang),jrsword@outlook.com(Ran Jiang), 1134430458@qq.com(Xianqiu Meng),lik@newpaltz.edu (Keqin Li)

1. Introduction

Optimization problems can be found in a number of fields such as software design (Palanikkumar et al., 2012), medicine (Jobin et al., 2017), engineering (Mallick et al., 2013; Rodriguez et al., 2015; Jiang et al., 2017), business (Kumar et al., 2013; Jiang et al., 2019b,c), data mining (Jiang et al., 2018a,b, 2019a), and so forth. Over the last few decades, a growing interest can be seen in algorithms inspired by the observation of natural phenomena, and more recently, close attention has been paid to meta-heuristic optimization algorithms, which considerably improve the flexibility of heuristic algorithms as well as provide a feasible solution to the problem at an acceptable cost (both in time and space). Moreover, meta-heuristic algorithms adopt the method of generating random numbers, considering optimization problems as black boxes (Mirjalili, 2016), and such paradigms only alter the inputs and monitor the outputs of the system in maximizing or minimizing its outputs, which means that the deviation degree of the feasible solution from the optimal solution does not necessarily need to be predicted in advance. Therefore, this kind of algorithms is readily applicable to problems in various fields (Mirjalili, 2016).

Meta-heuristic algorithms can be divided into two categories: individual-based and population-based (Mirjalili, 2016). Individual-based algorithms include simulated annealing (SA) (Metropolis et al., 1953; Kirkpatrick et al., 1983), tabu search (TS) (Glover, 1990) and so forth. Population-based algorithms include genetic algorithm (GA) (Simpson et al., 1994), bat algorithm (BA) (Yang & Gandomi, 2012), ant colony optimization (ACO) (Blum, 2005), particle swarm optimization (PSO) (Kennedy, 2011), firefly algorithm (FA) (Yang, 2009), grey wolf optimizer (GWO) (Mirjalili et al., 2014), gravitational search algorithm (GSA) (Chen et al., 2011), teaching-learning-based optimization (TLBO) (Rao et al., 2011) and so forth. The superiority of individual-based algorithms is the low demand for the number of function evaluations since a single solution only requires one set of fitnesses. However, such single candidate solutions are easily trapped in the local optima close to the global optimum, which seldom happens in population-based algorithms (Mirjalili, 2015).

GSA is a population-based meta-heuristic algorithm(Rashedi et al., 2009). Inspired by the law of gravity and Newton's second law, it has been cited for thousands of times. In addition, chaotic gravitational constants for the gravitational search algorithm (CGSA) is one of the novel algorithms based on gravitational search algorithm (GSA) , which performs well with its chaotic maps. Hence, it is chosen as the basis of our enhancement. However, CGSA underperforms in part of CEC 2014 benchmark functions especially in composite functions, which is often the case in the real world. For this reason, our main motivation is to implement a further innovation. Sine chaotic gravitational search algorithm (SCGSA) redefines the movement pattern for each agent to the next position, and it keeps a better balance between exploration and exploitation than former algorithms.

This paper gives some insights into the role of velocity. There are two main hypotheses for SCGSA.

- Hypothesis 1: k is able to adjust the degree of influence by other agents so as to improve exploration ability.
- Hypothesis 2: The larger possibilities for v to multiple higher random values within $[0, 1]$ from sine function give v a bigger weight in v_{t+1}, enabling search agents to converge faster.

This paper is organized as follows: the related theories and principles of the basic SCGSA is depicted and explained in the Section 2; the mechanism of SCGSA is illustrated in Section 3; the effects of parameter k and sine function are shown in Section 4; the features, advantage and limitations are discussed in Section 5; some real examples are shown in Section 6; finally, the conclusion and the expectation of future improvement are elaborated in Section 7.

2. Related works

2.1. Gravitational search algorithm

The term "gravitational force" was put forward by Newton in his mathematical treatise *Principia* in 1687, which illustrates that every two objects in nature are attracted to each other, and the intensity of force is positively correlated to the product of the mass of these two objects, while is negatively correlated to the distance between them (Holliday et al., 1972).

Inspired by this law, a famous population-based meta-heuristic called GSA was put forward. In GSA, each agent possesses four specifications: position, inertial mass, active gravitational mass and passive gravitational mass (Holliday et al., 1972), where every individual in the masses needs to present its solution and then update the value of gravity and velocity, as shown in the processes below:

First, suppose there are a set of masses shown in Eq. 1:

$$X_i = (x_i^1, ..., x_i^d, ..., x_i^n), i = 1, 2, 3, ...N \tag{1}$$

where N denotes the number of masses, n the number of variables, and x_i^d is x_i in the d_{th} dimension.

Next, calculate the fitness of every mass in each dimension. After that, record both the best and the worst solutions in the iteration. In GSA, the gravity of each mass can be calculated by the following Eqs. 2 and 3 (Rashedi et al., 2009).

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{2}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{3}$$

where t means the current iteration, i and j the solution of i_{th} and j_{th}, fit_i the fitness of a certain mass in X_i, *worst* and *best* represent the worst and the best fitness in X_i respectively.

Similar to Newton's gravitation theory, the forces between the masses in Eq. 4 are calculated as follows:

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

where M_{pi} is the passive gravitational mass of x_i, M_{aj} the active gravitational mass of x_j, $G(t)$ a function which calculates gravitational constant, $R_{ij}(t)$ the Euclidean distance between x_i and x_j. As for ε, it is a small constant that avoids the denominator from being equal to 0.

Hence, the resultant force that acts on x_i in a dimension d is a randomly weighted sum of each component of the forces exerted by d_{th} $kbest$ agents, and it is calculated by Eq. 5.

$$F_i^d(t) = \sum_{j \in kbest, j \neq i} rand_j \times F_{ij}^d(t) \tag{5}$$

where $kbest$ indicate the first k best fitness and $rand_j$ falls in the interval of [0,1].

The G and R can be calculated respectively in the following Eqs. 6 and 7.

$$G(t) = G_0 \times e^{(-\alpha \frac{t}{T})} \tag{6}$$

$$R_{ij}(t) = ||X_i(t), X_j(t)||_2 \tag{7}$$

where G_0 is the initial value for the gravitational constant, α is a coefficient, e is a constant and T means the total number of iterations.

According to the *Newton's second law of motion-force and acceleration*, $F = ma$. Thus, acceleration can be calculated in Eq. 8.

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

where M_{ii} is the inertial mass of agent i.

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{9}$$

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

where *t+1* means the next iteration.

In the GSA, it is obvious that masses will become heavier and heavier over iterations and finally stop (Chen et al., 2011). The best individual posses the biggest mass with the least tendency to displace and will attract each population to reach maximal acceleration, thus generating better results.

However, a brilliant algorithm should balance exploration and exploitation well (Fister et al., 2013). Although the GSA makes some modifications to initial gravitational equations for better optima, owing to the heavier gravity over times of iteration, it is hard to explore new regions of the search space and to exploit the best solution in the last phase.

2.2. Chaotic gravitational constants for the gravitational search algorithm

CGSA is an enhanced version of GSA, and its main contribution is the introduction of chaotic maps. In former GSA, G (gravitational constant) represents the intensity of the total gravitational forces, and the value of G will shrink remarkably by the lapse of time, leading to a significant influence on the balance between exploration and exploitation (Mirjalili & Gandomi, 2017). Therefore, Mirjalili introduced a hybrid of chaotic maps into GSA from chaos theories to deal with the weak exploratory of GSA (Mirjalili & Gandomi, 2017) by changing the value of G drastically.

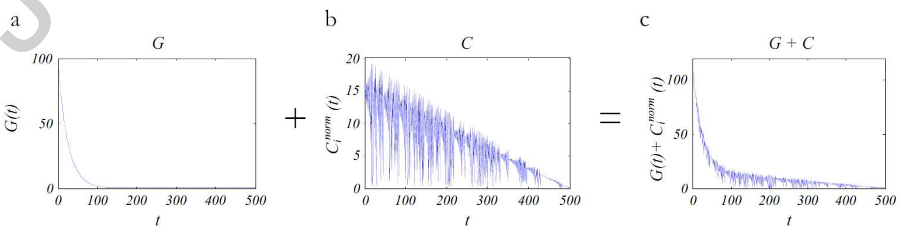


Figure 1: The procedure of embedding chaotic maps into G (single map) (Mirjalili & Gandomi, 2017)

Figure 1 depicts one of the procedures of embedding chaotic maps into G.

To refine G, the range of normalization is decreased proportionally to the iterations in Eq. 11:

$$V(t) = MAX - \frac{t}{T}(MAX - MIN) \tag{11}$$

where T is the maximum number of iterations, [MAX,MIN] represents the adaptive interval.

Normalize ($C_i(t)$ from [a,b] to [0,V(t)])

$$C_i^{norm}(t) = \frac{(C_i(t) - a) \times V(t)}{b - a} \tag{12}$$

where i represents the index of chaotic map in Appendix 8.1.

After that, combine C_i^{norm} with initial $G(t)$ as Eq. 13.

$$G(t) = C_i^{norm}(t) + G_0 \times e^{(-\alpha \frac{t}{T})} \tag{13}$$

Figure 2 illustrates these steps.

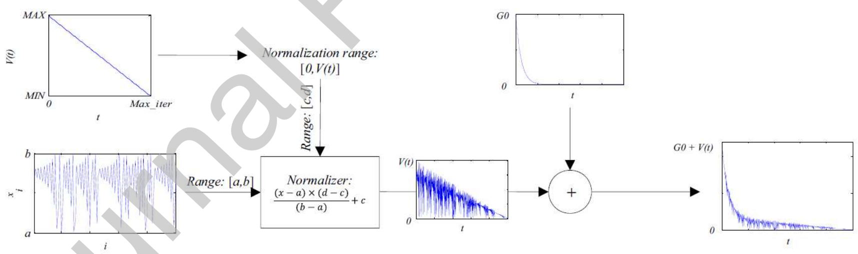


Figure 2: The steps of the proposed adaptive normalization process (Mirjalili & Gandomi, 2017)

By altering the value of G, CGSA assists the search agents in enhancing the exploratory ability. On top of this, its flexibility allows users to choose chaotic maps largely contingent on the situations.

However, it doesn't consider the inherent factors which may influence the overall performance of the algorithm. Hence, it's an innovative attempt to alter internal parameters to yield better results.

2.3. Sine cosine algorithm

Sine cosine algorithm is a novel population-based optimization algorithm (Mirjalili, 2016) from which SCGSA gains many insights ideas. Due to its convenience and simplicity, it is adopted widely in various fields such as feature selection. The movement function in this algorithm is shown in Eq. 14:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times sin(r_2) \times |r_3 P_i^t - X_i^t| & r_4 < 0.5 \\ X_i^t + r_1 \times cos(r_2) \times |r_3 P_i^t - X_i^t| & r_4 \geq 0.5 \end{cases} \tag{14}$$

where r_1 dictates the regions of the next position, r_2 defines how far the movement should be towards or outwards the destination, r_3 gives stochastic weights for destination, r_4 is a random number in [0,1], P means destination, and X represents solution.

Equation 15 is the specific function of r_1.

$$r_1 = \alpha - t \times \frac{\alpha}{T} \tag{15}$$

where T represents the maximum number of iterations and α is a constant. Obviously, as time goes by, r_1 will diminish (Fig. 3), as shown in the pattern below (α=2).

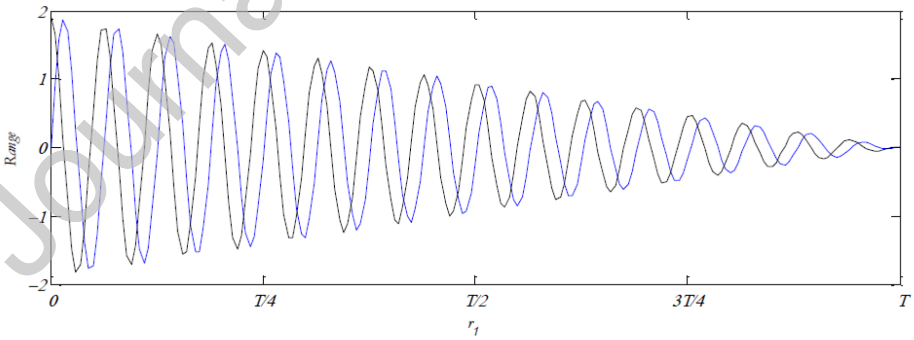


Figure 3: The range of sine and cosine ($\alpha = 2$) (Mirjalili, 2016)

3. Sine chaotic gravitational constants for the gravitational search algorithm

We put forward SCGSA in an attempt to deal with the aforementioned problems of CGSA, $v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t)$. Our first hypothesis is that a more centralized distribution of possibilities can improve convergence speed remarkably compared with the equal distribution of probable parameters of present v in $[0,1]$ in CGSA for each iteration. Inspired by SCA (Mirjalili, 2016) and Gaussian distribution, this paper changes direct random numbers to $sin(random_i \times \pi)$ in order to alter the distributional probabilities and allocate more time to seek for solutions in better positions. The Fig. 6 is an example when the peak reaches the value of 1. Apparently, 6, numbers closer to $\pi/2$ are more likely to be adopted according to the pattern of distribution.

Secondly, inspired by the normalizer in SCA algorithm (Mirjalili, 2016), we introduce a parameter k in an attempt to improve exploratory ability in the early stage, and enhance the exploitative ability in the later stage. The specific function of k is shown in Eq. 16, where k is negatively correlated with t. In the very beginning, k is greater than 1, hence, multiplying k by v and by a can enhance exploratory capability greatly in the initial phase, making it free from local optima, as solutions become closer to the global optima over time. In other words, they stand in better positions. By this time, however, it's vital to increase exploitative ability so as to seek for more exact solutions. Thus, in later stage, k is smaller than 1, which will increase convergence speed and generate better results.

$$k = 2 \times (1 - \frac{t}{T}) \tag{16}$$

where t means the present iteration, T is the maximum number of iterations. In this equation, the initial k equals 2 and finally equals 0. The Fig. 5 depicts this trend.

Some modifications are also implemented in SCGSA. In CGSA, v means the step length of a particle that moves to the next better position and a is a parameter which affects v. In another aspect, according to the equation $a = \frac{F}{m}$, a can also indicate the effect of forces exerted by other particles. Here we increase the proportion of acceleration on the basis of CGSA in order to increase the escapable capability from local optimum. This is done by adding two constants 0.5 and 2 before k. Hence, the final equation for v in SCGSA is Eq. 17.

$$V_i^d(t+1) = 0.5k \times sin(random_i \times \pi) \times V_i^d(t) + 2k \times a_i^d(t) \tag{17}$$

where $V_i^d(t+1)$ means the next velocity of x_i^d.

The pseudo code of SCGSA and total flow chart are shown as follows:

Algorithm 1 SCGSA

  Generate initialize population
  Calculate the fitness for all search agents
  while the end criterion is not satisfied **do**
    Update G by Eq. 6
    Update M, forces, accelerations
    Update k by Eq. 16
    Update V and X by Eq. 17 and 10
  end while
  return the best solution

It's worth noting that this novel algorithm doesn't downplay the chaotic maps in CGSA. Rather, it redefines the way v and a change.

On the whole, the impact of present velocity and acceleration are smaller with the elapse of time, and acceleration takes a lager proportion. The range of $0.5 \times k \times sin(random_i \times \pi)$ and $2 \times k$ are in $[0, 1]$ and $[0, 4]$ respectively so that SCGSA keeps a good balance between exploration and exploitation.

4. Experiments and results

In the field of optimization, test cases need to be employed to prove the performance of an algorithm. Thus, 30 benchmark functions are introduced in

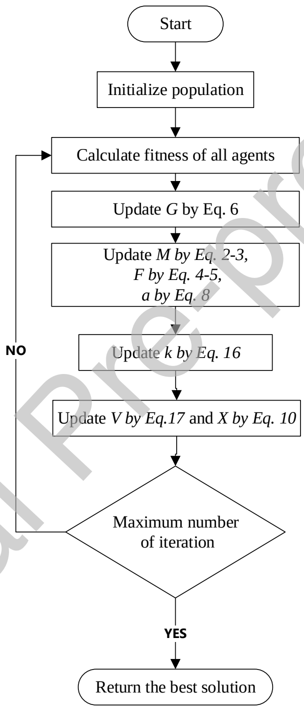


Figure 4: The flowchart of SCGSA

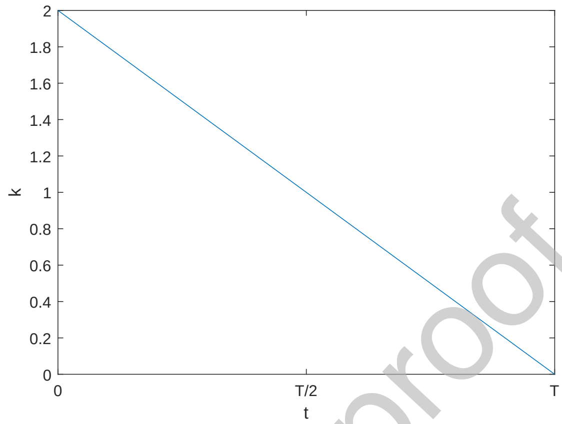


Figure 5: $k = 2 \times (1 - \frac{t}{T})$

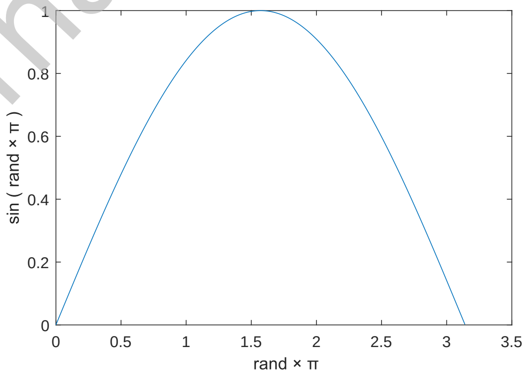


Figure 6: Probability of sine function

our study to test its efficiency. The sets of functions belong to four families: unimodal, simple multimodal, hybrid and composite test functions. Unimodal functions can test the exploitative capability while simple multimodal, hybrid and composite functions can test exploratory capability (Mirjalili, 2016). All of these functions are shown in Appendix 8.4 and 8.5.

In addition, as SCGSA is an enhanced algorithm based on CGSA and the original version of GSA, it's also crucial to compare them with one another in terms of convergence speed and final value. 100 search agents offered by this paper are allowed to determine the global optima (in this set of experiments the best solutions are minimum values) over 500 iterations. In order to pursue more reliable data, all agents in each case run 30 times and are then calculated to get their mean and standard deviation.

4.1. Experimental settings

For experiments, these variants are coded in Matlab R2018a environment under Windows 10 operating system, all simulations are running on a computer with Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40GHz and its memory is 8G.

4.2. Tests for influence of parameter k

The performance of $c \times k$ (c is a constant) is tested by CEC 2014 benchmark functions which include unimodal, multimodal, hybrid and composite test functions in order to confirm whether the performances are influenced by chance. This set of experiments compare Eq. 9 with Eq. 18 to explore the effects of $c \times k$.

$$v_i^d(t+1) = 0.5 \times k \times rand_i \times v_i^d(t) + 2 \times k \times a_i^d(t) \tag{18}$$

Table 1 depicts 30 benchmark functions in various algorithms, and this paper chooses the 7_{th} chaotic map as an example. It is obvious that the introduction of parameter k has a significant impact on seeking optima, especially in unimodal and multimodal functions. This paper adopts the method of scientific notation, and in some columns, the values are equal due to rounding, hence the optima are made bold in Table 1.

Table 1: Results for the influence of k

D=50,map=7		F1	F2	F3	F4
GSA	best	1.8194E+09	8.5930E+10	1.4664E+05	1.9896E+04
	worst	2.4676E+09	1.0029E+11	1.6436E+05	2.4611E+04
	mean	2.2087E+09	9.4816E+10	1.5519E+05	2.2839E+04
	std	3.1362E+08	6.0620E+09	6.9282E+03	1.7772E+03
CGSA	best	1.3851E+08	1.8897E+10	9.9779E+04	2.5886E+03
	worst	2.3212E+08	3.0507E+10	1.1812E+05	4.2200E+03
	mean	1.8996E+08	2.4895E+10	1.0984E+05	3.3128E+03
	std	2.8124E+07	3.4737E+09	5.6525E+03	4.4208E+02
kCGSA	best	1.2478E+07	1.6915E+03	1.1784E+05	5.7773E+02
	worst	3.8058E+07	1.3825E+04	1.6445E+05	7.3611E+02
	mean	**2.5452E+07**	**6.1408E+03**	**1.3285E+05**	**6.4660E+02**
	std	6.0299E+06	3.5171E+03	9.9093E+03	3.5631E+01
		F5	F6	F7	F8
GSA	best	5.2000E+02	6.6294E+02	1.4820E+03	1.0689E+03
	worst	5.2000E+02	6.6759E+02	1.6443E+03	1.0945E+03
	mean	5.2000E+02	6.6471E+02	1.5874E+03	1.0862E+03
	std	4.0277E-05	1.9526E+00	6.5997E+01	1.0206E+01
CGSA	best	5.2000E+02	6.5561E+02	8.7323E+02	1.0443E+03
	worst	5.2000E+02	6.6475E+02	9.8776E+02	1.0945E+03
	mean	5.2000E+02	6.5860E+02	9.4047E+02	1.0719E+03
	std	1.0250E-04	2.2556E+00	2.7270E+01	1.2827E+01
kCGSA	best	5.2000E+02	6.2292E+02	7.0000E+02	9.6914E+02
	worst	5.2000E+02	6.3606E+02	7.0001E+02	1.0368E+03
	mean	**5.2000E+02**	**6.2844E+02**	**7.0000E+02**	**1.0044E+03**
	std	1.0377E-04	2.9823E+00	4.4243E-03	1.6632E+01
		F9	F10	F11	F12
GSA	best	1.2426E+03	8.0352E+03	9.8726E+03	1.2000E+03
	worst	1.3437E+03	9.1210E+03	1.1261E+04	1.2001E+03
	mean	1.2749E+03	8.5798E+03	1.0368E+04	1.2001E+03
	std	3.9854E+01	4.2673E+02	5.7023E+02	2.4820E-02
CGSA	best	1.2273E+03	7.5234E+03	8.7877E+03	1.2000E+03
	worst	1.3020E+03	9.8496E+03	1.1193E+04	1.2000E+03
	mean	1.2641E+03	8.8029E+03	9.8660E+03	1.2000E+03
	std	1.6994E+01	6.2604E+02	6.9619E+02	9.0683E-03
kCGSA	best	1.0512E+03	6.6143E+03	6.5150E+03	1.2000E+03
	worst	1.1448E+03	9.0168E+03	9.2349E+03	1.2000E+03
	mean	**1.0883E+03**	**7.7149E+03**	**7.9360E+03**	**1.2000E+03**
	std	1.9523E+01	5.8197E+02	7.1192E+02	1.9280E-03
		F13	F14	F15	F16
GSA	best	1.3060E+03	1.6018E+03	5.3466E+05	1.6222E+03
	worst	1.3065E+03	1.6402E+03	8.3227E+05	1.6229E+03
	mean	1.3063E+03	1.6140E+03	6.2765E+05	1.6226E+03
	std	1.9967E-01	1.5353E+01	1.2079E+05	2.4875E-01
CGSA	best	1.3027E+03	1.4324E+03	3.7367E+03	1.6219E+03
	worst	1.3038E+03	1.4637E+03	1.3430E+04	1.6232E+03
	mean	1.3032E+03	1.4474E+03	7.2034E+03	**1.6225E+03**
	std	2.4806E-01	7.6316E+00	2.3141E+03	3.1555E-01
kCGSA	best	1.3003E+03	1.4002E+03	1.5679E+03	1.6219E+03
	worst	1.3005E+03	1.4004E+03	1.6851E+03	1.6236E+03
	mean	**1.3004E+03**	**1.4003E+03**	**1.6064E+03**	1.6227E+03
	std	4.5266E-02	2.4453E-02	3.1822E+01	4.3153E-01
		F17	F18	F19	F20

GSA	best	3.2328E+08	6.5264E+09	2.6412E+03	1.3752E+05
	worst	3.8117E+08	9.6694E+09	2.8864E+03	2.6373E+05
	mean	3.5712E+08	8.0952E+09	2.7500E+03	1.8234E+05
	std	2.4300E+07	1.2486E+09	1.0496E+02	4.9788E+04
CGSA	best	2.7468E+06	4.8432E+03	1.9677E+03	2.4354E+04
	worst	1.6806E+07	5.7171E+03	2.0619E+03	4.2171E+04
	mean	6.8570E+06	5.5931E+03	2.0166E+03	**3.2009E+04**
	std	2.9054E+06	1.8732E+02	2.2906E+01	3.9301E+03
kCGSA	best	8.6868E+05	2.2167E+03	1.9242E+03	3.6022E+04
	worst	6.7932E+06	5.3651E+03	1.9922E+03	7.6157E+04
	mean	**2.7069E+06**	**3.9157E+03**	**1.9446E+03**	5.6416E+04
	std	1.2557E+06	8.9054E+02	2.5938E+01	8.7350E+03
		F21	F22	F23	F24
GSA	best	2.0052E+07	4.8415E+04	2.5087E+03	2.6301E+03
	worst	6.2448E+07	8.6539E+04	2.5883E+03	2.6490E+03
	mean	3.0535E+07	6.6402E+04	2.5440E+03	2.6408E+03
	std	1.7986E+07	1.6204E+04	3.8947E+01	7.8494E+00
CGSA	best	1.1285E+06	3.3770E+03	2.5001E+03	2.6009E+03
	worst	3.1695E+06	4.5246E+03	2.5003E+03	2.6411E+03
	mean	**1.9032E+06**	4.0990E+03	**2.5002E+03**	**2.6083E+03**
	std	4.5781E+05	2.6576E+02	3.6078E-02	1.0945E+01
kCGSA	best	1.7078E+06	3.0783E+03	2.6700E+03	2.6495E+03
	worst	4.8386E+06	4.6923E+03	2.7107E+03	2.6604E+03
	mean	2.5898E+06	**4.0600E+03**	2.6895E+03	2.6565E+03
	std	7.1300E+05	3.5877E+02	1.0024E+01	2.6828E+00
		F25	F26	F27	F28
GSA	best	2.7042E+03	2.8002E+03	3.6758E+03	6.2910E+03
	worst	2.7091E+03	2.8003E+03	5.6887E+03	8.1086E+03
	mean	2.7065E+03	2.8003E+03	4.8009E+03	6.6644E+03
	std	2.0645E+00	4.0504E-02	7.6192E+02	8.0743E+02
CGSA	best	2.7000E+03	2.7195E+03	3.0309E+03	4.8502E+03
	worst	2.7000E+03	2.8001E+03	6.3511E+03	7.6073E+03
	mean	2.7000E+03	**2.7974E+03**	**4.7020E+03**	**6.0039E+03**
	std	5.7986E-04	1.4716E+01	8.6102E+02	6.2838E+02
kCGSA	best	2.7000E+03	2.8001E+03	3.8052E+03	6.5735E+03
	worst	2.7000E+03	2.8001E+03	6.7656E+03	1.0853E+04
	mean	**2.7000E+03**	2.8001E+03	4.7693E+03	8.6092E+03
	std	2.1453E-04	1.7336E-02	8.9657E+02	1.0502E+03
		F29	F30		
GSA	best	3.1002E+03	3.2000E+03		
	worst	2.2047E+08	1.7122E+07		
	mean	6.3517E+07	4.8557E+06		
	std	9.4153E+07	6.9557E+06		
CGSA	best	2.4270E+05	1.3483E+04		
	worst	4.7114E+05	2.2525E+04		
	mean	3.4838E+05	**1.7414E+04**		
	std	4.3281E+04	2.2063E+03		
kCGSA	best	9.5967E+03	1.2265E+05		
	worst	6.7835E+04	2.4485E+05		
	mean	**1.3452E+04**	1.6824E+05		
	std	1.0335E+04	3.2623E+04		

4.3. Test for influence of sine function

Table 2 illustrates the effects of sine function tested in CEC 2014 benchmark functions. The results of GSA have been presented in Table 1 above, so it will not listed in Table 2.

The tested equation (sine for CGSA) is shown as Eq. 19:

$$V_i^d(t+1) = sin(random_i \times \pi) \times V_i^d(t) + a_i^d(t) \tag{19}$$

There are 100 agents included in these experiments. Obviously, Equation 19 helps agents perform better than the initial movement pattern. Despite some under-performance in several composite functions, SCGSA remains highly competitive in a majority of other functions.

Table 2: Results for the influence of sine function

D=50,map=7		F1	F2	F3	F4
CGSA	best	1.3104E+08	1.8022E+10	9.7122E+04	2.8788E+03
	worst	2.9415E+08	3.1028E+10	1.3448E+05	5.0999E+03
	mean	1.8539E+08	2.4170E+10	1.1034E+05	3.5776E+03
	std	3.2326E+07	2.7841E+09	8.1340E+03	4.6358E+02
sinCGSA	best	4.2597E+07	6.1872E+09	9.2374E+04	1.2542E+03
	worst	1.5286E+08	1.5331E+10	1.2124E+05	2.3770E+03
	mean	**8.7327E+07**	**1.0534E+10**	**1.0526E+05**	**1.6727E+03**
	std	2.6114E+07	2.4386E+09	7.0149E+03	2.6204E+02
		F5	F6	F7	F8
CGSA	best	5.2000E+02	6.5288E+02	8.9715E+02	1.0517E+03
	worst	5.2000E+02	6.6538E+02	1.0079E+03	1.1064E+03
	mean	5.2000E+02	6.5878E+02	9.4427E+02	1.0802E+03
	std	1.1788E-04	2.2370E+00	2.9511E+01	1.4230E+01
sinCGSA	best	5.2000E+02	6.4702E+02	7.6677E+02	1.0308E+03
	worst	5.2000E+02	6.5996E+02	8.3586E+02	1.0985E+03
	mean	**5.2000E+02**	**6.5404E+02**	**7.9499E+02**	**1.0621E+03**
	std	1.1821E-04	3.1581E+00	1.6562E+01	1.4345E+01
		F9	F10	F11	F12
CGSA	best	1.2124E+03	7.4877E+03	8.5120E+03	1.2000E+03
	worst	1.3179E+03	1.0234E+04	1.2243E+04	1.2000E+03
	mean	1.2655E+03	8.7901E+03	9.9671E+03	1.2000E+03
	std	2.9172E+01	6.7362E+02	8.8717E+02	7.8763E-03
sinCGSA	best	1.1786E+03	6.9303E+03	8.3309E+03	1.2000E+03
	worst	1.2691E+03	9.9949E+03	1.0959E+04	1.2000E+03
	mean	**1.2239E+03**	**8.5145E+03**	**9.3689E+03**	**1.2000E+03**
	std	2.1151E+01	6.8138E+02	6.5451E+02	5.6541E-03
		F13	F14	F15	F16
CGSA	best	1.3026E+03	1.4315E+03	3.5255E+03	1.6222E+03
	worst	1.3035E+03	1.4670E+03	1.4668E+04	1.6233E+03
	mean	1.3032E+03	1.4503E+03	7.3491E+03	1.6227E+03
	std	2.0432E-01	7.7567E+00	2.5266E+03	2.3046E-01
sinCGSA	best	1.3005E+03	1.4003E+03	1.8346E+03	1.6224E+03
	worst	1.3023E+03	1.4269E+03	3.8063E+03	1.6232E+03
	mean	**1.3009E+03**	**1.4102E+03**	**2.5537E+03**	**1.6227E+03**
	std	6.2699E-01	8.4357E+00	4.5514E+02	1.8655E-01
		F17	F18	F19	F20
CGSA	best	1.4952E+06	3.9018E+03	1.9702E+03	2.4527E+04
	worst	2.0222E+07	5.7404E+03	2.0529E+03	3.9284E+04
	mean	6.9059E+06	5.5915E+03	2.0137E+03	3.2661E+04
	std	3.7706E+06	3.2419E+02	2.4442E+01	3.4059E+03

sinCGSA	best	9.1767E+05	3.5134E+03	1.9301E+03	2.6228E+04
	worst	1.2671E+07	5.7129E+03	1.9999E+03	3.5285E+04
	mean	**2.6835E+06**	**5.2947E+03**	**1.9627E+03**	**3.0922E+04**
	std	2.4984E+06	6.2116E+02	2.1114E+01	2.6601E+03
		F21	F22	F23	F24
CGSA	best	1.0233E+06	3.5529E+03	2.5001E+03	2.6007E+03
	worst	2.7273E+06	5.0207E+03	2.5003E+03	2.6493E+03
	mean	1.9353E+06	4.2914E+03	**2.5002E+03**	**2.6106E+03**
	std	3.7005E+05	3.7709E+02	3.1398E-02	1.2321E+01
sinCGSA	best	1.0905E+06	3.5323E+03	2.5002E+03	2.6010E+03
	worst	2.4901E+06	4.9315E+03	2.5004E+03	2.6508E+03
	mean	**1.6947E+06**	**4.1219E+03**	2.5003E+03	2.6172E+03
	std	3.5433E+05	3.6096E+02	5.3532E-02	1.5156E+01
		F25	F26	F27	F28
CGSA	best	2.7000E+03	2.8000E+03	3.0247E+03	5.1056E+03
	worst	2.7000E+03	2.8001E+03	6.6976E+03	8.0097E+03
	mean	**2.7000E+03**	2.8001E+03	**4.4883E+03**	**6.2061E+03**
	std	6.3377E-04	1.1912E-02	8.2792E+02	7.7446E+02
sinCGSA	best	2.7000E+03	2.7805E+03	3.0543E+03	5.0575E+03
	worst	2.7000E+03	2.8001E+03	6.2012E+03	8.2805E+03
	mean	2.7000E+03	**2.7994E+03**	4.8054E+03	6.6332E+03
	std	1.4859E-03	3.5664E+00	7.7992E+02	8.8761E+02
		F29	F30		
CGSA	best	2.5129E+05	1.3310E+04		
	worst	4.6191E+05	2.6578E+04		
	mean	**3.4949E+05**	**1.8343E+04**		
	std	5.8180E+04	2.7296E+03		
sinCGSA	best	3.9413E+05	2.0442E+04		
	worst	1.0040E+06	6.6381E+05		
	mean	5.9818E+05	7.4522E+04		
	std	1.2966E+05	1.4689E+05		

4.4. Comprehensive influence

It's natural to examine the comprehensive influence since the introduction of sine function and parameter k both have a positive effect on the overall performance. Likewise, Table 3 only presents results of SCGSA, for the results of GSA and CGSA are already listed in Table 1. It is clear that SCGSA performs better than kCGSA and sinCGSA in most functions compared with the data in Tables 1 and 2. This set of tests adopt the seventh chaotic map as an illustration. In Table 3, most of minimum values are calculated by SCGSA. It is shown that the results for the mixed movement pattern are very promising in a majority of functions compared with CGSA, kCGSA, sinCGSA. In spite of some under-performance in certain composite functions, little difference is found in final values among them. This can be explained by the suggestion that SCGSA explores more in initial phases to seek other promising areas.

Table 3: Results for the comprehensive influence

D=50,map=7		F1	F2	F3	F4
CGSA	best	1.2977E+08	1.8367E+10	1.0054E+05	2.3340E+03
	worst	2.8006E+08	3.1031E+10	1.2614E+05	4.6040E+03
	mean	1.9663E+08	2.4042E+10	**1.0946E+05**	3.3520E+03
	std	3.7521E+07	3.1743E+09	6.5545E+03	4.7812E+02
SCGSA	best	1.2785E+07	7.0240E+02	1.0567E+05	5.3825E+02
	worst	3.3507E+07	1.5591E+04	1.5602E+05	6.7165E+02
	mean	**2.0969E+07**	**6.6886E+03**	1.3156E+05	**6.0800E+02**
	std	5.4594E+06	4.4247E+03	1.3837E+04	3.1896E+01
		F5	F6	F7	F8
CGSA	best	5.2000E+02	6.5363E+02	8.7955E+02	1.0517E+03
	worst	5.2000E+02	6.6373E+02	9.8760E+02	1.1074E+03
	mean	5.2000E+02	6.5894E+02	9.3477E+02	1.0787E+03
	std	8.7548E-05	2.6437E+00	2.6505E+01	1.3152E+01
SCGSA	best	5.2000E+02	6.1606E+02	7.0000E+02	9.4029E+02
	worst	5.2000E+02	6.2675E+02	7.0002E+02	9.9700E+02
	mean	5.2000E+02	**6.2182E+02**	**7.0000E+02**	**9.6675E+02**
	std	1.0036E-04	2.4727E+00	4.1962E-03	1.4198E+01
		F9	F10	F11	F12
CGSA	best	1.2064E+03	7.8454E+03	8.3296E+03	1.2000E+03
	worst	1.2930E+03	9.4821E+03	1.1687E+04	1.2000E+03
	mean	1.2626E+03	8.7656E+03	9.8674E+03	1.2000E+03
	std	2.1212E+01	4.1911E+02	7.3151E+02	5.2492E-03
SCGSA	best	1.0154E+03	5.9835E+03	5.7882E+03	1.2000E+03
	worst	1.0851E+03	8.7900E+03	8.7988E+03	1.2000E+03
	mean	**1.0516E+03**	**7.4684E+03**	**7.6071E+03**	**1.2000E+03**
	std	1.9154E+01	6.6441E+02	6.9582E+02	3.6754E-03
		F13	F14	F15	F16
CGSA	best	1.3024E+03	1.4352E+03	3.2370E+03	1.6221E+03
	worst	1.3036E+03	1.4604E+03	1.6315E+04	1.6232E+03
	mean	1.3032E+03	1.4468E+03	8.4019E+03	1.6226E+03
	std	2.6183E-01	5.3758E+00	2.9961E+03	2.4437E-01
SCGSA	best	1.3003E+03	1.4003E+03	1.5266E+03	1.6219E+03
	worst	1.3005E+03	1.4004E+03	1.6119E+03	1.6234E+03
	mean	**1.3004E+03**	**1.4003E+03**	**1.5634E+03**	**1.6227E+03**
	std	5.3482E-02	2.8369E-02	2.0910E+01	3.6786E-01
		F17	F18	F19	F20
CGSA	best	2.6453E+06	2.2294E+03	1.9807E+03	2.5691E+04
	worst	1.4424E+07	5.7167E+03	2.0845E+03	4.8720E+04
	mean	6.7511E+06	5.3475E+03	2.0245E+03	**3.2975E+04**
	std	2.9620E+06	7.5506E+02	2.9533E+01	4.1742E+03
SCGSA	best	9.8073E+05	2.2012E+03	1.9235E+03	3.5255E+04
	worst	3.7865E+06	5.1455E+03	1.9917E+03	7.3234E+04
	mean	**2.0741E+06**	**3.4097E+03**	**1.9405E+03**	5.3117E+04
	std	6.7967E+05	8.7113E+02	2.6012E+01	1.0061E+04
		F21	F22	F23	F24
CGSA	best	1.1527E+06	3.6107E+03	2.5001E+03	2.6007E+03
	worst	3.1854E+06	4.9316E+03	2.5002E+03	2.6462E+03
	mean	**2.0040E+06**	4.2177E+03	**2.5002E+03**	**2.6094E+03**
	std	4.9289E+05	3.1465E+02	3.3445E-02	1.3007E+01
SCGSA	best	1.4881E+06	3.1637E+03	2.6614E+03	2.6556E+03
	worst	3.3142E+06	4.4634E+03	2.6939E+03	2.6617E+03
	mean	2.5013E+06	**3.8618E+03**	2.6728E+03	2.6584E+03
	std	5.2559E+05	3.0729E+02	8.2652E+00	1.5923E+00
		F25	F26	F27	F28
CGSA	best	2.7000E+03	2.8000E+03	3.0143E+03	5.1694E+03
	worst	2.7000E+03	2.8001E+03	5.9423E+03	8.5442E+03

	mean	2.7000E+03	**2.8001E+03**	4.5015E+03	**6.5633E+03**
	std	9.6785E-04	1.3733E-02	7.9788E+02	9.9138E+02
SCGSA	best	2.7000E+03	2.8001E+03	3.6217E+03	5.9005E+03
	worst	2.7000E+03	2.8002E+03	4.2380E+03	1.0604E+04
	mean	**2.7000E+03**	2.8001E+03	**3.9156E+03**	8.6096E+03
	std	2.3409E-04	2.3559E-02	1.6177E+02	1.1022E+03
		F29	F30		
CGSA	best	2.6003E+05	1.3631E+04		
	worst	4.6185E+05	2.7454E+04		
	mean	3.3439E+05	**1.8632E+04**		
	std	5.1519E+04	3.0823E+03		
SCGSA	best	9.3996E+03	9.8017E+04		
	worst	1.9447E+04	1.5269E+05		
	mean	**1.3256E+04**	1.2298E+05		
	std	2.3089E+03	1.7921E+04		

4.5. Algorithms for comparison

The results of SCGSA and other four recently proposed algorithms are compared one another in CEC 2014 in Table 4, with the adoption of the ninth chaotic map. Obviously, the total ranking of SCGSA is first, which further confirms the superiority of SCGSA. The comparative convergence curves are provided in Appendix 8.2.

Table 4: Result for total contrast experiments

		TSA	CLPSO	LIPS	HPSOTVAC	SCGSA
F1	mean	2.7409E+08	6.4796E+08	8.7091E+07	1.0210E+07	**4.7412E+06**
	std	3.9103E+07	1.3770E+08	2.7973E+07	4.5085E+06	1.3746E+06
F2	mean	8.9155E+06	2.1543E+10	4.8969E+08	9.4887E+05	**2.5665E+03**
	std	5.0245E+06	2.1656E+09	2.7752E+08	7.1022E+05	2.3750E+03
F3	mean	8.2883E+04	1.2068E+05	1.3221E+05	**1.5057E+04**	1.2557E+05
	std	6.6496E+03	1.0177E+04	1.6837E+04	4.1789E+03	1.2563E+04
F4	mean	7.3404E+02	5.0505E+03	1.0256E+03	6.4594E+02	**5.3590E+02**
	std	2.2591E+01	6.5077E+02	1.8938E+02	4.7756E+01	1.4451E+01
F5	mean	5.2111E+02	5.2087E+02	5.2121E+02	5.2073E+02	**5.2000E+02**
	std	3.0756E-02	5.7151E-02	2.7401E-02	1.4776E-01	2.5143E-04
F6	mean	6.5432E+02	6.5696E+02	6.3321E+02	6.4888E+02	**6.2028E+02**
	std	2.3362E+00	2.0524E+00	2.9529E+00	4.3530E+00	2.7505E+00
F7	mean	7.0080E+02	8.8777E+02	7.0616E+02	7.0070E+02	**7.0000E+02**
	std	4.6644E-02	3.1950E+01	2.2276E+00	2.2567E-01	1.4736E-05
F8	mean	1.1783E+03	1.0503E+03	**9.3526E+02**	9.7374E+02	9.5074E+02
	std	1.6898E+01	1.6379E+01	1.0286E+01	1.7533E+01	2.7075E+01
F9	mean	1.3396E+03	1.4181E+03	1.0718E+03	1.1472E+03	**1.0399E+03**
	std	1.2672E+01	1.5144E+01	2.0853E+01	3.0262E+01	1.7751E+01
F10	mean	1.2208E+04	6.7827E+03	5.5648E+03	**4.5685E+03**	6.5946E+03
	std	5.4729E+02	2.4085E+02	7.6180E+02	7.2773E+02	5.7942E+02
F11	mean	1.3947E+04	1.1274E+04	**7.1479E+03**	7.1501E+03	7.2576E+03
	std	4.3407E+02	4.3648E+02	7.2954E+02	6.0499E+02	6.4633E+02
F12	mean	1.2032E+03	1201.42669	1201.27643	1.2007E+03	**1.2000E+03**
	std	1.0455E-01	0.15623993	1.07227904	2.2575E-01	0.000842781
F13	mean	1.3006E+03	1.3031E+03	1.3005E+03	1.3006E+03	**1.3004E+03**
	std	9.0267E-02	4.4539E-01	4.0931E-02	6.9805E-02	2.9101E-02

F14	mean	1.4003E+03	1.4556E+03	1.4004E+03	1.4004E+03	**1.4003E+03**
	std	3.6176E-02	6.7917E+00	4.0072E-02	2.0975E-01	1.4132E-02
F15	mean	1.5548E+03	1.6734E+05	1.6368E+03	1.6603E+03	**1.5204E+03**
	std	6.9462E+00	7.8265E+04	6.2718E+01	1.8859E+01	2.5727E+00
F16	mean	1.6222E+03	1.6216E+03	**1.6209E+03**	1.6210E+03	1.6228E+03
	std	1.5907E-01	2.6159E-01	4.9890E-01	3.3978E-01	3.8388E-01
F17	mean	1.2293E+07	6.8571E+07	3.4048E+06	2.2544E+06	**1.2762E+06**
	std	2.5325E+06	2.8948E+07	1.2558E+06	1.7750E+06	1.8865E+05
F18	mean	**2.5434E+03**	4.6008E+08	2.6978E+03	3.3494E+03	2.9604E+03
	std	4.4310E+02	2.1334E+08	1.5832E+02	1.0668E+03	1.1210E+03
F19	mean	1.9393E+03	2.1464E+03	1.9631E+03	1.9640E+03	**1.9229E+03**
	std	8.0194E+00	6.1519E+01	1.9027E+01	3.8369E+01	1.6784E+00
F20	mean	1.7326E+04	6.7633E+04	5.6365E+04	**1.6901E+04**	4.2120E+04
	std	5.8690E+03	2.6219E+04	1.7109E+04	8.8804E+03	3.9259E+03
F21	mean	4.8133E+06	1.6553E+07	1.9215E+06	1.7313E+06	**1.3200E+06**
	std	1.2106E+06	3.8894E+06	8.6777E+05	1.2293E+06	2.5057E+05
F22	mean	3.4687E+03	4.5836E+03	**2.7627E+03**	3.7002E+03	3.5549E+03
	std	2.5122E+02	2.4693E+02	1.7916E+02	4.9034E+02	5.8546E+02
F23	mean	2.6441E+03	2.8615E+03	2.6679E+03	**2.6378E+03**	2.6499E+03
	std	3.8011E-02	2.9845E+01	7.7572E+00	6.3549E-01	6.7898E-01
F24	mean	2.6839E+03	2.7631E+03	2.6982E+03	2.6834E+03	**2.6563E+03**
	std	9.6178E-01	1.3141E+01	4.0331E+00	9.3285E+00	7.1261E-01
F25	mean	2.7627E+03	2.8036E+03	2.7402E+03	2.7332E+03	**2.7000E+03**
	std	4.3598E+00	6.4132E+00	3.8386E+00	6.5567E+00	8.4787E-05
F26	mean	**2.7008E+03**	2.7240E+03	2.7513E+03	2.8008E+03	2.8001E+03
	std	3.4666E-02	1.6710E+01	5.8505E+01	2.6987E-01	2.1993E-02
F27	mean	4.2097E+03	4.5318E+03	3.9922E+03	4.4094E+03	**3.7294E+03**
	std	5.8435E+01	1.2667E+02	4.7480E+01	1.3952E+02	1.2055E-03
F28	mean	**5.3725E+03**	9.3457E+03	6.6541E+03	5.6493E+03	8.3660E+03
	std	2.2605E+01	6.9536E+02	2.3324E+02	1.9590E+03	3.4785E+02
F29	mean	1.3233E+06	1.4143E+08	1.9247E+05	**3.1540E+03**	1.1794E+04
	std	3.9999E+05	2.2537E+07	3.3635E+05	3.2097E+01	8.1456E+02
F30	mean	8.8386E+04	1.4497E+06	2.3716E+05	**5.0404E+03**	5.7626E+04
	std	1.2678E+04	3.4977E+05	1.3893E+05	6.7627E+02	2.7228E+03
Average ranking		3.17	4.57	2.90	2.43	1.93
Total ranking		4	5	3	2	1

All data generated in Table 4 are introduced into Wilcoxon's rank sum test to confirm their validity. The results are listed in Table 5. Undoubtedly, R^+ values were lager than the R^- values in all cases and all tests reject null hypothesis Except HPSOTVAC, which proves the superiority of SCGSA over others.

Table 5: Results of Wilcoxon's rank sum test

	Better	Equal	Worst	R^+	R^-	p-value	α=0.1	α=0.05
SCGSA versus TSA	22	0	8	342	123	0.026666	Yes	Yes
SCGSA versus CLPSO	27	0	3	435	30	0.000031	Yes	Yes
SCGSA versus LIPS	22	0	8	343	122	0.023031	Yes	Yes
SCGSA versus HPSOTVAC	21	0	9	294	171	0.205840	No	No
SCGSA versus CGSA	22	0	8	337	128	0.041139	Yes	Yes

5. Discussion

All experiments in Section 4 (Experiments and results) prove the superiority of SCGSA. The features of SCGSA can be generalized into four aspects:

- SCGSA adopts sine function in order to reserve enough time for searching in higher probability optimum areas and thus agents can find global optima faster within the same time scale.
- k is negatively correlated with t. Hence, agents assisted by k are affected by others greatly in the initial stage, which strengthens its exploratory ability. In the later stage, k decreases the growth of velocity to zero so as to enhance the ability of exploitation. It is unquestionably an excellent algorithm that keeps a good balance between exploration and exploitation in a way to rapidly find solutions in remote positions at the beginning while exploiting in final iterations with the knowledge the probable position.
- This algorithm still uses chaotic maps to avoid local optima because agents become heavier as the number of iteration increases.
- Its flexibility allows users to choose chaotic maps largely contingent on the situations without any alteration of other factors.

In spite of the substantial improvements, there still exist some minor weaknesses in regards to the value of constant.

The first one is the equation of k. In Eq. 16, $k = 2 \times (1 - t/T)$. Since 2 is greater than 1, it can strengthen the exploration ability at the very beginning. Therefore, it's interesting to explore whether larger numbers can yield better performance. In this part, the equation is altered to $k = m - mt/T$ and the ninth chaotic map is selected.

Table 6 shows that larger numbers may break the balance of exploration and exploitation. At present, m=2 is the most appropriate choice. Occasionally however, for some specific functions when m=2, solutions are still trapped in local optima. Hence, in the future, people may refine this algorithm by finding better m.

Table 6: Results for changing m

F(x)	m=2	m=4	m=6	m=8	m=10	m=12
F1	**5.9378E+06**	3.1738E+09	3.1284E+09	6.1820E+09	6.4282E+09	8.4679E+09
F2	**4.6652E+03**	1.5071E+11	9.7344E+10	1.6539E+11	2.1554E+11	2.8182E+11
F3	**1.2557E+05**	2.4389E+05	2.6156E+05	2.7503E+05	5.4000E+05	6.4051E+05
F4	**5.5043E+02**	3.5704E+04	2.7101E+04	4.7047E+04	8.6265E+04	1.0621E+05
F5	**5.2000E+02**	5.2032E+02	5.2126E+02	5.2129E+02	5.2133E+02	5.2132E+02
F6	**6.2028E+02**	6.4965E+02	6.7610E+02	6.7554E+02	6.8278E+02	6.8174E+02
F7	**7.0000E+02**	2.1577E+03	1.5549E+03	2.0923E+03	2.8629E+03	3.0887E+03
F8	**9.6118E+02**	1.4643E+03	1.3450E+03	1.5444E+03	1.6803E+03	1.7203E+03
F9	**1.0399E+03**	1.6727E+03	1.5860E+03	1.7733E+03	1.8995E+03	1.9662E+03
F10	**7.3822E+03**	1.1562E+04	1.5709E+04	1.6749E+04	1.7115E+04	1.7125E+04
F11	**7.7444E+03**	9.7504E+03	1.5679E+04	1.6560E+04	1.6658E+04	1.6956E+04
F12	**1.2000E+03**	1.2001E+03	1.2044E+03	1.2056E+03	1.2050E+03	1.2059E+03
F13	**1.3004E+03**	1.3083E+03	1.3063E+03	1.3090E+03	1.3108E+03	1.3121E+03
F14	**1.4003E+03**	1.7693E+03	1.6475E+03	1.7542E+03	2.0165E+03	2.0669E+03
F15	**1.5204E+03**	6.2274E+06	6.8858E+06	1.0157E+07	1.2934E+08	1.0580E+08
F16	**1.6228E+03**	1.6234E+03	1.6234E+03	1.6234E+03	1.6239E+03	1.6241E+03
F17	**1.2762E+06**	2.7499E+08	4.8999E+08	8.6464E+08	9.9623E+08	9.3528E+08
F18	**3.9974E+03**	1.4696E+10	1.1049E+10	1.2130E+10	2.6326E+10	3.6101E+10
F19	**1.9229E+03**	3.6963E+03	3.0840E+03	5.1874E+03	4.9252E+03	7.0680E+03
F20	**4.2120E+04**	1.7668E+06	3.7469E+05	2.6704E+06	1.1033E+07	1.6290E+07
F21	**1.3200E+06**	9.9815E+07	1.6621E+08	1.3193E+08	2.3060E+08	3.3419E+08
F22	**3.9607E+03**	9.4078E+03	1.0118E+04	7.2976E+04	4.6339E+04	1.5127E+05
F23	**2.6499E+03**	3.9400E+03	3.7094E+03	4.2457E+03	5.9903E+03	6.1964E+03
F24	**2.6573E+03**	2.8269E+03	2.7756E+03	2.9744E+03	3.2015E+03	3.2993E+03
F25	**2.7000E+03**	2.7697E+03	2.7525E+03	2.9228E+03	3.2070E+03	3.1686E+03
F26	2.8001E+03	2.7486E+03	**2.7214E+03**	2.7372E+03	2.7241E+03	2.9117E+03
F27	**3.7294E+03**	4.5518E+03	4.9703E+03	5.2516E+03	5.3224E+03	5.3029E+03
F28	9.5373E+03	**7.3753E+03**	1.2587E+04	1.2844E+04	1.5537E+04	1.6519E+04
F29	**1.3943E+04**	5.6135E+07	1.8415E+08	1.9085E+08	2.7054E+08	1.9162E+08
F30	**5.7626E+04**	2.1670E+06	1.0802E+07	9.5783E+06	1.6770E+07	1.5552E+07

Moreover, in SCA, velocity has different directions to keep the balance, so whether direction has significant effects on performance still needs to be explored. Maybe in initial phases, agents in GSA, CGSA and SCGSA have missed possible areas but still keep exploring forward. Introducing $(-1)^n$ provides a chance to correct errors. This set of tests add $(-1)^n$, (n=1,2) in front of sine function (named markedSCGSA) in order to change the direction of present v. And n equals 1 or 2 so the probability of positive and negative are equal. The set of experiments below adopt the ninth chaotic map and are tested by 30 benchmark functions.

As shown in Table 7, markedSCGSA doesn't work well except in specific composite benchmark functions, but it still offers an interesting idea for future works.

Table 7: Results for marked equations

	GSA	CGSA	SCGSA	markedSCGSA
F1	2.4712E+09	3.7154E+07	**5.9378E+06**	5.4943E+07
F2	9.7990E+10	7.6530E+09	**4.6652E+03**	3.6149E+08
F3	1.5351E+05	**8.9303E+04**	1.2557E+05	1.3364E+05
F4	2.1076E+04	1.3721E+03	**5.5043E+02**	1.0111E+03
F5	5.2000E+02	5.2000E+02	5.2000E+02	**5.2000E+02**
F6	6.6436E+02	6.5701E+02	**6.2028E+02**	6.3324E+02
F7	1.5275E+03	7.7132E+02	**7.0000E+02**	7.0940E+02
F8	1.0788E+03	1.0642E+03	**9.6118E+02**	1.0149E+03
F9	1.2896E+03	1.2686E+03	**1.0399E+03**	1.1129E+03
F10	8.7514E+03	8.1048E+03	**7.3822E+03**	8.3421E+03
F11	1.0003E+04	9.4699E+03	**7.7444E+03**	8.3813E+03
F12	1.2001E+03	1.2000E+03	**1.2000E+03**	1.2000E+03
F13	1.3065E+03	1.3005E+03	**1.3004E+03**	**1.3004E+03**
F14	1.6131E+03	1.4032E+03	**1.4003E+03**	**1.4003E+03**
F15	8.1498E+05	2.0338E+03	**1.5204E+03**	2.1962E+03
F16	1.6228E+03	1.6226E+03	1.6228E+03	**1.6224E+03**
F17	4.3769E+08	9.7153E+05	**1.2762E+06**	3.9762E+06
F18	8.2964E+09	5.6085E+03	**3.9974E+03**	4.2438E+03
F19	2.8270E+03	1.9459E+03	**1.9229E+03**	1.9544E+03
F20	1.6308E+05	2.6179E+04	**4.2120E+04**	4.7484E+04
F21	3.0556E+07	7.5256E+05	**1.3200E+06**	2.4829E+06
F22	2.6491E+04	4.1996E+03	**3.9607E+03**	4.3005E+03
F23	2.5131E+03	**2.5001E+03**	2.6499E+03	2.6214E+03
F24	2.6550E+03	2.6007E+03	**2.6573E+03**	2.6590E+03
F25	2.7063E+03	2.7000E+03	2.7000E+03	**2.7000E+03**
F26	2.8004E+03	2.8000E+03	2.8001E+03	**2.8000E+03**
F27	4.9341E+03	4.3848E+03	**3.7294E+03**	5.1541E+03

F28	6.9627E+03	**6.4974E+03**	9.5373E+03	7.1135E+03
F29	1.0936E+08	2.3139E+05	**1.3943E+04**	2.0892E+04
F30	1.4555E+07	1.3927E+04	**5.7626E+04**	2.0505E+05

6. Real cases of SCGSA

6.1. Example 1: Pressure Vessel Design (PVD)

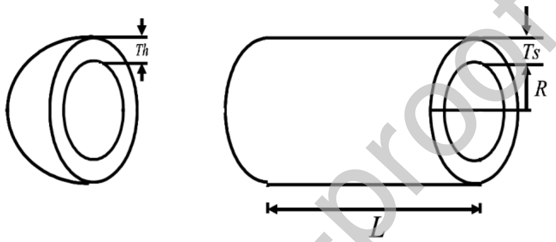


Figure 7: The pressure vessel design problem

Figure 7 shows a pressure vessel with the volume of 750 ft^3 and a working pressure of 3000 PSI. The aim of the pressure vessel design problem is to minimize the manufacturing cost (Onwubolu & Babu, 2004). The 4 variables are x_1 (thickness T_s of the shell), x_2 (thickness T_h of the head), x_3 (inner radius R) and x_4 (length L of the cylindrical section of the vessel, not including the head). Besides, x_1 and x_2 are to be in integral multiples of 0.0625 inch which are the available thickness of rolled steel plates. The radius x_3 and the length x_4 are continuous variables.

The minimization function is shown as Eq. 20.

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1611x_1^2x_4 + 19.84x_1^2x_3 \tag{20}$$

Table 8: Computational results for pressure vessel design problem

Pop size=100	T_s	T_h	R	L	Best cost
SCGSA	1.1000	0.6000	56.9594	51.2787	7024.5700
CGSA	1.1000	0.6000	47.4837	119.6530	7892.8700

The subject conditions are shown as Eq. 21.

$$
\begin{aligned}
& g_1(x_1, x_3) = -x_1 + 0.0193x_3 \le 0 \\
& g_2(x_2, x_3) = -x_2 + 0.0095x_3 \le 0 \\
& g_3(x_3, x_4) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0 \\
& g_4(x_4) = x_4 - 240 \le 0 \\
& g_5(x_1) = 1.1 - x_1 \le 0 \\
& g_6(x_2) = 0.6 - x_2 \le 0
\end{aligned}
\tag{21}
$$

Table 8 shows results of the pressure vessel design problem by two algorithms. Compared with CGSA, it proves that SCGSA is better to solve this real problem.

6.2. Example 2: Tension/Compression Spring Design problem (T/CSD)

The Tension/Compression Spring Design problem (T/CSD) is a continuous constrained problem. It was described by Belegundu (Belegundu, 1985). The variables are the mean coil diameter D, the diameter d and the number of active coils N. Figure 8 shows the model.

The problem can be stated as:

Consider : x=[x_1,x_2,x_3]=[d,D,N]

Minimize : $f(x) = (N + 2)Dd^2$

Subject to :

$g_1(x) = 1 - \dfrac{D^3 N}{71785D^4} \le 0$

$g_2(x) = \dfrac{4D^2 - dD}{12566(Dd^3 - d^4)} + \dfrac{1}{5108d^2} - 1 \le 0$

$g_3(x) = 1 - \dfrac{140.45d}{D^2 N} \le 0$

Table 9: Computational results for TCSD

Pop size=100	x_1	x_2	x_3	Best cost	mean	std
SCGSA	0.0500	0.3137	14.5458	0.0127	0.0131	0.0002
CGSA	0.0500	0.3136	14.5851	0.0130	0.0148	0.0027

$g_4(x) = \frac{D+d}{1.5} - 1 \leq 0$

The problem consists of the design variables as the following: $0.050000 \leq d \leq 2.000000$, $0.25000 \leq D \leq 1.300000$ and $2.000000 \leq N \leq 15.000000$.

Table 9 presents the solutions. In this case we adopt the fifth map and the result reveal the superiority of SCGSA.

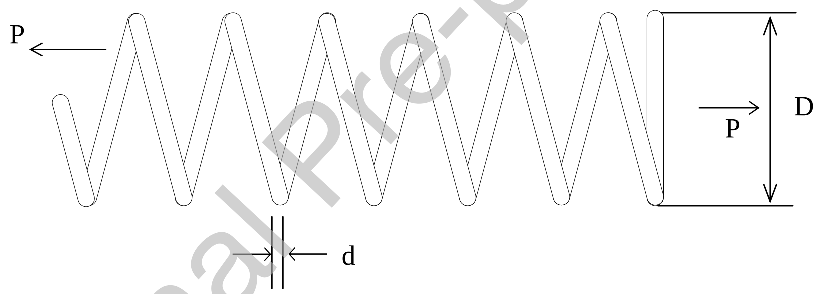


Figure 8: The Tension/Compression Spring Design problem (T/CSD)

6.3. Example 3: Welded Beam Design (WBD)

A welded beam is designed for minimum cost subject to constraints on shear stress (τ) , bending stress in the beam (σ), buckling load on the bar (p_c), end deflection of the beam (δ), and side constraints (Coello, 2000). There are four variables: *h*(x_1), *l*(x_2), *t*(x_3) and *b*(x_4).

The problem can be stated as follows:

Minimize:

$F(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$

Table 10: Computational results for WBD

Pop size=100	h	l	d	b	Best cost	mean	std
SCGSA	0.1584	4.5030	9.0779	0.2390	2.0560	2.8469	0.3409
CGSA	0.2672	2.3728	7.2995	0.3343	2.1093	2.8956	0.4399

Subject to :

$g_1(x) = \tau(x) - \tau_{max} \leq 0$

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

$g_3(x) = x - 1 - x_4 \leq 0$

$g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0$

$g_5(x) = 0.125 - x_1 \leq 0$

$g_6(x) = \delta(x) - \delta_{max} \leq 0$

$g_7(x) = P - P_c(x) \leq 0$

where

$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$

$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = p(L + \frac{X_2}{2})$

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

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

$\sigma(x) = \frac{6PL}{x_4x_3^2}, \ \delta(x) = \frac{4PL^3}{EX_3^3x_4}$

$P_c(x) = \frac{4.013E\sqrt{\frac{X_3^2X_4^6}{36}}}{L^2}(1 - \frac{X_3}{2L}\sqrt{\frac{E}{4G}})$

$P = 6000lb, \ L = 14in, \ E = 30 \times 10^6psi, \ G = 12 \times 10^6psi$

$\tau_{max} = 13600psi, \ \sigma_{max} = 3000psi, \ \delta_{max} = 0.25in$

In this case we adopt the fifth map and Table 10 illustrates the results of SCGSA. Obviously the results of SCGSA is better than CGSA.

7. Conclusion and future work

To improve the exploration and exploitation capability of CGSA, this paper proposes a hybridization method with Chaotic maps and sine functions to enhance the random searching capability of exploration and exploitation. After evaluating on CEC 2014 test functions and some real engineering optimization examples, the proposed SCGSA has its strengths in its the exploration and exploitation. The major **contributions** can be outlined as follows:

- Sine-cosine algorithm is adopted to enhance the movement pattern of agents in SCGSA;
- Parameter k is applied to enhance the convergence capability for the proposed SCGSA;
- The proposed SCGSA has good computational results in CEC 2014;
- SCGSA has good computational results in some real constrained continuous optimization problems when compared with CGSA;

Although the proposed SCGSA has many merits as a new hybrid optimization algorithm, it still has some **limitations** as follows:

- As evaluated by CEC 2014, SCGSA has advantage in most test functions, such as unimodal, multimodal, hybrid and composition functions, but it still has weakness in the test functions of F29 and F30;
- The sensibility of parameter k is tested in this paper, the k value should not be fixed in advance, and it needs to be determined automatically.

In the **future**, many movement patterns with sine or cosine functions can be proposed to improve the hybridization mechanism in the SCGSA algorithm. More and more hybridization methods can be proposed and implemented to achieve more efficient and effective hybrid SCGSA variants. In addition, an automatic k mechanism should be proposed as new SCGSA variant.

References

References

Belegundu, A. D. (1985). A study of mathematical programming methods for structural optimization. *International Journal for Numerical Methods in Engineering*, *21*, 1601–1623.

Blum, C. (2005). Ant colony optimization: introduction and recent trends. *Physics of Life Reviews*, *2*, 353–373. doi:10.1016/j.plrev.2005.10.001.

Chen, H. Q., Li, S., & Zheng, T. (2011). Hybrid gravitational search algorithm with randomkey encoding scheme combined with simulated annealing. *International Journal of Computer Science and Network Security*, *11*, 208–217.

Coello, C. A. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, *41*, 113–127.

Fister, I., Yang, X., Brest, J., & Fister, D. (2013). A brief review of nature-inspired algorithms for optimization. *arXiv: Neural and Evolutionary Computing*, .

Glover, F. (1990). Tabu search: a tutorial. *Interfaces*, *20*, 74–94. doi:10.1287/inte.20.4.74.

Guo, Z., Cheng, B., Ye, M., & Cao, B. (2006). Self-adaptive chaos differential evolution. In *International Conference on Advances in Natural Computation* (pp. 972–975). doi:10.1007/11881070_128.

Holliday, D., Resnick, R., & Walker, J. (1972). Fundamentals of physics. *Physics Today*, *25*, 53–54.

Ibrahim, R. A., Oliva, D., Ewees, A., & Lu, S. (2017). Feature selection based on improved runner-root algorithm using chaotic singer map and opposition-based learning. In *International Conference on Neural Information Processing* (pp. 156–166). doi:10.1007/978-3-319-70139-4_16.

Jiang, J., Chen, Y., Hao, D., & Li, K. (2019a). DPC-LG: Density peaks clustering based on logistic distribution and gravitation. *Physica A-statistical Mechanics and Its Applications*, *514*, 25–35. doi:10.1016/j.physa.2018.09.002.

Jiang, J., Feng, Y., Zhao, J., & Li, K. (2017). DataABC: A fast abc based energy-efficient live vm consolidation policy with data-intensive energy evaluation model. *Future Generation Computer Systems*, *74*, 132–141. doi:10.1016/j.future.2016.05.013.

Jiang, J., Hao, D., Chen, Y., Parmar, M., & Li, K. (2018a). GDPC: Gravitation-based density peaks clustering algorithm. *Physica A-Statistical Mechanics and Its Applications*, *502*, 345–355. doi:10.1016/j.physa.2018.02.084.

Jiang, J., Tao, X., & Li, K. (2018b). DFC: Density fragment clustering without peaks. *Journal of Intelligent & Fuzzy Systems*, *34*, 525–536. doi:10.3233/JIFS-17678.

Jiang, J., Wu, D., Chen, Y., & Li, K. (2019b). Complex network oriented artificial bee colony algorithm for global bi-objective optimization in three-echelon supply chain. *Applied Soft Computing*, *76*, 193–204. doi:10.1016/j.asoc.2018.12.006.

Jiang, J., Wu, D., Chen, Y., Wang, L., & Li, K. (2019c). Fast artificial bee colony algorithm with complex network and naive bayes classifier for supply chain network management. *Soft Computing*, (pp. 1–17). doi:10.1007/s00500-019-03874-y.

Jobin, C. M., Lakshmi, N. A., & Rahul, K. (2017). Detection of septic arthritis using meta heuristic algorithms. *Journal of Arthritis*, *6*, 1–2. doi:10.4172/2167-7921.1000259.

Jothiprakash, V., & Arunkumar, R. (2013). Optimization of hydropower reservoir usingevolutionary algorithms coupled with chaos. *Water Res. Manage.*, *27*, 19631979. doi:10.1007/s11269-013-0265-8.

Kennedy, J. (2011). Particle swarm optimization. *Proc. of 1995 IEEE Int. Conf. Neural Networks, (Perth, Australia), Nov. 27-Dec.*, *4*, 1942–1948. doi:10.1007/978-0-387-30164-8_630.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*, 671–680. doi:10.1126/science.220.4598.671.

Kumar, J. V., Kumar, D. M. V., & Edukondalu, K. (2013). Strategic bidding using fuzzy adaptive gravitational search algorithm in a pool based electricity market. *Applied Soft Computing*, *13*, 2445–2455. doi:10.1016/j.asoc.2012.12.003.

Li, Y. T., Deng, S. J., & Xiao, D. (2011). A novel hash algorithm construction based on chaotic neural network. *Neural Computing and Applications*, *20*, 133–141. doi:10.1007/s00521-010-0432-2.

Mallick, S., Ghoshal, S. P., Acharjee, P., & Thakur, S. S. (2013). Optimal static state estimation using improved particle swarm optimization and gravitational search algorithm. *International Journal of Electrical Power & Energy Systems*, *52*, 254–265.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, *21*, 1087–1092. doi:10.1063/1.1699114.

Mirjalili, S. (2015). Moth-flame optimization algorithm. *Knowledge based Systems*, *89*, 228–249. doi:10.1016/j.knosys.2015.07.006.

Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge based Systems*, *96*, 120–133. doi:10.1016/j.knosys.2015.12.022.

Mirjalili, S., & Gandomi, A. H. (2017). Chaotic gravitational constants for the gravitational search algorithm. *Applied Soft Computing*, *53*, 407–419. doi:10.1016/j.asoc.2017.01.008.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61. doi:10.1016/j.advengsoft.2013.12.007.

Onwubolu, G. C., & Babu, B. V. (2004). *New Optimization Techniques in Engineering*. Springer-Verlag Berlin Heidelberg.

Palanikkumar, D., Anbuselvan, P., & Rithu, B. (2012). A gravitational search algorithm for effective web service selection for composition with enhanced qos in soa. *International Journal of Computer Applications*, *42*, 12–15. doi:10.5120/5711-7768.

Rao, R. V., Savsani, V., & Vakharia, D. P. (2011). Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-aided Design*, *43*, 303–315. doi:10.1016/j.cad.2010.12.015.

Rashedi, E., Nezamabadipour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information Sciences*, *179*, 2232–2248. doi:10.1016/j.ins.2009.03.004.

Rodriguez, A., Gutierrez, A., Rivera, L., & Ramirez, L. (2015). Rwa: Comparison of genetic algorithms and simulated annealing in dynamic traffic. *Lecture Notes in Electrical Engineering*, *315*, 3–14.

Saremi, S., Mirjalili, S. M., & Mirjalili, S. (2014). Chaotic krill herd optimization algorithm. *Procedia Technol.*, *12*, 180185. doi:10.1016/j.protcy.2013.12.473.

Simpson, A. R., Dandy, G. C., & Murphy, L. J. (1994). Genetic algorithms compared to other techniques for pipe optimization. *Journal of Water Resources Planning and Management*, *120*, 423–443. doi:10.1061/(asce)0733-9496(1994)120:4(423).

Teh, J. S., Samsudin, A., & Akhavan, A. (2015). Parallel chaotic hash function based on the shuffle-exchange network. *Nonlinear Dynamics*, *81*, 1067–1079. doi:10.1007/s11071-015-2049-6.

Wang, G., Guo, L., Gandomi, A. H., Hao, G., & Wang, H. (2014). Chaotic krill herd algorithm. *Information Sciences*, *274*, 17–34. doi:10.1016/j.ins.2014.02.123.

Wang, N., Liu, L., & Liu, L. (2001). Genetic algorithm in chaos. *OR Trans*, *5*, 110.

Yang, X. (2009). Firefly algorithms for multimodal optimization. *Mathematics*, *5792*, 169–178. doi:10.1007/978-3-642-04944-6_14.

Yang, X., & Gandomi, A. H. (2012). Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*, *29*, 464–483.

Y.Jiang, & C.Lun (2002). Application of chaos in genetic algorithms. *Commun.Theor. Phys.*, *38*, 168. doi:10.1088/0253-6102/38/2/168.

8. Appendixes

8.1. Chaotic maps

Table 11: Chaotic maps for SCGSA

Order	Name	Equation	Range
1	Chebyshev (Wang et al., 2001)	$x_{i+1} = cos(icos^{-1}(x_i))$	$(-1, 1)$
2	Circle (Y.Jiang & C.Lun, 2002)	$x_{i+1} = mod(x_i + b - (\frac{a}{2\pi})sin(2\pi x_i), 1), a = 0.5, b = 0.2$	$(0, 1)$
3	Gauss/ mouse (Jothiprakash & Arunkumar, 2013)	$x_{i+1} = \begin{cases} 1, x_i = 0 \\ \frac{1}{mod(x_i,1)}, otherwise \end{cases}$	$(0, 1)$
4	Iterative (Guo et al., 2006)	$x_{i+1} = sin(\frac{a\pi}{x_i}), a = 0.7$	$(-1, 1)$
5	Logistic (Guo et al., 2006)	$x_{i+1} = ax_i(1 - x_i), a = 4$	$(0, 1)$
6	Piecewise (Saremi et al., 2014)	$x_{i+1} = \begin{cases} \frac{x_i}{P}, 0 \leq x_i < P \\ \frac{x_i - P}{0.5 - P}, P \leq x_i < 0.5, P = 0.4 \\ \frac{1 - P - x_i}{0.5 - P}, 0.5 \leq x_i < 1 - P \\ \frac{1 - x_i}{P}, 1 - P \leq x_i < 1 \end{cases}$	$(0, 1)$
7	Sine (Wang et al., 2014)	$x_{i+1} = \frac{a}{4} sin(\pi x_i), a = 4$	$(0, 1)$
8	Singer (Ibrahim et al., 2017)	$x_{i+1} = \mu(7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.302875x^4), \mu = 2.3$	$(0, 1)$
9	Sinusoidal (Li et al., 2011)	$x_{i+1} = ax_i^2 sin(\pi x_i), a = 2.3$	$(0, 1)$
10	Tent (Teh et al., 2015)	$x_{i+1} = \begin{cases} \frac{x_i}{0.7}, x_i < 0.7 \\ \frac{10}{3}(1 - x_i), x_i \geq 0.7 \end{cases}$	$(0, 1)$

8.2. Total convergence curves

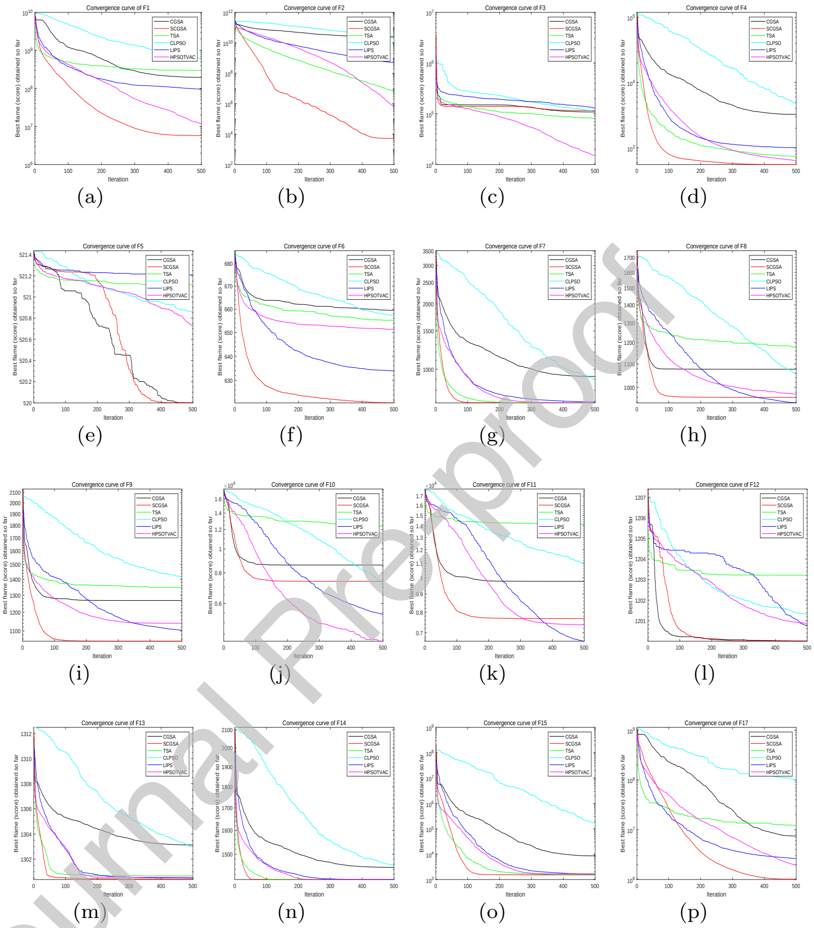
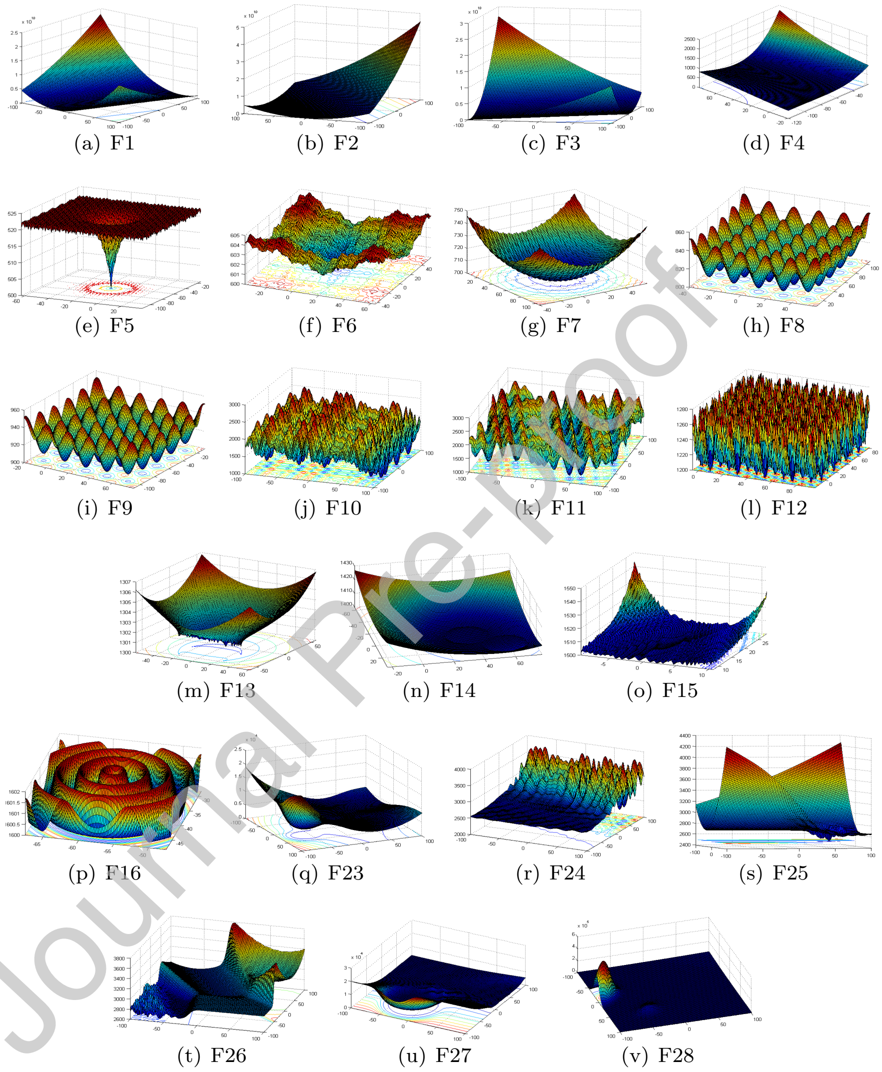


Figure 9: Convergence curves with map9

8.3. Plots of functions in 2-D



(a) F1 (b) F2 (c) F3 (d) F4

(e) F5 (f) F6 (g) F7 (h) F8

(i) F9 (j) F10 (k) F11 (l) F12

(m) F13 (n) F14 (o) F15

(p) F16 (q) F23 (r) F24 (s) F25

(t) F26 (u) F27 (v) F28

8.4. Basic functions

Table 12: Basic functions

Function name	Function formulation
Rotated High Conditioned	
Elliptic Function	$f_1(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$
Bent Cigar Function	$f_2(x) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$
Discus Function	$f_3(x) = 10^6 x_1^2 + \sum_{i=2}^{D} x_i^2$
Rosenbrocks Function	$f_4(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1))$
Ackleys Function	$f_5(x) = -20exp(-0.2\sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2}) - exp(\frac{1}{D} \sum_{i=1}^{D} cos(2\pi_i)) + 20 + e$
Weierstrass Function	$f_6(x) = \sum_{i=1}^{D} (\sum_{k=0}^{kmax} [a^k cos(2\pi b^k (x_i + 0.5))]) - D \sum_{k=0}^{kmax} [a^k cos(2\pi b^k \cdot 0.5)]$
	$a = 0.5, b = 3, kmax = 20$
Griewanks Function	$f_7(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} cos(\frac{x_i}{\sqrt{i}}) + 1$
Rastrigins Function	$f_8(x) = \sum_{i=1}^{D} (x_i^2 - 10cos(2\pi x_i) + 10)$
Modified Schwefels Function	$f_9(x) = 418.9829 \times D - \sum_{i=1}^{D} g(z_i)$
	$g(z_i) = \begin{cases} z_i sin(\lvert z_i \rvert^{1/2}), \lvert z_i \rvert \leq 500 \\ (500 - mod(z_i, 500)) sin(\sqrt{\lvert 500 - mod(z_i, 500) \rvert}) - \frac{(z_i - 500)^2}{10000D}, z_i > 500 \\ (mod(\lvert z_i \rvert, 500) - 500) sin(\sqrt{\lvert mod(\lvert z_i \rvert, 500) - 500 \rvert}) - \frac{(z_i + 500)^2}{10000D}, z_i < -500 \end{cases}$
Katsuura Function	$f_{10}(x) = \frac{10}{D^2} \prod_{D}^{i=1} (1 + i \sum_{j=1}^{32} \frac{\lvert 2^j x_i - round(2^j x_i) \rvert}{2^j})^{\frac{10}{D^{12}}} - \frac{10}{D^2}$
HappyCat Function	$f_{11}(x) = \lvert \sum_{i=1}^{D} x_i^2 - D \rvert^{\frac{1}{4}} + (0.5 \sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i)/D + 0.5$
HGBat Function	$f_{12}(x) = \lvert (\sum_{i=1}^{D} x_i^2)^2 - \sum_{i=1}^{D} x_i)^2 \rvert^{\frac{1}{2}} + (0.5 \sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i)/D + 0.5$
Expanded Griewanks plus	
Rosenbrocks Function	$f_{13}(x) = f_7(f_4(x_1, x_2)) + f_7(f_4(x_2, x_3)) + \ldots + f_7(f_4(x_{D-1}, x_D)) + f_7(f_4(x_D, x_1))$
Expanded Scaffers F6 Function	
	Scaffer's F6 Function: $g(x, y) = 0.5 + \frac{sin^2(\sqrt{x^2+y^2})-0.5}{(1+0.001(x^2+y^2))^2}$
	$f_{14}(x) = g(x_1, x_2) + g(x_2, x_3) + \ldots + g(x_{D-1}, x_D) + g(x_D, x_1)$

8.5. Test functions

Table 13: Test functions

Function name	Function formulation
Unimodal functions	Note: M-rotation matrix
High Conditioned Elliptic Function	$F_1(x) = f_1(M(x - o_1)) + 100$
Rotated Bent Cigar Function	$F_2(x) = f_2(M(x - o_2)) + 200$
Rotated Discus Function	$F_3(x) = f_3(M(x - o_3)) + 300$
Multimodal functions	
Shifted and Rotated Rosenbrocks Function	$F_4(x) = f_4(M(\frac{2.048(x-o_4)}{100}) + 1) + 400$
Shifted and Rotated Ackleys Function	$F_5(x) = f_5(M(x - o_5)) + 500$
Shifted and Rotated Weierstrass Function	$F_6(x) = f_6(M(\frac{0.5(x-o_6)}{100})) + 600$
Shifted and Rotated Griewanks Function	$F_7(x) = f_7(M(\frac{600(x-o_7)}{100})) + 700$
Shifted Rastrigins Function	$F_8(x) = f_8(\frac{5.12(x-o_8)}{100}) + 800$
Shifted and Rotated Rastrigins Function	$F_9(x) = f_8(M(\frac{5.12(x-o_9)}{100})) + 900$
Shifted Schwefels Function	$F_{10}(x) = f_9(\frac{1000(x-o_{10})}{100}) + 1000$
Shifted and Rotated Schwefels Function	$F_{11}(x) = f_9(M(\frac{1000(x-o_{11})}{100})) + 1100$
Shifted and Rotated Katsuura Function	$F_{12}(x) = f_{10}(M(\frac{5(x-o_{12})}{100})) + 1200$
Shifted and Rotated HappyCat Function	$F_{13}(x) = f_{11}(M(\frac{5(x-o_{13})}{100})) + 1300$
Shifted and Rotated HGBat Function	$F_{14}(x) = f_{12}(M(\frac{5(x-o_{14})}{100})) + 1400$
Shifted and Rotated Expanded Griewanks plus Rosenbrocks Function	$F_{15}(x) = f_{13}(M(\frac{5(x-o_{15})}{100} + 1)) + 1500$
Shifted and Rotated Expanded Scaffers F6 Function	$F_{16}(x) = f_{14}(M(x - o_{16}) + 1) + 1600$
Hybrid Functions	
$F17 = f_9(M_1 z_1) + f_8(M2z_2) + f_1(M_3 z_3) + 1700$	$p = [0.3, 0.3, 0.4]$
$F18 = f_2(M_1 z_1) + f_{12}(M2z_2) + f_8(M_3 z_3) + 1800$	$p = [0.3, 0.3, 0.4]$

$F19 = f_7(M_1 z_1) + f_6(M2 z_2) + f_4(M_3 z_3) + f_{14}(M_4 z_4) + 1900$	$p = [0.2, 0.2, 0.3, 0.3]$
$F20 = f_{12}(M_1 z_1) + f_3(M2 z_2) + f_{13}(M_3 z_3) + f_8(M_4 z_4) + 2000$	$p = [0.2, 0.2, 0.3, 0.3]$
$F21 = f_{14}(M_1 z_1) + f_{12}(M2 z_2) + f_4(M_3 z_3) + f_9(M_4 z_4) + f_1(M_5 z_5) + 2100$	$p = [0.1, 0.2, 0.2, 0.2, 0.3]$
$F22 = f_{10}(M_1 z_1) + f_{11}(M2 z_2) + f_{13}(M_3 z_3) + f_9(M_4 z_4) + f_5(M_5 z_5) + 2200$	$p = [0.1, 0.2, 0.2, 0.2, 0.3]$

Notes:

$z_1 = [y_{S_1}, y_{S_2}, \ldots, y_{S_{n1}}]$,

$z_2 = [y_{S_{n1+1}}, y_{S_{n1+2}}, \ldots, y_{S_{n1+n2}}]$,

$z_N = [y_{S_{\sum_{i=1}^{N-1} n_i+1}}, y_{S_{\sum_{i=1}^{N-1} n_i+2}}, \ldots, y_{S_D}]$

$y = x - o_i, S = randperm(1:D), p_i : percentage of g_i(x)$

$n_1 = \lceil p_1 D \rceil, n_2 = \lceil p_2 D \rceil, \ldots, n_{N-1} = \left\lceil p_{N-1} D \right\rceil, n_N = D - \sum_{i=1}^{N-1} n_i$

Composition Functions

$$F_{23} = \omega_1 * F_4'(x) + \omega_2 * \left[1e^{-6} F_1'(x) + 100\right] + \omega_3 * \left[1e^{-26} F_2'(x) + 200\right] + \omega_4 * \left[1e^{-6} F_3'(x) + 300\right] + \omega_5 * \left[1e^{-6} F_1'(x) + 400\right] + 2300$$

$\sigma = [10, 20, 30, 40, 50]$

$$F_{24} = \omega_1 * F_{10}'(x) + \omega_2 * \left[F_9'(x) + 100\right] + \omega_3 * \left[F_{14}'(x) + 200\right] + 2400$$

$\sigma = [20, 20, 20]$

$$F_{25} = \omega_1 * 0.25 F_{11}'(x) + \omega_2 * \left[F_9'(x) + 100\right] + \omega_3 * \left[1e^{-7} F_1'(x) + 200\right] + 2500$$

$\sigma = [10, 30, 50]$

$$F_{26} = \omega_1 * 0.25 F_{11}'(x) + \omega_2 * \left[F_{13}'(x) + 100\right] + \omega_3 * \left[1e^{-7} F_1'(x) + 200\right] + \omega_4 * \left[2.5 F_6'(x) + 300\right] + \omega_5 * \left[10 F_7'(x) + 400\right] + 2600$$

$\sigma = [10, 10, 10, 10, 10]$

$$F_{27} = \omega_1 * 10 F_{14}'(x) + \omega_2 * \left[10 F_9'(x) + 100\right] + \omega_3 * \left[2.5 F_{11}'(x) + 200\right] + \omega_4 * \left[25 F_6'(x) + 300\right] + \omega_5 * \left[1e^{-6} F_1'(x) + 400\right] + 2700$$

$\sigma = [10, 10, 10, 20, 20]$

$$F_{28} = \omega_1 * 2.5 F_{15}'(x) + \omega_2 * \left[10 F_{13}'(x) + 100\right] + \omega_3 * \left[2.5 F_{11}'(x) + 200\right] + \omega_4 * \left[5e^{-4} F_{16}'(x) + 300\right] + \omega_5 * \left[1e^{-6} F_1'(x) + 400\right] + 2800$$

$\sigma = [10, 20, 30, 40, 50]$

$$F_{29} = \omega_1 * F_{17}'(x) + \omega_2 * \left[F_{18}'(x) + 100\right] + \omega_3 * \left[F_{19}'(x) + 200\right] + 2900$$

$\sigma = [10, 30, 50]$

$$F_{30} = \omega_1 * F_{20}'(x) + \omega_2 * \left[F_{21}'(x) + 100\right] + \omega_3 * \left[F_{22}'(x) + 200\right] + 3000$$

$\sigma = [10, 30, 50]$

Notes:

$$\omega_i = \frac{1}{\sqrt{\sum_{j=1}^{D}(x_j - o_{ij})^2)}} exp(-\frac{\sum_{j=1}^{D}(x_j - o_{ij})^2}{2D\sigma_i^2})$$

Author Statement

Jianhua Jiang: Supervision, Conceptualization, Methodology.

Ran Jiang: Software, Test, Writing- Original draft preparation.

Xianqiu Meng: Visualization, Validation, Writing- Final draft preparation.

Keqin Li: Supervision, Writing- Reviewing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.