Expert Systems With Applications 113 (2018) 515–530

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A hybrid algorithm based on self-adaptive gravitational search algorithm and differential evolution

Fuqing Zhao[a,*], Feilong Xue[a], Yi Zhang[b], Weimin Ma[c], Chuck Zhang[d], Houbin Song[a]

[a] School of Computer and Communication Technology, Lanzhou University of Technology, Lanzhou 730050, China
[b] School of Mechanical Engineering, Xijin University Xi'an 710123, China
[c] School of Economics and Management, Tongji University, Shanghai 200092, China
[d] H. Milton Stewart School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

ARTICLE INFO

Article history:
Received 2 March 2018
Revised 22 May 2018
Accepted 2 July 2018
Available online 21 July 2018

Keywords:
Gravitational search algorithm
Differential evolution
Self-adaptive mechanism
Crossover and mutation operation
Exploration and exploitation

ABSTRACT

The Gravitational Search Algorithm (GSA) has excellent performance in solving various optimization problems. However, it has been demonstrated that GSA tends to trap into local optima and are easy to lose diversity in the late evolution process. In this paper, a new hybrid algorithm based on self-adaptive Gravitational Search Algorithm (GSA) and Differential Evolution (DE) is proposed for solving single objective optimization, named SGSADE. Firstly, a self-adaptive mechanism based on GSA is proposed for improving the convergence speed and balancing exploration and exploitation. Secondly, the diversity of the population is maintained in the evolution process by using crossover and mutation operation from DE. Besides, to improve the performance of the algorithm, a new perturbation based on Levy flight theory is embedded to enhance exploitation capacity. The simulated results of SGSADE on 2017 CEC benchmark functions show that the SGSADE outperforms the state-of-the-art variant algorithms of the GSA.

1. Introduction

Single-objective real parameter optimization problem is an important issue that has been widely studied in the past decades. Many real word optimization problems include complex nonlinear functions and other different domains (engineering, data clustering, etc.) can be formulated as a continuous function optimization problem. The challenge task for solving single-objective real parameter optimization problem is quite complex and time consuming due to its large-scale. Without loss of generality, the single-objective global optimization problem can be defined as follows, for a function $f = f(x_1, x_2, \ldots, x_D)$, we need to find variables of vector x_i in specified domain D, which maximizes/minimizes f (Brest, Maučec, & Bošković, 2017).

In the past decades, many researchers have proposed a wide variety of methods for solving complex optimization problem. However, traditional mathematical methods are not perform well in solving the complex optimization problem due to non-linear, non-convex and require derivable fitness functions, so evolutionary algorithms are applied to overcome the drawbacks of traditional mathematical methods for solving optimization problem by a mounting number of researchers. Evolutionary algorithms are inspired by biological systems and physical processes (Yang, 2018) include particle swarm optimization (PSO) (Kennedy & Eberhart, 1995), water wave optimization (WWO) (Zhao, Liu, Zhang, Ma, & Zhang, 2017; Zheng, 2015), ant colony optimization (ACO) (Dorigo, Birattari, & Stutzle, 2004), differential evolution algorithm (DE) (Storn & Price, 1997) and other typical hybrid evolutionary algorithms (Zhao, et al., 2016; Zhao, Liu, Zhang, & Wang, 2015; Zhao, Liu, Zhang, Ma, & Zhang, 2017; Zhao, Shao, Wang, & Zhang, 2017; Zhao, Zhang, Zhang, & Wang, 2015). Evolutionary algorithms play a significant role in solving optimization problems, especially in complex problems. The key success of evolutionary algorithms is to balance exploration and exploitation at different evolutionary stages. However, most of evolutionary algorithms only perform well either exploration or exploitation. Consequently, those algorithms may fall into local optimum in solving the optimization problem.

Gravitational search algorithm (GSA) (Rashedi, Nezamabadi-Pour, & Saryazdi, 2009) is a new population-based intelligent optimization algorithm, which is based on Newton's law of gravitation and motion. In GSA, each agent corresponds to the planet in the universe, and the fitness value of agent corresponds to the mass of the planet. The heavier mass means a more efficient agent in the population. The global movement will be produced by the

* Corresponding author:
E-mail addresses: Fzhao2000@hotmail.com (F. Zhao), 767098126@qq.com (F. Xue), 1049440546@qq.com (Y. Zhang), mawm@tongji.edu.cn (W. Ma), chuck.zhang@isye.gatech.edu (C. Zhang), 523712418@qq.com (H. Song).

https://doi.org/10.1016/j.eswa.2018.07.008
0957-4174/

interaction between agents. Compared with other evolutionary algorithms, GSA has many advantages such as simple theory, few control parameters, and retains a unique way of balancing exploration and exploitation in search process. Therefore, the performance of GSA in solving continuous optimization problem is better than PSO and Genetic Algorithm (GA) (Haupt & Haupt, 2004) on convergence speed and solution accuracy. Due to the excellent performance of GSA, it has been successfully applied to solve different optimization problems, such as data clustering (Han, et al., 2017), virtual enterprises (Xiao, Niu, Chen, Leung, & Xing, 2016), short term hydrothermal scheduling (Gouthamkumar, Sharma, & Naresh, 2015), design of concrete gravity dams (Khatibinia & Khosravi, 2014) and others (Goel, Singhal, Mishra, & Mohanty, 2017; Jiang, Wang, & Ji, 2014; Soleimanpour-Moghadam, Nezamabadi-Pour, & Farsangi, 2014). However, the main drawbacks of the GSA are obvious and the details as follows: First, Kbest strategy is introduced in GSA, i.e., all agents are affected by Kbest agents. The diversity of the population rapidly declines in this model, and the premature phenomenon is appeared (Sun, et al., 2016). Second, in the late evolution process, the mass of all agents tends to be consistent. In this case, there are no remedies to jump out once the population fall into local optima. Finally, the exploitation ability of the GSA is demonstrated to be highly sensitive with the value of G (Ji, et al., 2017), yet the parameters $G0$ and α which determine the G are set to a fixed value. To overcome the drawbacks of the GSA, various operators have been used to improve the search capacity by researchers.

The improvement of the GSA mainly includes two aspects: (1) Changes based on the behavior of the GSA (such as parameters settings, introduction of disturbance and others); (2) Hybrid with other evolutionary algorithms.

The Kbest strategy is a fully informed learning strategy. It overlooks the impact of environmental heterogeneity on individual behavior, which resulting in premature convergence and time consuming. Therefore, Sun, et al. (2016) employed a locally informed learning strategy to replace global learning strategy which promotes exploitation, called LIGSA. In the LIGSA, each agent learns from k local neighbors and the historically global best agent rather than from Kbest agents. Learning from the k local neighbors effectively prevents premature convergence and the guidance of global best agent can accelerate the convergence speed. Experimental results of evaluating on 30 CEC2014 benchmark functions show that LIGSA outperforms the compared algorithms. A dynamic neighborhood learning strategy to replace the Kbest model is proposed by A. Zhang, et al. (2016) with aim to balance exploration and exploitation in the search process. The local and global neighborhood topologies integration not only enhance the convergence speed, but also balance exploration and exploitation. Mittal, Pal, Kulhari, and Saraswat (2017) introduced a novel chaotic Kbest strategy which uses logistic mapping to computer the values of Kbest to balance exploration and exploitation. The parameters setting significantly affect the performance of GSA, but determining the proper parameters is a huge challenge. Ji, et al. (2017) analyzed the sensitivity of gravitational coefficient G for the convergence speed and solution quality, a self-adaptive mechanism to adjust the value of G and a modified chaotic local search algorithm (IGSA) is proposed. Mirjalili and Gandomi (2017) embedded ten chaotic maps into the gravitational constant G to improve the performance of GSA and an adaptive normalization method is proposed to transform from the exploration phase to the exploitation phase smoothly. The results of Wilcoxon rank-sum show that sinusoidal map is the best map in the ten chaotic maps for improving the performance of the GSA. Jordehi (2017) proposed a linearly decreasing gravitational constant with the GSA which a linearly decreased formula replace exponentially decreased formula to generate gravitational constant G for solving photovoltaic power (PV) cell parameter estimation problem. A simplified gravitational search algorithm (SGSA) is proposed by Zhang, Zou, and Shen (2017) to overcome the shortcoming that the GSA is easy to fall into the local optima. In SGSA, the velocity of the agent is discarded, and the position update formula only includes the agent's acceleration. Pelusi, et al. (2018) proposed a revised GSA (NFGSA) with intelligent adaption of parameters. NFGSA adjusts the most relevant parameters of GSA through neural network and fuzzy systems to avoid the candidates fall into local optima and lead to premature convergence. Experiment results show that NFGSA not only outperforms the compared algorithms on test functions, but also has better computational complexity than the compared algorithms. In Sun, Ma, Ren, Zhang, and Jia (2018), a stability constrained adaptive α for GSA which the value of α is adaptively adjusted through combining with the variation of the agent's position and fitness feedback is proposed. The novel α adjusting method accelerates the convergence speed, alleviates the premature problem and ensures the stable convergence.

Hybridization with other evolutionary algorithms is the other aspect to improve the performance of the GSA. Naik and Panda (2015) integrated the Cuckoo Search into the GSA which the update formula of agent is modified for function minimization problem to enhance the exploration capability of the GSA. Xiao, et al. (2016) combined GSA and PSO (I-GSA/PSO) to solve the green partner selection problem (G-PSP) in virtual enterprises. In I-GSA/PSO, the update formula of velocity and position are modified by utilizing the supplementary potentialities of PSO and GSA. Experimental results show that I-GSA/PSO outperforms the compared algorithms for solving G-PSP. Yin, Guo, Liang, and Yue (2017) introduced a modified GSA with crossover from DE called CROGSA. The crossover-based search scheme to update the position of each agent and take full advantage of the information of global optimal position are executed for enhancing the exploitation capability in CROGSA. In Ismail, et al. (2017), an improved hybrid of particle swarm optimization and the gravitational search algorithm is proposed to improve the efficiency of a global optimum search, called IGSAPSO. IGSAPSO improves the convergence speed by exploiting the feasible solution areas to narrow down the search space. Kang, Bae, Yeung, and Chung (2018) presented the hybrid GSA (HGSA) which combines the fast convergence property of PSO and fully utilizing all current information of GSA to enhance the utilization of particle information and to facilitate thorough search inside the video frame before convergence. In Li, Lin, Tseng, Tan, and Ming (2018), a maximum power point tracking (MPPT) method with improved GSA which the related factor of memory and population information exchange are added into the updating formula of particle velocity is proposed for photovoltaic system, called IGSA-MPPT. In IGSA-MPPT, the adaptive mechanism of the gravitational constant is added to balance exploitation and exploration.

Although the performance of GSA has been improved through the aforementioned methods, there are still some intrinsic problems in the original GSA. Therefore, a new hybrid algorithm SGSADE is proposed that tries to achieve balance exploration and exploitation. Usually, parameters α and $G0$ are set to a fixed value. In SGSADE, a new self-adaptive adjusting scheme for parameters α and $G0$ is designed which balance exploitation and exploration and accelerate convergence speed. Many evolutionary algorithms are combined with the GSA to enhance the local search ability in the literature. In this paper, the crossover and mutation operation from DE is utilized to enhance exploitation of the GSA and maintain the diversity of the population in the evolution process. In SGSADE, Kbest strategy from the original GSA is adopted. Additionally, a disturbance based on Levy flight is employed to enhance the local search ability. In summary, SGSADE has better performance than the state-of-art variant of GSA by testing on CEC 2017

Algorithm 1 The standard GSA.

Randomly initialize the population in the search space
$t = 0$
While $(t < T)$
  Evaluate the fitness of all agents
  Update $worst(t)$, $best(t)$ and $M_i(t)$ for $i = 1 \ldots N$ based on Eqs. (2)–(5)
  Update $G(t)$ based on Eq. (7)
  Calculation of the total force in different dimension based on Eqs. (6) and (8)
  Update acceleration and velocity of all agents based on Eqs. (9) and (10)
  Update agent's position based on Eq. (11)
End While
Output Result

benchmark functions with various dimensions. In this research, the main contributions could be stated as follows:

- A self-adaptive mechanism of parameters α and $G0$ is introduced in GSA. In search process, the gravitational coefficient G of each agent will be adjusted according to the fitness of each agent through parameters α and $G0$.
- Crossover and mutation operations from DE are introduced in GSA during the evolution process. In this research, the positional update formula of agents is modified through crossover and mutation operations. The new scheme is added to enhance the exploitation capacity and maintain the diversity of the population during the evolutionary process.
- A disturbance based on Levy flight is embedded in GSA. Due to the random property of Levy flight, the exploitation capacity of SGSADE is improved.

The remainder of this paper is organized as follows. The basic GSA is introduced in Section 2. Section 3 gives a detailed introduction of the proposed SGSADE. The experiment and comparisons are presented in Section 4. A conclusion is given in Section 5.

2. Gravitational search algorithm

In this paper, the notation is described as follows:

X_i	the position of the tth agent
D	the dimension of the search space
x_i^d	the position of ith agent in the dth dimension
v_i^d	the velocity of ith agent in the dth dimension
N	the number of the population
T	the maximum number of iteration
t	the current iteration
$\mathrm{fit}_i(t)$, $M_i(t)$	the fitness and the mass of the ith agent at the current tth iteration
M_i	the mass of ith agent
$R_{i,j}(t)$	the Euclidian distance between agent i and agent j at the current tth iteration
ε	a small constant
$G0$	the initial value of the gravitational constant
$G(t)$	the value of the gravitational constant at tth iteration
α	the shrinking parameter
a_i	the acceleration of ith agent
$Kbest$	the set of first $Kbest$ agents with the best fitness value and the biggest mass
$randc(\mu, \sigma^2)$	the values randomly selected from Cauchy distributions with mean μ and variance σ^2
$randn(\mu, \sigma^2)$	the values randomly selected from normal distributions with mean μ and variance σ^2
c	the learning rate (c is set to 0.1)
Γ	the Gamma Function
$sf(\bullet)$	the mutation factor
$jrand$	a decision variable index which is uniformly randomly selected from [1, D]
$cr(\bullet)$	the crossover rate and $cr(\bullet) \in [0, 1]$
H	the length of historical memory

The standard GSA is a novel meta-heuristic algorithm that is inspired by Newton's law of gravity and motion. In GSA, each agent presents a solution of the problem, and its mass is determined using a fitness function. In other words, the mass will present a solution in the search space. GSA implements the following strategies based on the law of gravity and law of motion:

2.1. Law of gravity

Each particle attracts every other particle through the gravitational force, and the magnitude of the gravitational force is proportional to the product of their masses, inversely proportional to the square of their distance**.**

2.2. Law of motion

The acceleration of the particle is proportional to the force acted on the particle, inversely proportional to the mass of the particle.

The main steps of the standard GSA are described in Algorithm 1. The GSA consists of the following main steps:

Step 1: Randomly initialize the population in the search space.

$$X_i = \left(x_i^1, \ldots, x_i^d, \ldots, x_i^D\right) \qquad for\ i = 1 \ldots N; \tag{1}$$

Step 2: The mass of each agent is calculated by the fitness of the current population. The equations are shown as follows:

$$q_i(t) = \frac{\mathrm{fit}_i(t) - worst(t)}{best(t) - worst(t)} \tag{2}$$

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

For a minimization problem, $worst(t)$ and $best(t)$ are defined as follows:

$$best\ (t) = \min_{j \in \{1 \ldots N\}} fit_j(t) \tag{4}$$

$$\mathrm{worst}(t) = \max_{j \in \{1 \ldots N\}} fit_j(t) \tag{5}$$

Step 3: At iteration time t, the force acting on tth agent from jth agent in the dth dimension is defined as follows:

$$F_{ij}^d(t) = G(t) \frac{M_i(t) \times M_j(t)}{R_{i,j}(t) + \varepsilon} \left(x_j^d(t) - x_i^d(t)\right) \tag{6}$$

$G(t)$ is shown as:

$$\mathrm{G(t)} = \mathrm{G0} \times e^{-\alpha \frac{t}{T}} \tag{7}$$

The total force of all agents acting on the ith agent in the dth dimension is defined as:

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i} rand_j F_{ij}^d(t) \tag{8}$$

At the beginning, all agents apply the force, and as time pass, *Kbest* is decreased linearly. There is only one agent applying force to the others at the end.

Step 4: Update acceleration, velocity and position.

According to the law of motion, the acceleration of the agent i at time t in dimension d is calculated as follows:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \tag{9}$$

Furthermore, the velocity of the ith agent and its position in the next iteration could be calculated as follows:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{10}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{11}$$

where $rand_i$ is a uniformly distributed random variable in the interval [0,1].

3. The proposed algorithm

3.1. Parameter self-adaptation schemes

In the GSA algorithm, the value of G has a significant influence on balancing exploration and exploitation in the optimization process. Based on Eqs. (6), (7) and (9), the value of G is determined by parameters α and $G0$. Therefore, α and $G0$ are the key parameter for balancing exploration and exploitation of the GSA (Sun, et al., 2018).

For enhancing the performance of GSA, a self-adaptive mechanism which can get information from current population to control the value of G through change the parameter α and $G0$ is proposed in the paper. In SGSADE, each agent has its corresponding α and $G0$.

The parameters are probabilistically set at the beginning of each generation according to control parameters $\mu\alpha$ and $\mu G0$, shown as follows:

$$\begin{aligned} \alpha_i &= randc_i(\mu\alpha, 0.1) \\ G0_i &= randc_i(\mu G0,\ 0.1) \end{aligned} \tag{12}$$

When $\alpha_i \le 0$ and $G0_i \le 0$, Eq. (12) is repeatedly applied until generate a valid value. The parameters are adapted during the search as follows.

In each generation, α_i and $G0_i$ will be recorded as S_α, S_{G0} when the parameters generate trail vector which is better than the parents. At the end of the generation, $\mu\alpha$ and $\mu G0$ are updated as:

$$\begin{aligned} \mu\alpha &= (1-c)\ \times \mu\alpha + c \times mean(S_\alpha) \\ \mu G0 &= (1-c)\ \times \mu G0 + c \times mean(S_{G0}) \end{aligned} \tag{13}$$

where *mean*(•)is an arithmetic mean.

3.2. Modified local search

A new disturbed mechanism (NDM) is incorporated into the search process with the aim to further improve the search performance of SGSADE, called Levy flight disturbance. The Levy flight provides random walk for step size from the Levy distribution. In Gandomi, Yang, and Alavi (2013), Levy flight has been applied to optimal search, and preliminary results show its promising capability. NDM is defined as follows:

$$X_{newbest}(t) = X_{best}(t) + (Levy(\lambda) - 0.5)\ \times\ (X_p(t) - X_q(t)) \tag{14}$$

where, $X_{best}(t)$ is the position of current best agent in the tth iteration. $X_{newbest}(t)$ is the newly generated agent. $X_p(t)$ and $X_q(t)$ are two different agents randomly selected from current population and $p \neq q \neq best$. $Levy\,(\lambda)$ is the random walk based on the Levy flight. In this paper, $Levy(\lambda)$ can be determined by

$$Levy(\lambda) = \frac{\varphi \times \mu}{|\upsilon|^{\frac{1}{\lambda}}}, \qquad \left(\varphi = \left\{ \frac{\Gamma(1+\lambda)\ \times sin\left(\pi \times \frac{\lambda}{2}\right)}{\Gamma\left\{\left[\frac{1+\lambda}{2}\right]\ \times \lambda \times 2^{\frac{\lambda-1}{2}}\right\}} \right\}^{\frac{1}{\lambda}}, \lambda = 1.5 \right) \tag{15}$$

where, μ and υ are obtained from the normal distribution. If the new agent $X_{newbest}(t)$ generated by NDM has a high fitness, it will replace the current best agent.

Inspired by Levy flight, the perturbation can enhance exploitation ability of the SGSADE.

3.3. Combined with self-adaptive DE algorithm

Differential Evolution (DE) algorithm is a stochastic, population based search method that was designed for numerical optimization problems. In recent years, DE has been shown to be competitive with other more complex optimization algorithms, and has been applied to solve real application problems. Pant, Thangaraj, and Abraham (2011) combined two phases global optimization algorithm include differential evolution (DE) and particle swarm optimization (PSO), called DE-PSO. The strengths of both the algorithms are preserved in DE-PSO. Biswas, Kundu, Bose, Das, and Suganthan (2013) proposed a modified affinity-based mutation framework based on the information conveyed by neighboring individuals. This framework is easily combined with DE and its state-of-the-art variants with minor changes. In Kundu, Biswas, Das, and Suganthan (2013) , a niching parameter free algorithm integrates with DE which can locate multiple optima in changing environment is designed for solving multimodal and dynamic problems, called CLDES. Experimental results show that CLDES outperforms other peer algorithms. In Biswas, Kundu, and Das (2015), three improved versions of DE-based methods with an efficient niching framework which an improved information-sharing mechanism among the individuals of an evolutionary algorithm are proposed for multimodal optimization problems. Zhao, Shao, Wang, and Zhang (2015) presented a new hybrid optimization algorithm called cDE/EDA which combines estimation of distribution algorithms (EDAs) and differential evolution (DE). Due to effective nature of harmonizing the strong exploration of EDAs with the good exploitation of DE, cDE/EDA can discover the optimal solution in a fast manner.

In summary, combining DE with other evolutionary algorithms can enhance the search ability and maintain the population diversity. Hence, combined with DE is applied in SGSADE. In the GSA, the acceleration and velocity are the key to update the position of each agent in the current population. In the early stages of evolution, the update mechanism of the position is beneficial to accelerate convergence speed. However, population diversity rapidly decline in the late of evolution process. In this paper, a mechanism based on the "current-to-pbest/1" and "rand/1" mutation and crossover is introduced into the GSA to improve the exploitation ability and maintain population diversity (Tanabe & Fukunaga, 2013). The trial variables based mutation mechanism is described as the following formula:

$$u_i(t) = x_i(t) + sf(i)\ \times\ (x_{best} - x_i(t) + x_{r_1}(t) - x_{r_2}(t)) \tag{16}$$

$$u_i(t) = x_{r_3}(t) + sf(i) \times (x_{r_4}(t) - x_{r_5}(t)) \tag{17}$$

x_{best} is the best agent in the tth iteration, $x_{r_1}^t$, $x_{r_2}^t$, $x_{r_3}^t$, $x_{r_4}^t$ and $x_{r_5}^t$ is different particle in the current population. In the evolution process, trial variables based GSA is generated as the following formula:

$$u_i(t) = rand_i \times x_i(t) + a_i\ \ (t) \tag{18}$$

Algorithm 2
SGSADE.

Initialize the population of N agents
Initialize the value of M_{G0},M_{α},M_{sf},M_{cr}
Evaluate the fitness of population
$t=0$
While $(t < T)$
    Generate a levy number using Eq. (15), then implement disturbance operation using Eq. (14)
    Generate the parameters α and G0 using Eq. (12)
    Calculate G_i of each agent using Eq. (7)
    Generate sf and cr using Eqs. (21) and (22)
    randi ← select from [1, H] randomly
    If $randi = H$ **then**
        $cr=0.2$
        $sf=0.9$
    End if
    Calculate the mass of the population using Eqs. (2) and (3)
    Calculate the total force with its own G_i using Eqs. (6) and (8)
    Calculate the acceleration of the population using Eq. (9)
    For $i=1:\ N$
        If $randa < 0.9 - 0.8 \times \frac{t}{T}$ **then** (*randa* is randomly selected from standard normal distributions)
            If $randb < 0.1 + 0.8 \times \frac{t}{T}$ **then** (*randb* is randomly selected from standard normal distributions)
                Using Eq. (16)
            Else
                Using Eq. (17)
            End if
        Else
            Using Eq. (18)
        End if
    End for
    Crossover operation using Eq. (19)
    Update the position of the population using Eq. (20)
    Update $\mu\alpha$, μG0, M_{sf} and M_{cr} using Eqs. (13), (23) and (24)
End while
Output Result

Table 1
The historical memory.

Index	1	2	…	*H-1*	*H*
M_{G0}	$M_{G0,1}$	$M_{G0,2}$	…	$M_{G0,H-1}$	$M_{G0,H}$
M_{α}	$M_{\alpha,1}$	$M_{\alpha,2}$	…	$M_{\alpha,H-1}$	$M_{\alpha,H}$
M_{sf}	$M_{sf,1}$	$M_{sf,2}$	…	$M_{sf,H-1}$	$M_{sf,H}$
M_{cr}	$M_{cr,1}$	$M_{cr,2}$	…	$M_{cr,H-1}$	$M_{cr,H}$

where, $rand_i$ denotes a uniformly selected random number from (0,1).

After generating the mutant vector $u_i^{\mathrm{d}}(t)$, bin crossover is implemented as follows:

$$\begin{cases} v_i^d(t) = u_i^d(t) & (if \quad rand < cr(i) \quad or \quad j = jrand) \\ v_i^d(t) = x_i^d(t) & (otherwise) \end{cases} \tag{19}$$

After all of the trial vector $v_i^d(t)$ have been generated, a selection process determines the survivors for the next generation is introduced. The selection operator is compared to agent x_i with its corresponding trial vector v_i, and keep the vector with high fitness in the population into next generation.

$$x_i(t+1) = \begin{cases} v_i(t) & \text{if} \quad f(v_i(t)) \le f(x_i(t)) \\ x_i(t) & otherwise \end{cases} \tag{20}$$

3.4. History based parameter adaption

As shown in Table 1, SGSADE maintains a historical memory with H entries for four control parameters M_{G0},M_{α},M_{sf},M_{cr}.

In each generation, the generate rules and update rules of $G0_i$ and α_i are given in Eqs. (12) and (13). $\mu\alpha$ and μG0 are randomly generated from historical memory. sf_i and cr_i are generated by applying the equations below:

$$sf_i = randc_i\left(M_{sf,i}, 0.1\right) \tag{21}$$

$$cr_i = randn_i(M_{cr,i}, 0.1) \tag{22}$$

When $sf_i > 1$, sf_i is truncated to1, and when $sf_i \le 0$, Eq. (21) is repeatedly applied to try to generate a valid value. When cr_i outside of [0, 1] is generated, it is replaced by the limit value (0 or 1) closest to the generated value.

In each generation, sf_i and cr_i values that succeed in generating a trial vector which is better than the parent agent is recorded as S_{sf} and S_{cr}, and at the end of the generation, $M_{sf,i}$ and $M_{cr,i}$ are updated as:

$$M_{sf,i} = (1-c) \times M_{sf,i} + c \times mean_L\left(S_{sf}\right) \tag{23}$$

$$M_{cr,i} = (1-c) \times M_{cr,i} + c \times mean_A(S_{cr}) \tag{24}$$

$mean_L(\bullet)$ is computed as:

$$\mathrm{mean}_L\left(S_{sf}\right) = \frac{\sum_{sf\in S_{sf}} sf^2}{\sum_{sf\in S_{sf}} sf} \tag{25}$$

$mean_A(\bullet)$ is computed as:

$$\mathrm{mean}_A(S_{cr}) = \sum_{k=1}^{|S_{cr}|} \omega_k \times S_{cr,k} \tag{26}$$

$$\omega_k = \frac{\Delta f_k}{\sum_{k=1}^{|S_{cr}|} \Delta f_k} \tag{27}$$

where $\Delta f_k = |f(v_k) - f(x_k)|$.

3.5. The Process Of SGSADE

The steps of SGSADE are described in Algorithm 2.

Table 2
Parameter settings.

Algorithms	Parameter settings
GSA	$G0=100, \alpha=20$
LDGSA	$G_i=80, G_f=0$
CROGSA	$G0=100$, $\alpha=20$, $CR = 0.1$
CGSA	$G0=100$, $\alpha=20$, Chaotic map is Sinusoidal, $x(1)=0.7$
CKGSA	$G0=100$, $\alpha=20$, $\mu=4$, $z(1)=0.4$
IGSA	$\alpha=20$, $z(1)=0.152$, $\sigma=0.3$, $p=0.1$, $k=6$
IGSAPSO	$G0=100$, $\alpha=20$, $c_{1i}=2.5$, $c_{1f}=0.5$, $c_{2f}=0.5$, $c_{2i}=2.5$

Table 3
Combinations of parameters.

Factor level	1	2	3	4
α	10	15	20	25
G0	50	75	100	125
sf	0.3	0.5	0.7	0.9
cr	0.3	0.5	0.7	0.9

Table 4
Rank of parameter.

Level	Parameters			
	α	G0	sf	*cr*
1	1.85E+04	3.08E+04	3.98E+04	4.29E+04
2	4.17E+04	3.19E+04	2.64E+04	2.07E+04
3	1.80E+04	2.90E+04	2.69E+04	3.71E+04
4	4.92E+04	3.58E+04	3.44E+04	2.67 E+04
Std.	1.60E+04	2.88E+03	6.43E+03	1.15E+04
Rank	1	4	3	2

Table 5
Orthogonal array.

No.	Parameters				AVE		TAVE	
	α	G0	sf	*cr*	10D	30D	50D	
1	1	1	1	1	1.87E+03	9.22E+03	6.69E+04	2.60E+04
2	1	2	2	2	1.17E+03	8.71E+03	2.95E+04	1.31E+04
3	1	3	3	3	1.10E+03	5.85E+03	3.81E+04	1.50E+04
4	1	4	4	4	1.02E+03	6.07E+03	5.27E+04	1.99E+04
5	2	1	2	3	2.03E+03	6.81E+03	1.13E+05	4.06E+04
6	2	2	1	4	3.27E+03	1.16E+04	1.16E+05	4.36E+04
7	2	3	4	1	1.98E+03	2.25E+04	1.66E+05	6.35E+04
8	2	4	3	2	1.23E+03	8.98E+03	4.72E+04	1.91E+04
9	3	1	3	4	1.11E+03	7.46E+03	4.53E+04	1.80E+04
10	3	2	4	3	1.17E+03	7.50E+03	3.74E+04	1.54E+04
11	3	3	1	2	1.54E+03	1.42E+04	2.01E+04	1.19E+04
12	3	4	2	1	1.31E+03	1.10E+04	6.77E+04	2.67E+04
13	4	1	4	2	1.36E+03	1.85E+04	9.60E+04	3.86E+04
14	4	2	3	1	1.52E+03	1.73E+04	1.47E+05	5.53E+04
15	4	3	2	4	1.51E+03	9.74E+03	6.47E+04	2.53E+04
16	4	4	1	3	1.34E+03	9.84E+03	2.21E+05	7.74E+04

4. The experiment and comparisons

In this paper, SGSADE is evaluated on the CEC 2017 benchmark problem set and compared with the state-of-the-art variant algorithms of the GSA and standard GSA algorithm:

- GSA (Rashedi, et al., 2009)
- LDGSA (Jordehi, 2017)
- CROGSA (Yin, et al., 2017)
- CGSA (Mirjalili & Gandomi, 2017)
- CKGSA (Mittal, et al., 2017)
- IGSA (Ji, et al., 2017)
- IGSAPSO (Xiao, et al., 2016)

According to benchmark properties, they are divided into four class: Unimodal Functions (f1–f3), Simple Multimodal Functions (f4–f10), Hybrid Functions (f11–f20) and Composition Functions (f21–f30). More details description of the benchmark functions can be found in Awad, Ali, Liang, Qu, and Suganthan (2016).

For ensuring the fairness of all algorithms, the experiments were implemented on a PC with 3.4 GHz Intel(R) Core^{TM} i7-6700 CPU, 8GB of RAM and 64-bit OS, all of the algorithms were programmed with MATLAB (2016b).

The parameters of the comparison algorithms are configured as recommended in the corresponding literature. Then, the parameters set of the involved algorithms are listed in Table 2. In our experimental study, all the functions are tested in ten, thirty and fifty dimensions. For all algorithms, the maximum number of iterations for ten dimensions is set as 2000, thirty dimensions is set as 6000 and fifty dimensions is set as 10000. All algorithms are run 51 times independently for each case with the aim to ensure the reliability of experimental results.

4.1. Parameters analysis

The SGSADE has four critical parameters: α (shrinking parameters), $G0$ (gravitational constant), *sf* (scaling factor), *cr* (crossover rate). In this paper, Design of Experiments method (Montgomery, 2006) is applied to determine a set of suitable parameters for the SGSADE and CEC2017 benchmark set is employed to as test set. In order to make the experiment to be statistically significant, different dimensions (D = 10, D = 30 and D = 50, out of F2, the results of F2 are an unstable) are selected to be as test cases, and each function runs 10 times independently for each parameters group. The orthogonal array $L_{16}(4^4)$ is selected according to the number of parameters and the factor levels. The different combinations of parameters are shown in Table 3.

In Table 4, the rank of each parameter is listed. For each parameters group, the average value (AVE) for each dimension and the total AVE (TAVE) are shown in Table 5.

Fig. 1 shows the change tendency of each parameter according to the Table 4. It can be seen that α is the most significant one among the four parameters. The Table 4 shows that the results of SGSADE is the best when the α is 20. The reason is that α has a significant influence on the balance exploitation and exploration (Ji, et al., 2017). The second parameter is the *cr* and the third parameter is *sf*. According to Table 4 and Fig. 1, it can be seen that an appropriate value of *sf* and *cr* will effectively improve the performance of SGSADE due to the population diversity is maintained in the late of search process. The last parameter is $G0$. From Fig. 1, the appropriate value of $G0$ can improve the quality of solution effectively. According to the above analysis, the contents of $M_{(sf,i)}$, $M_{(cr,i)}$ (i=1...H) are all initialized to 0.5, $M_{(G0,i)}$ is initialized to 100, $M_{(\alpha,i)}$ is initialized to 20, $H = 100$.

4.2. Analysis and discussion

The results are shown in Tables 6–8. In this research, the best value in each run will be set to zero if the value is less than 10e-8. In the tables, the mean and standard deviation of 51 times independently runs and the optimal value are shown. The best result for each function is shown in boldface.

For the evolution algorithms, solving CEC2017 benchmark functions is an immense challenge. With the dimensions increasing of functions, the optimization problems will become more and more complicated.

To give some insights into the search performance of SGSADE and the compared algorithms, some representative cases of convergence graphs of the compared algorithms and SGSADE are shown

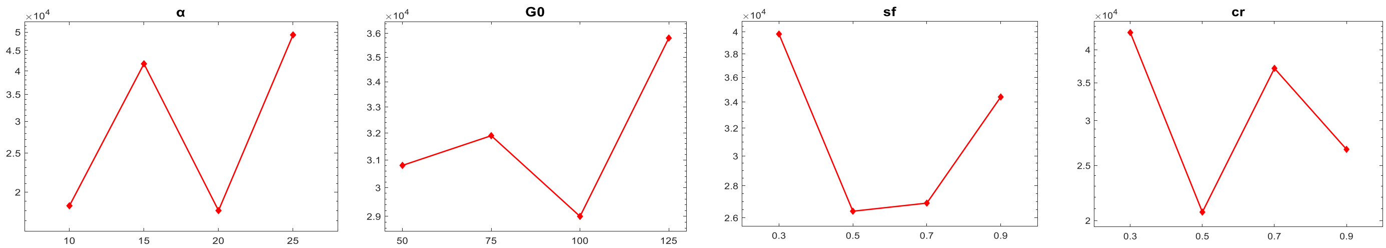

Fig. 1. Change tendency of the parameters.

Table 6
The results for $D = 10$.

Function	Criterion	GSA	LDGSA	CROGSA	CGSA	CKGSA	IGSA	IGSAPSO	SGSADE
1	Mean	2.50E+02	1.06E+03	3.66E+03	2.44E+02	3.83E+02	3.73E+02	2.43E+03	**0.00E+00**
	Std.	2.74E+02	1.31E+03	3.36E+03	3.25E+02	6.32E+02	5.39E+02	2.11E+03	0.00E+00
2	Mean	1.26E+03	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.67E+02	5.92E+02	0.00E+00	**0.00E+00**
	Std.	8.44E+02	0.00E+00	0.00E+00	0.00E+00	7.24E+02	6.48E+02	0.00E+00	0.00E+00
3	Mean	4.87E+03	1.13E-05	**0.00E+00**	4.65E-07	4.25E+03	3.72E+03	4.06E-03	**0.00E+00**
	Std.	1.51E+03	1.08E-05	0.00E+00	2.45E-07	1.28E+03	1.31E+03	2.06E-03	0.00E+00
4	Mean	5.12E+00	2.84E-02	5.73E+00	2.14E-02	4.57E+00	5.08E+00	2.69E+00	**1.24E-02**
	Std.	3.53E-01	7.78E-03	1.22E+01	2.53E-03	1.40E+00	1.81E-01	1.85E+00	1.04E-02
5	Mean	4.39E+01	3.43E+01	2.32E+01	3.33E+01	4.44E+01	**3.73E+00**	2.02E+01	8.00E+00
	Std.	7.33E+00	7.23E+00	1.02E+01	6.88E+00	8.21E+00	1.49E+00	7.11E+00	2.41E+00
6	Mean	8.24E+00	2.71E-01	2.22E-01	1.33E+00	6.30E+00	4.14E-07	4.36E-01	**0.00E+00**
	Std.	8.08E+00	1.00E+00	1.58E+00	2.29E+00	6.40E+00	9.93E-07	1.57E+00	0.00E+00
7	Mean	1.34E+01	1.56E+01	2.57E+01	1.32E+01	**1.29E+01**	1.33E+01	2.08E+01	1.88E+01
	Std.	1.66E+00	2.44E+00	1.02E+01	1.53E+00	1.48E+00	1.43E+00	4.79E+00	2.53E+00
8	Mean	2.04E+01	2.03E+01	2.05E+01	1.77E+01	1.88E+01	**2.89E+00**	1.65E+01	7.08E+00
	Std.	4.77E+00	4.84E+00	9.88E+00	4.37E+00	4.97E+00	1.74E+00	5.84E+00	2.15E+00
9	Mean	**0.00E+00**	1.64E-06	1.24E+01	7.23E-08	**0.00E+00**	0.00E+00	6.73E+00	**0.00E+00**
	Std.	0.00E+00	1.48E-06	3.63E+01	6.29E-08	0.00E+00	0.00E+00	3.38E+01	0.00E+00
10	Mean	1.73E+03	1.43E+03	6.11E+02	1.60E+03	1.65E+03	7.87E+02	7.17E+02	**4.23E+02**
	Std.	2.81E+02	2.87E+02	2.93E+02	2.95E+02	3.07E+02	2.72E+02	2.51E+02	1.45E+02
11	Mean	3.16E+01	2.67E+01	2.75E+01	2.87E+01	3.12E+01	2.84E+01	2.34E+01	**1.07E+00**
	Std.	1.06E+01	1.40E+01	2.25E+01	9.17E+00	8.95E+00	1.09E+01	1.11E+01	8.44E-01
12	Mean	2.65E+05	7.88E+03	1.77E+04	7.22E+03	2.75E+05	1.82E+05	8.57E+03	**3.00E+02**
	Std.	1.95E+05	4.66E+03	1.72E+04	2.37E+03	2.08E+05	2.17E+05	7.19E+03	1.33E+02
13	Mean	1.00E+04	6.50E+03	9.71E+03	7.21E+03	3.16E+04	8.75E+03	8.96E+03	**6.06E+00**
	Std.	2.45E+03	2.71E+03	1.05E+04	2.17E+03	9.73E+04	3.18E+03	6.69E+03	2.72E+00
14	Mean	4.94E+03	3.78E+03	4.18E+03	4.18E+03	4.36E+03	4.25E+03	1.05E+02	**1.06E+00**
	Std.	1.60E+03	1.73E+03	6.39E+03	1.47E+03	1.46E+03	2.23E+03	6.42E+01	6.43E-01
15	Mean	1.48E+04	2.66E+03	4.82E+03	2.25E+03	1.63E+04	1.06E+04	3.16E+02	**3.60E-01**
	Std.	5.56E+03	2.03E+03	7.92E+03	1.32E+03	5.55E+03	5.08E+03	4.11E+02	2.05E-01
16	Mean	5.35E+02	4.40E+02	1.93E+02	4.78E+02	1.35E+03	2.90E+02	1.12E+02	**6.02E-01**
	Std.	9.03E+01	7.96E+01	1.44E+02	9.09E+01	3.62E+03	7.82E+01	1.01E+02	2.34E-01
17	Mean	1.84E+02	1.64E+02	7.63E+01	1.49E+02	1.70E+02	6.12E+01	4.05E+01	**3.28E+00**
	Std.	1.16E+02	9.11E+01	5.37E+01	1.03E+02	1.19E+02	1.85E+01	2.72E+01	2.91E+00
18	Mean	6.79E+03	7.40E+03	1.63E+04	6.65E+03	7.53E+03	6.73E+03	6.60E+03	**4.44E-01**
	Std.	4.29E+03	6.57E+03	1.24E+04	4.94E+03	4.96E+03	4.88E+03	6.31E+03	2.47E-01
19	Mean	1.47E+04	4.95E+03	9.62E+03	4.73E+03	1.32E+04	5.96E+03	3.71E+02	**7.82E-02**
	Std.	4.04E+03	3.23E+03	1.03E+04	2.15E+03	3.56E+03	4.01E+03	4.36E+02	5.99E-02
20	Mean	2.55E+02	1.92E+02	7.41E+01	2.32E+02	1.36E+03	8.85E+01	3.34E+01	**0.00E+00**
	Std.	8.20E+01	7.83E+01	5.97E+01	7.48E+01	3.77E+03	5.96E+01	1.19E+01	0.00E+00
21	Mean	2.49E+02	2.39E+02	2.12E+02	2.38E+02	2.47E+02	1.96E+02	2.01E+02	**1.53E+02**
	Std.	1.70E+01	8.10E+00	3.83E+01	2.77E+01	2.67E+01	2.79E+01	4.81E+01	5.45E+01
22	Mean	1.00E+02	1.00E+02	1.27E+02	1.00E+02	1.00E+02	1.00E+02	**9.81E+01**	1.00E+02
	Std.	8.26E-02	2.41E-01	1.55E+02	7.33E-02	0.00E+00	4.85E-02	1.39E+01	2.26E-01
23	Mean	3.97E+02	3.50E+02	3.28E+02	3.74E+02	4.01E+02	3.08E+02	3.29E+02	**3.06E+02**
	Std.	7.45E+01	1.50E+01	1.15E+01	2.26E+01	3.35E+01	3.91E+00	8.15E+00	2.37E+00
24	Mean	1.94E+02	**1.71E+02**	3.50E+02	2.02E+02	2.27E+02	2.66E+02	3.49E+02	2.90E+02
	Std.	1.37E+02	1.15E+02	5.34E+01	1.26E+02	1.40E+02	1.04E+02	6.43E+01	8.91E+01
25	Mean	4.40E+02	4.38E+02	4.30E+02	4.36E+02	4.38E+02	4.43E+02	4.27E+02	**4.22E+02**
	Std.	1.20E+01	1.49E+01	2.75E+01	1.63E+01	1.42E+01	1.24E+01	2.21E+01	2.28E+01
26	Mean	5.42E+02	2.43E+02	5.83E+02	2.76E+02	5.49E+02	**2.35E+02**	3.74E+02	3.00E+02
	Std.	5.19E+02	5.00E+01	4.26E+02	1.70E+02	5.47E+02	4.83E+01	2.33E+02	0.00E+00
27	Mean	4.95E+02	4.01E+02	4.16E+02	4.49E+02	4.83E+02	3.98E+02	4.04E+02	**3.84E+02**
	Std.	2.97E+01	8.82E+00	3.06E+01	2.63E+01	5.24E+01	3.55E+00	1.59E+01	6.03E+00
28	Mean	5.59E+02	5.29E+02	5.14E+02	5.30E+02	5.47E+02	5.56E+02	5.12E+02	**3.15E+02**
	Std.	6.99E+01	1.15E+02	1.33E+02	9.76E+01	8.69E+01	6.44E+01	1.27E+02	4.68E+01
29	Mean	4.66E+02	3.60E+02	3.67E+02	4.11E+02	4.85E+02	3.07E+02	3.13E+02	**2.58E+02**
	Std.	1.30E+02	1.09E+02	7.36E+01	1.45E+02	1.36E+02	3.81E+01	4.17E+01	8.90E+00
30	Mean	3.87E+05	3.59E+03	3.26E+05	3.44E+03	3.43E+05	2.26E+05	2.01E+05	**4.40E+02**
	Std.	1.05E+05	1.42E+03	4.91E+05	8.20E+02	9.48E+04	7.50E+04	4.05E+05	1.82E+02

Table 7
The results for $D = 30$.

Function	Criterion	GSA	LDGSA	CROGSA	CGSA	CKGSA	IGSA	IGSAPSO	SGSADE
1	Mean	2.09E+03	2.90E+03	6.93E+03	2.47E+03	2.04E+03	2.30E+03	1.95E+04	**0.00E+00**
	Std.	1.86E+03	3.30E+03	6.41E+03	2.17E+03	1.45E+03	2.75E+03	6.88E+03	0.00E+00
2	Mean	1.93E+13	1.10E+00	9.97E+26	**0.00E+00**	5.28E+12	6.93E+10	2.66E+14	**0.00E+00**
	Std.	7.25E+13	7.84E+00	4.96E+27	0.00E+00	2.16E+13	2.48E+11	1.90E+15	0.00E+00
3	Mean	7.40E+04	**2.12E+01**	1.14E+03	9.35E+02	7.25E+04	6.30E+04	7.32E+03	1.33E+02
	Std.	6.89E+03	8.50E+01	5.85E+03	5.52E+02	9.14E+03	9.28E+03	4.98E+03	1.05E+02
4	Mean	8.56E+01	6.36E+01	8.18E+01	1.52E+01	8.45E+01	8.45E+01	8.23E+01	**1.35E+01**
	Std.	2.19E+01	3.97E+00	1.49E+01	2.22E+01	1.75E+01	1.66E+00	2.46E+01	2.63E+01
5	Mean	2.03E+02	1.87E+02	1.23E+02	1.84E+02	2.05E+02	**1.45E+01**	1.47E+02	8.25E+01
	Std.	1.93E+01	1.96E+01	3.13E+01	1.93E+01	2.11E+01	3.62E+00	3.45E+01	1.15E+01
6	Mean	3.94E+01	2.06E+01	1.16E+01	2.89E+01	4.70E+01	1.51E-06	2.21E+01	**0.00E+00**
	Std.	6.19E+00	7.68E+00	7.89E+00	6.30E+00	3.37E+01	3.09E-06	8.22E+00	0.00E+00
7	Mean	4.96E+01	5.33E+01	1.76E+02	**4.48E+01**	5.01E+01	4.63E+01	1.58E+02	1.20E+02
	Std.	5.31E+00	5.64E+00	5.14E+01	3.71E+00	5.67E+00	4.30E+00	3.36E+01	1.43E+01
8	Mean	1.44E+02	1.28E+02	1.21E+02	1.26E+02	1.47E+02	**1.43E+01**	1.16E+02	8.10E+01
	Std.	1.67E+01	1.54E+01	2.96E+01	1.49E+01	1.54E+01	3.86E+00	2.49E+01	1.38E+01
9	Mean	4.40E+02	1.61E-05	3.12E+03	4.52E+01	4.68E+02	1.76E-03	3.10E+03	**0.00E+00**
	Std.	3.21E+02	1.04E-05	1.65E+03	1.15E+02	3.55E+02	1.25E-02	9.70E+02	0.00E+00
10	Mean	3.78E+03	3.49E+03	3.21E+03	3.77E+03	3.67E+03	**1.39E+03**	3.57E+03	4.64E+03
	Std.	5.11E+02	4.97E+02	5.90E+02	4.67E+02	5.85E+02	5.13E+02	7.09E+02	5.85E+02
11	Mean	9.48E+01	9.53E+01	2.17E+02	8.67E+01	9.48E+01	1.12E+02	1.28E+02	**4.97E+01**
	Std.	2.60E+01	2.79E+01	7.35E+01	2.01E+01	2.89E+01	3.40E+01	4.48E+01	2.61E+01
12	Mean	6.06E+05	1.75E+04	2.46E+05	**1.15E+04**	5.09E+05	3.04E+05	1.08E+05	1.46E+04
	Std.	2.64E+05	1.21E+04	4.43E+05	3.51E+03	2.67E+05	3.36E+05	6.72E+04	8.64E+03
13	Mean	2.94E+04	1.29E+04	2.56E+04	1.43E+04	2.96E+04	1.87E+04	1.30E+04	**3.07E+02**
	Std	7.76E+03	6.35E+03	2.34E+04	6.86E+03	8.06E+03	1.91E+04	1.42E+04	2.86E+02
14	Mean	6.55E+04	3.23E+03	7.83E+04	2.56E+03	5.30E+04	2.36E+04	9.47E+03	**5.42E+01**
	Std.	3.62E+04	2.53E+03	6.78E+04	1.18E+03	2.92E+04	2.61E+04	8.39E+03	9.88E+00
15	Mean	5.05E+03	8.98E+02	1.62E+04	9.70E+02	3.60E+03	1.36E+04	5.79E+03	**4.97E+01**
	Std.	1.60E+03	7.93E+02	1.27E+04	7.56E+02	1.47E+03	3.08E+03	8.13E+03	2.70E+01
16	Mean	1.50E+03	1.32E+03	9.55E+02	1.36E+03	1.54E+03	4.84E+02	1.87E+03	**4.65E+02**
	Std.	2.97E+02	2.80E+02	3.12E+02	3.46E+02	3.05E+02	2.22E+02	1.87E+03	1.61E+02
17	Mean	1.10E+03	1.13E+03	5.91E+02	9.70E+02	1.11E+03	3.19E+02	5.36E+02	**7.74E+01**
	Std.	2.55E+02	2.09E+02	2.10E+02	2.11E+02	2.53E+02	1.77E+02	1.98E+02	1.84E+01
18	Mean	1.56E+05	4.04E+04	6.09E+05	4.17E+04	1.45E+05	1.89E+05	1.05E+05	**2.67E+03**
	Std.	7.99E+04	1.12E+04	1.05E+06	7.45E+03	6.67E+04	1.69E+05	7.95E+04	2.18E+03
19	Mean	3.55E+03	3.53E+03	1.75E+04	3.04E+03	3.32E+03	6.46E+03	7.36E+03	**2.52E+01**
	Std.	1.73E+03	1.62E+03	1.79E+04	1.66E+03	1.59E+03	4.24E+03	7.99E+03	6.20E+00
20	Mean	1.05E+03	9.94E+02	5.81E+02	9.54E+02	1.01E+03	3.51E+02	4.72E+02	**1.24E+02**
	Std.	2.32E+02	1.93E+02	2.10E+02	2.21E+02	2.34E+02	1.35E+02	1.53E+02	5.41E+01
21	Mean	4.22E+02	3.73E+02	3.39E+02	3.75E+02	4.11E+02	**2.18E+02**	3.48E+02	2.73E+02
	Std.	2.80E+01	4.50E+01	3.10E+01	4.84E+01	2.87E+01	6.06E+00	3.54E+01	1.31E+01
22	Mean	2.99E+03	1.42E+03	3.03E+03	1.32E+03	2.44E+03	**1.00E+02**	2.47E+03	**1.00E+02**
	Std.	2.36E+03	2.09E+03	1.32E+03	2.02E+03	2.42E+03	0.00E+00	2.15E+03	0.00E+00
23	Mean	1.10E+03	8.82E+02	5.55E+02	8.31E+02	1.05E+03	**3.61E+02**	6.72E+02	4.20E+02
	Std.	1.49E+02	1.91E+02	6.11E+01	1.43E+02	1.59E+02	7.89E+00	9.17E+01	1.86E+01
24	Mean	7.68E+02	6.27E+02	6.61E+02	6.49E+02	7.16E+02	**4.29E+02**	7.17E+02	4.91E+02
	Std.	1.13E+02	4.28E+01	8.54E+01	5.50E+01	7.06E+01	4.45E+00	9.22E+01	1.52E+01
25	Mean	4.68E+02	**3.86E+02**	3.89E+02	3.87E+02	4.10E+02	4.87E+02	3.89E+02	3.88E+02
	Std.	1.94E+02	2.56E+00	9.45E+00	3.52E+00	6.34E+01	3.43E-02	8.41E+00	5.22E+00
26	Mean	1.09E+03	7.12E+02	2.95E+03	1.06E+03	1.76E+03	**3.27E+02**	2.93E+03	1.50E+03
	Std	1.51E+03	1.26E+03	7.66E+02	1.59E+03	1.87E+03	4.51E+01	1.72E+03	3.57E+02
27	Mean	1.12E+03	6.99E+02	5.73E+02	7.57E+02	1.12E+03	5.09E+02	5.60E+02	**5.00E+02**
	Std.	2.78E+02	1.35E+02	4.15E+01	9.73E+01	3.22E+02	6.64E+00	3.01E+01	1.11E+00
28	Mean	3.83E+02	**3.02E+02**	4.12E+02	**3.02E+02**	4.10E+02	5.97E+02	3.76E+02	3.20E+02
	Std.	2.69E+01	1.45E+01	4.85E+01	1.45E+01	1.18E+02	2.11E+01	5.98E+01	4.18E+01
29	Mean	1.48E+03	1.24E+03	1.09E+03	1.28E+03	1.56E+03	8.37E+02	1.02E+03	**6.46E+02**
	Std.	2.03E+02	2.13E+02	2.13E+02	2.45E+02	2.09E+02	1.97E+02	2.13E+02	4.96E+01
30	Mean	3.53E+04	3.32E+03	9.41E+03	4.94E+03	3.34E+04	4.91E+05	5.60E+03	**1.92E+03**
	Std.	6.26E+03	4.37E+02	4.31E+03	7.38E+02	7.95E+03	2.94E+04	2.78E+03	9.70E+02

in Figs. 2–5. Due to a large number of functions in the CEC 2017 problem set, we select one function from each category to illustrate the convergence of the algorithms. The selected functions are F1, F9, F19 and F29. Shown in Fig. 2, SGSADE achieved the fastest convergence speed and the best convergence accuracy on F1 for D = 30 and D = 50. For Unimodal Functions (F1 is a unimodal function, the reason of getting high quality solution with SGSADE is that it incorporates the exploration of GSA and the exploitation of DE. Therefore, SGSADE achieves a trade-off between exploration and exploitation. The convergence graphs of Function F6, F19 and F29 is shown in Figs. 3, 4 and 5, respectively. For other three types of functions (Simple Multimodal Functions, Hybrid Functions and Composition Functions), due to the introduction of self-adaptive mechanism, the diversity of the population is preserved in the early iterations. Therefore, although the convergence speed of SGSADE is not the fastest in the early iterations, it produced the best results of the iteration. In summary, SGSADE achieves better balance exploration and exploitation than the compared algorithms and obtains competitive superiority on the most of the test functions.

Table 8
The results for $D = 50$.

Function	Criterion	GSA	LDGSA	CROGSA	CGSA	CKGSA	IGSA	IGSAPSO	SGSADE
1	Mean	1.20E+03	2.28E+03	1.12E+04	1.22E+03	1.17E+03	1.03E+04	5.22E+04	**4.08E+00**
	Std.	1.35E+03	3.66E+03	1.10E+04	1.86E+03	1.58E+03	8.90E+03	1.07E+04	2.02E+01
2	Mean	8.53E+34	7.79E+07	8.31E+56	**7.06E-01**	7.22E+30	5.53E+21	1.45E+28	1.37E+08
	Std.	5.17E+35	2.70E+08	5.76E+57	4.63E+00	4.71E+31	3.25E+22	1.04E+29	6.83E+08
3	Mean	1.60E+05	1.31E+04	1.90E+04	2.02E+04	1.54E+05	1.36E+05	4.62E+04	**2.16E+03**
	Std.	1.35E+04	4.70E+03	2.05E+04	3.76E+03	1.13E+04	1.83E+04	1.61E+04	9.91E+02
4	Mean	1.05E+02	5.78E+01	9.68E+01	**4.57E+01**	1.13E+02	1.24E+02	1.17E+02	7.29E+01
	Std.	5.13E+01	3.99E+01	5.66E+01	4.38E+01	4.57E+01	4.99E+01	5.73E+01	4.29E+01
5	Mean	3.18E+02	3.02E+02	2.78E+02	2.98E+02	3.27E+02	**2.95E+01**	2.97E+02	1.70E+02
	Std.	2.16E+01	2.68E+01	5.73E+01	2.53E+01	2.52E+01	5.74E+00	3.91E+01	3.86E+01
6	Mean	4.70E+01	2.98E+01	2.63E+01	3.80E+01	4.76E+01	1.89E-06	3.75E+01	**0.00E+00**
	Std.	3.57E+00	5.26E+00	8.72E+00	3.66E+00	4.35E+00	2.32E-06	8.57E+00	0.00E+00
7	Mean	1.22E+02	9.38E+01	5.28E+02	8.72E+01	1.20E+02	**8.14E+01**	4.35E+02	2.53E+02
	Std.	1.90E+01	9.59E+00	1.33E+02	9.26E+00	1.71E+01	6.45E+00	6.93E+01	3.59E+01
8	Mean	3.52E+02	3.39E+02	2.81E+02	3.24E+02	3.53E+02	**3.00E+01**	2.95E+02	1.76E+02
	Std.	2.34E+01	2.29E+01	5.27E+01	2.59E+01	2.84E+01	5.59E+00	4.21E+01	4.96E+01
9	Mean	4.36E+03	8.27E+02	1.61E+04	2.19E+03	4.82E+03	**0.00E+00**	1.03E+04	2.65E-01
	Std.	8.99E+02	4.73E+02	1.25E+04	4.84E+02	9.15E+02	0.00E+00	2.47E+03	6.27E-01
10	Mean	6.54E+03	5.90E+03	5.80E+03	6.21E+03	6.44E+03	**1.52E+03**	6.02E+03	8.60E+03
	Std.	7.05E+02	5.42E+02	8.36E+02	6.67E+02	6.86E+02	4.65E+02	8.18E+02	1.19E+03
11	Mean	1.50E+02	1.24E+02	3.71E+02	1.22E+02	1.37E+02	1.73E+02	1.66E+02	**8.73E+01**
	Std.	2.13E+01	1.44E+01	8.74E+01	1.20E+01	1.74E+01	3.36E+01	4.36E+01	2.44E+01
12	Mean	1.13E+06	9.25E+04	8.15E+06	**7.37E+04**	1.20E+06	1.85E+06	1.16E+06	9.24E+04
	Std.	3.95E+05	3.07E+04	3.02E+07	2.94E+04	4.18E+05	1.08E+06	1.45E+06	8.69E+04
13	Mean	1.94E+04	**1.36E+03**	1.89E+04	5.20E+03	1.60E+04	2.95E+04	6.00E+03	1.88E+03
	Std.	4.83E+03	1.92E+03	1.41E+04	1.67E+03	3.57E+03	1.15E+04	6.45E+03	1.44E+03
14	Mean	2.43E+04	5.29E+03	1.92E+05	5.42E+03	2.29E+04	4.00E+04	5.09E+04	**8.06E+02**
	Std.	1.39E+04	3.02E+03	1.68E+05	2.42E+03	1.35E+04	2.84E+04	8.54E+04	9.03E+02
15	Mean	1.23E+04	1.19E+04	1.21E+04	1.12E+04	1.13E+04	1.46E+04	8.69E+03	**3.48E+02**
	Std.	4.22E+03	4.80E+03	9.02E+03	4.98E+03	6.05E+03	3.46E+03	7.28E+03	2.85E+02
16	Mean	1.99E+03	1.85E+03	1.79E+03	1.92E+03	2.03E+03	1.24E+03	1.89E+03	**1.09E+03**
	Std.	3.83E+02	3.54E+02	4.44E+02	3.40E+02	3.23E+02	2.92E+02	4.38E+02	3.06E+02
17	Mean	1.79E+03	1.80E+03	1.82E+03	1.81E+03	1.85E+03	9.32E+02	1.61E+03	**6.58E+02**
	Std.	4.46E+02	3.39E+02	3.91E+02	2.72E+02	3.30E+02	2.57E+02	3.31E+02	1.69E+02
18	Mean	4.61E+05	3.75E+04	6.85E+05	4.01E+04	3.97E+05	3.27E+05	1.51E+05	**3.42E+04**
	Std.	1.69E+05	1.30E+04	7.28E+05	8.06E+03	1.01E+05	1.95E+05	7.41E+04	2.12E+04
19	Mean	1.51E+04	1.47E+04	1.44E+04	1.41E+04	1.61E+04	1.48E+04	1.27E+04	**9.42E+01**
	Std.	4.06E+03	3.53E+03	1.52E+04	3.43E+03	5.05E+03	7.42E+03	8.83E+03	1.15E+02
20	Mean	1.52E+03	1.44E+03	1.17E+03	1.47E+03	1.54E+03	5.82E+02	1.19E+03	**4.83E+02**
	Std.	3.03E+02	3.42E+02	3.09E+02	3.24E+02	3.48E+02	1.89E+02	3.35E+02	1.40E+02
21	Mean	6.04E+02	5.48E+02	5.03E+02	5.49E+02	5.84E+02	**2.32E+02**	5.55E+02	3.78E+02
	Std.	3.92E+01	3.76E+01	5.54E+01	3.86E+01	3.60E+01	6.47E+00	5.91E+01	4.56E+01
22	Mean	8.60E+03	8.11E+03	6.14E+03	8.29E+03	8.52E+03	**1.00E+02**	7.27E+03	5.99E+03
	Std.	6.97E+02	5.33E+02	8.72E+02	5.95E+02	5.45E+02	0.00E+00	1.34E+03	4.65E+03
23	Mean	1.76E+03	1.55E+03	9.53E+02	1.44E+03	1.74E+03	**4.42E+02**	1.26E+03	5.79E+02
	Std.	1.70E+02	1.71E+02	1.03E+02	1.85E+02	2.06E+02	1.09E+01	1.51E+02	4.54E+01
24	Mean	1.18E+03	1.02E+03	1.06E+03	1.03E+03	1.17E+03	**5.19E+02**	1.09E+03	6.37E+02
	Std.	6.13E+01	4.79E+01	1.05E+02	5.24E+01	6.98E+01	7.90E+00	1.70E+02	3.99E+01
25	Mean	5.70E+02	5.35E+02	**5.28E+02**	5.35E+02	5.58E+02	5.64E+02	5.56E+02	5.58E+02
	Std.	2.51E+01	4.52E+01	3.85E+01	4.15E+01	3.08E+01	1.61E+01	2.93E+01	3.01E+01
26	Mean	1.39E+03	5.67E+02	6.38E+03	5.55E+02	9.52E+02	**3.47E+02**	7.60E+03	2.50E+03
	Std.	2.36E+03	1.33E+03	1.09E+03	1.28E+03	1.94E+03	1.91E+02	1.37E+03	3.92E+02
27	Mean	2.77E+03	2.30E+03	9.10E+02	2.20E+03	2.77E+03	5.30E+02	1.00E+03	**5.00E+02**
	Std.	3.97E+02	2.66E+02	1.34E+02	2.59E+02	3.19E+02	1.58E+01	2.52E+02	9.99E-05
28	Mean	5.07E+02	4.99E+02	5.82E+02	4.94E+02	5.06E+02	6.59E+02	5.05E+02	**4.85E+02**
	Std.	3.07E+01	1.77E+01	3.04E+02	1.64E+01	2.28E+01	5.53E-03	2.17E+01	1.90E+01
29	Mean	2.22E+03	1.89E+03	1.97E+03	1.90E+03	2.09E+03	1.08E+03	1.98E+03	**7.96E+02**
	Std.	2.80E+02	3.80E+02	3.70E+02	3.41E+02	3.64E+02	2.99E+02	3.74E+02	1.92E+02
30	Mean	6.54E+06	7.63E+05	1.50E+06	7.51E+05	5.10E+06	2.49E+06	9.18E+05	**2.20E+04**
	Std.	8.85E+05	4.18E+04	5.19E+05	3.00E+04	9.22E+05	4.47E+05	1.79E+05	1.45E+04

Some Box plots of the compared algorithms and SGSADE for F1, F6, F19 and F29 are shown in Figs. 6–9 so as to illustrate the stability and performance of SGSADE. See from Figs. 6–9, SGSADE is the best algorithm in the compared algorithms.

To testify the performance of the SGSADE, the statistical tests are implemented to illustrate that SGSADE has a significant improvement with the compared algorithms. This paper chooses two nonparametric statistical tests, namely Wilcoxon's sign rank test (Molina, Lozano, & Herrera, 2009) that set the confidence level to be 0.95 and Friedman test.

The *p*-values of Wilcoxon's rank-sum test are calculated by comparing different algorithms. The results are shown in Tables 9–11. In the tables, the null hypothesis H0 is that the p-values which are more than 0.05 can be considered as the strong evidence to accept the null hypothesis H0 that the two algorithms come from distributions with equal means. If the *p*-values are less than 0.05 which can be considered to reject the null hypothesis and accept the right-tailed hypothesis H1 that the two algorithms come from distributions with different means and we can know two algorithms which is better than another based on symbol rank

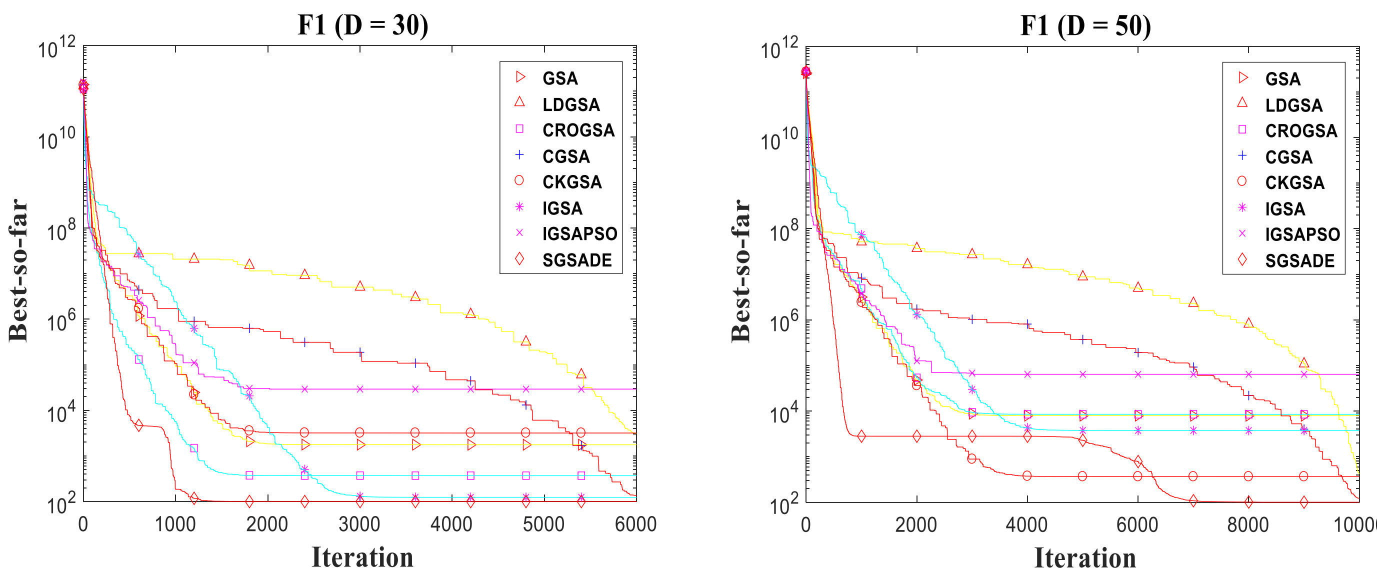

Fig. 2. Convergence performance comparison for best-so-far of F1.

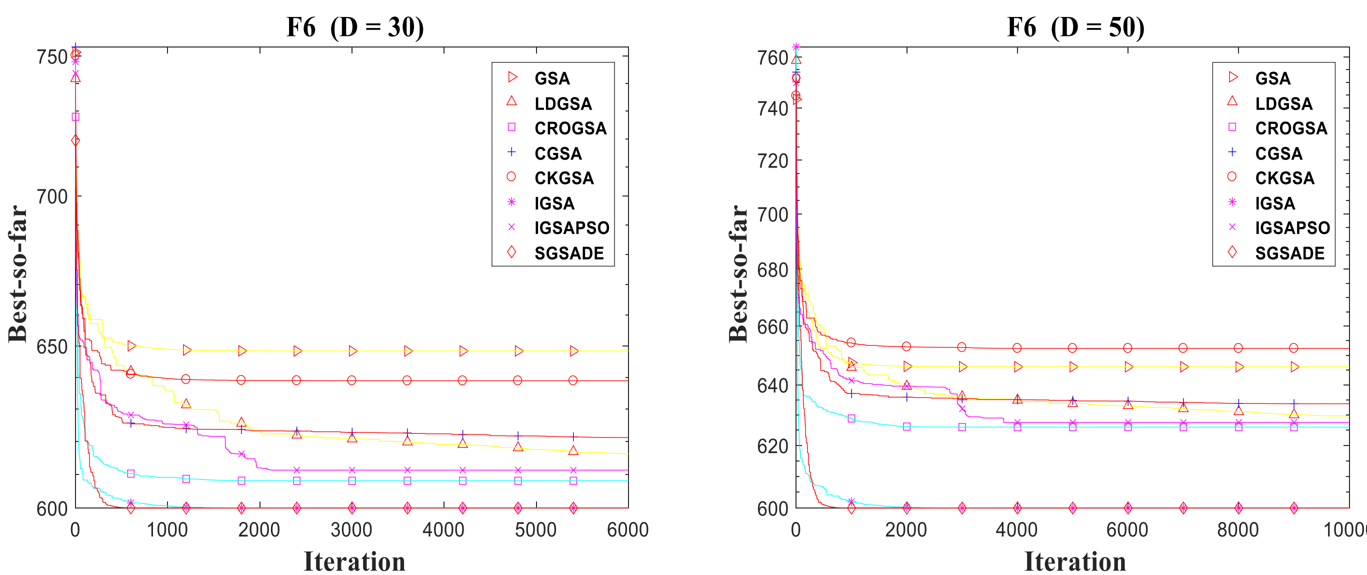

Fig. 3. Convergence performance comparison for best-so-far of F6.

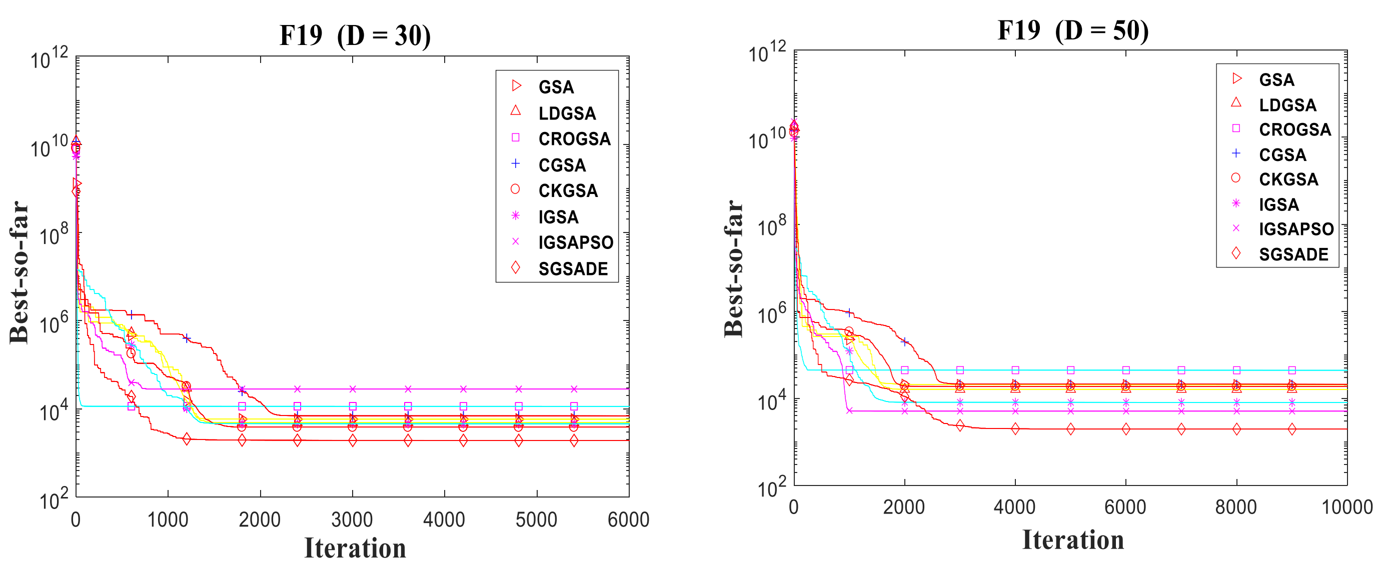

Fig. 4. Convergence performance comparison for best-so-far of F19.

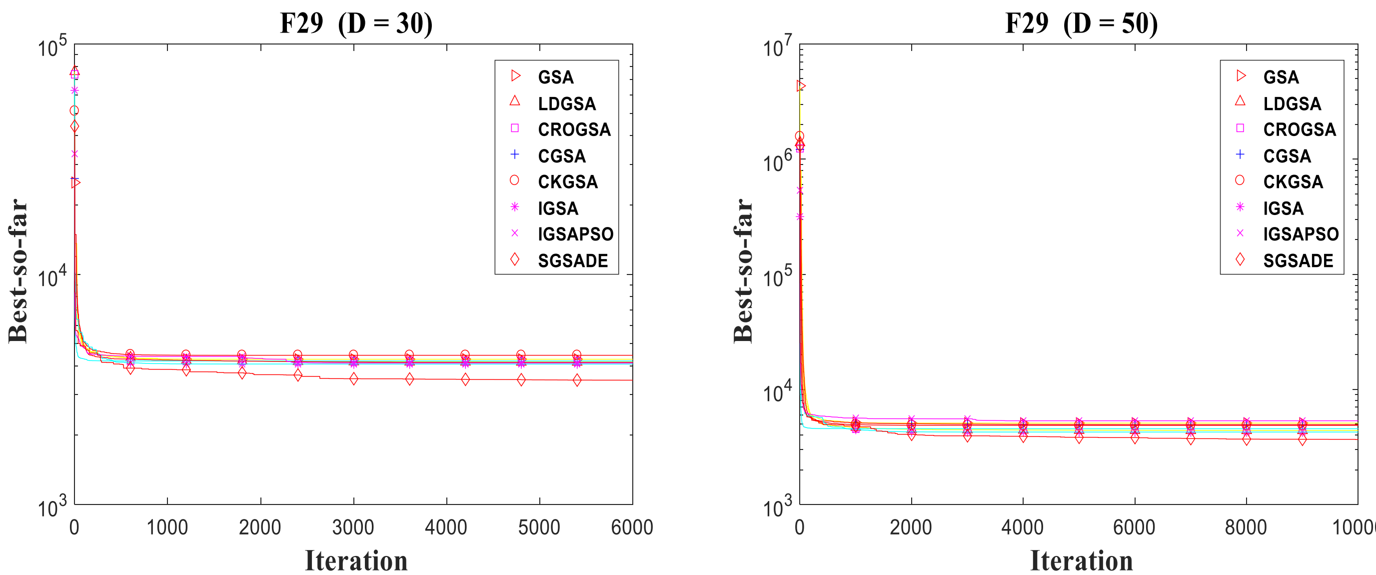

Fig. 5. Convergence performance comparison for best-so-far of F29.

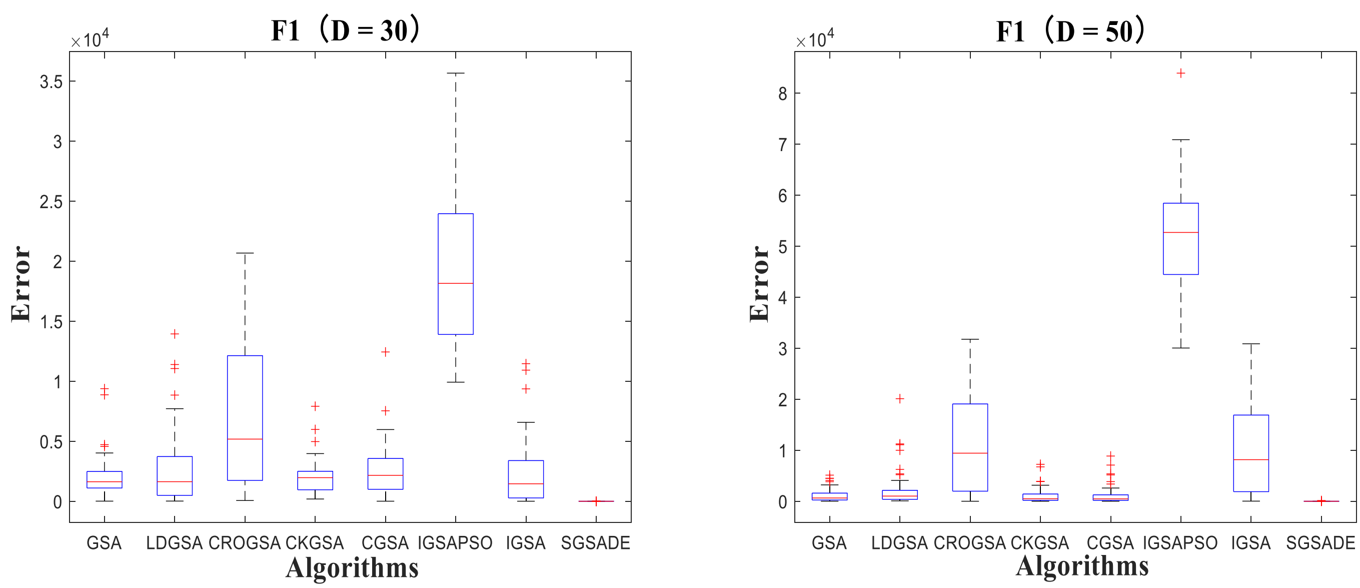

Fig. 6. Box plot for Error of F1.

sum. From Tables 9–11, it can be found that SGSADE outperforms the compared algorithms. The performance of SGSADE is the best among the compared algorithms on the CEC 2017 benchmark functions.

The results of Friedman test are shown in Figs. 10–12. The ranking of the algorithms will be obtained through the Friedman test and the Bonferroni-Dunn's test are shown in Figs. 10–12. The results of pictures and data show that SGSADE are the best algorithm among the compared algorithms.

The following conclusions can be drawn from the above description.

Table 9
p-value of Wilcoxon's rank-sun test for $D = 10$.

SGSADE VS	R+	R-	Z	*p*-value	$\alpha = 0.05$
GSA	394.5	11.5	−4.361	1.3E−04	Yes
LDGSA	374	32	−3.894	9.9E−05	Yes
CROGSA	435	0	−4.703	3E−05	Yes
CGSA	379	27	−4.008	6.1E−05	Yes
CKGSA	396	10	−4.395	1.1E−05	Yes
IGSA	370	36	−3.803	1.43E−04	Yes
IGSAPSO	432	3	−4.638	4E−05	Yes

(a) The performance of SGSADE is the best among the studied algorithms. It provides better mean results than the compared algorithms. Table 6 shows that SGSADE significantly outperforms other algorithms in almost all functions for D = 10. For D = 30 and D = 50, SGSADE yields 18 and 15 the best results in all the algorithms respectively. According to the Tables 9–11, SGSADE has significant differences with the compared algorithms and it is better than the compared algorithms through Wilcoxon

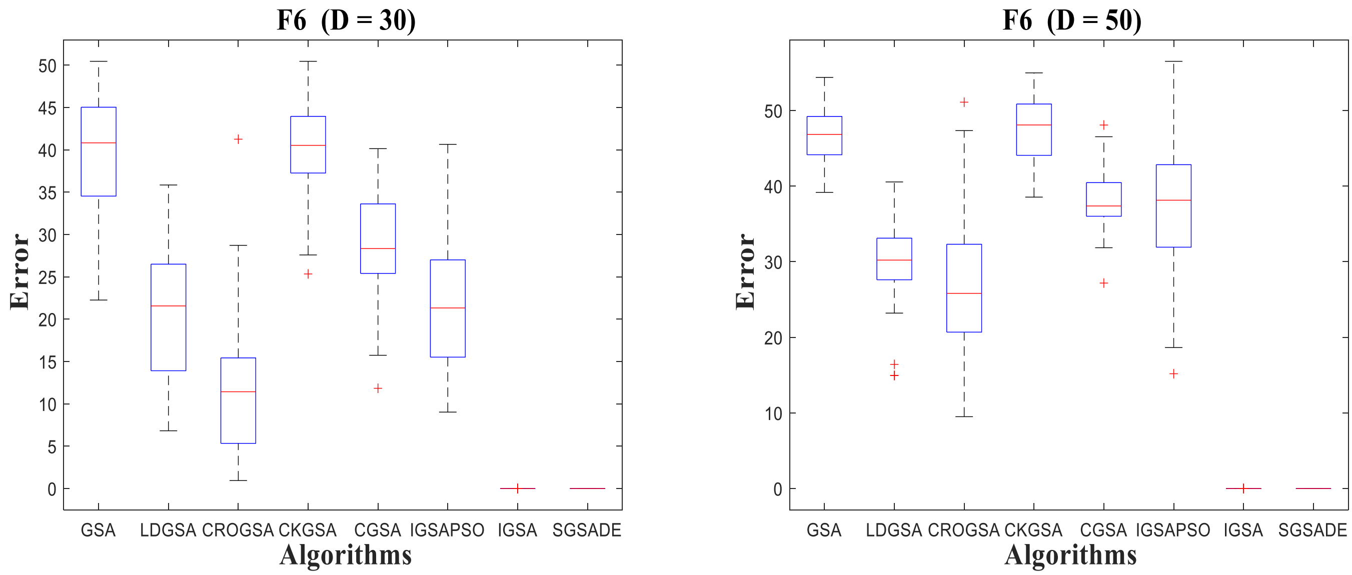

Fig. 7. Box plot for Error of F6.

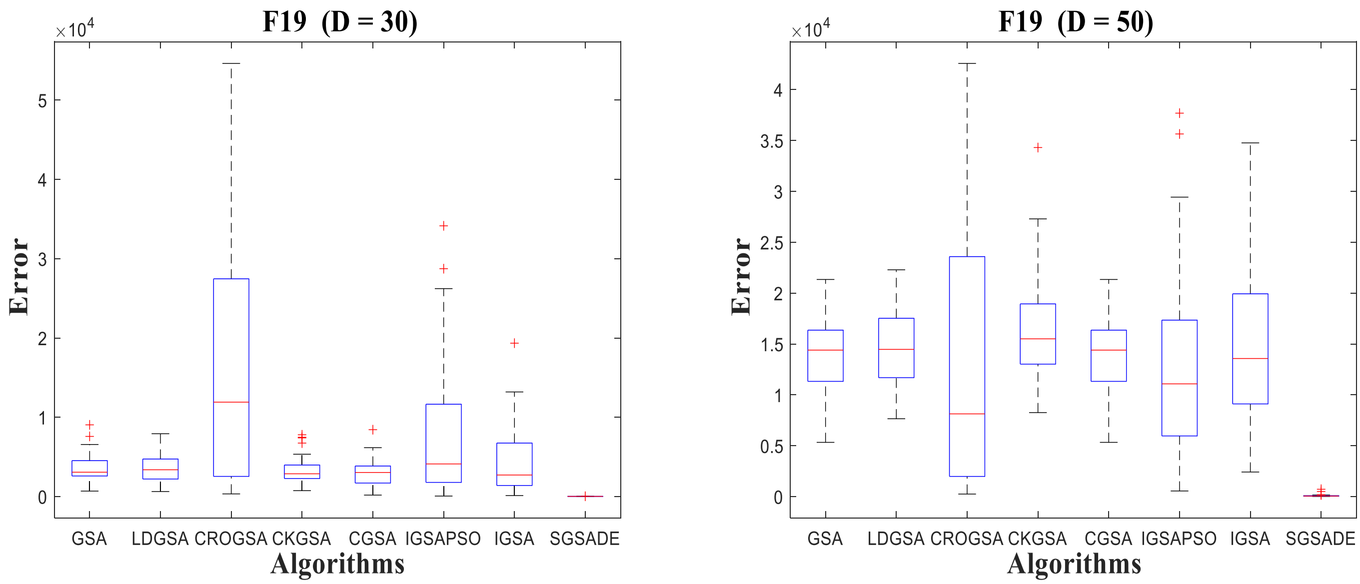

Fig. 8. Box plot for Error of F19.

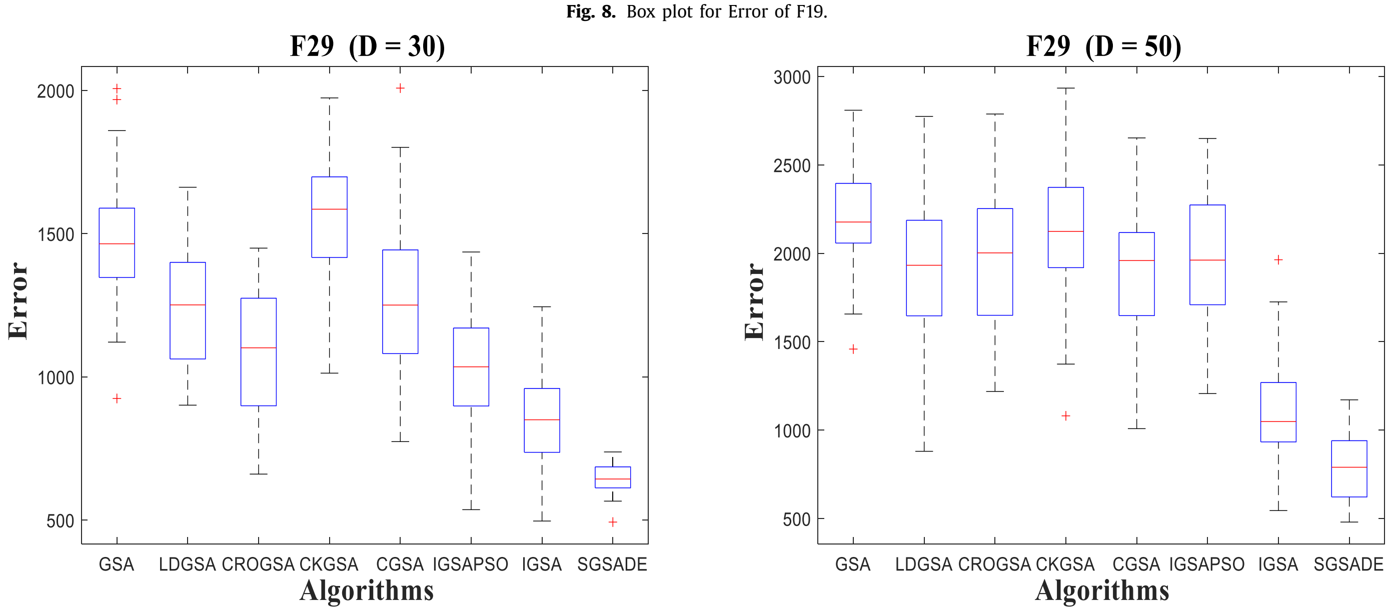

Fig. 9. Box plot for Error of F29.

Algorithms	Mean Rank
GSA	6.67
LDGSA	4.23
CROGSA	5.18
CGSA	4.13
CKGSA	6.45
IGSA	4.03
IGSAPSO	3.57
SGSADE	1.73

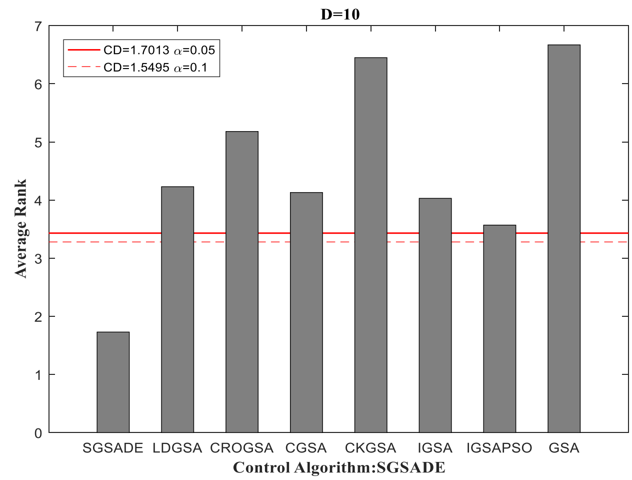

Fig. 10. Rankings for $D = 10$

Table 10
p-value of Wilcoxon's rank-sun test for $D = 30$.

SGSADE VS	R+	R-	Z	p-value	$\alpha = 0.05$
GSA	433	32	−4.124	3.7E−05	Yes
LDGSA	398	67	−3.404	1E−03	Yes
CROGSA	447	18	−4.412	1E−05	Yes
CGSA	364	71	−3.168	2E−03	Yes
CKGSA	443	22	−4.33	1.5E−05	Yes
IGSA	348	87	−2.822	5E−03	Yes
IGSAPSO	449	16	−4.453	8E−05	Yes

Table 11
p-value of Wilcoxon's rank-sun test for $D = 50$.

SGSADE VS	R+	R-	Z	p-value	$\alpha = 0.05$
GSA	428	37	−4.021	5.8E-05	Yes
LDGSA	367	98	−2.766	6E-03	Yes
CROGSA	414	21	−4.249	2.1E−05	Yes
CGSA	355	110	−2.52	1.2E−02	Yes
CKGSA	397	38	−3.881	1.04E−04	Yes
IGSA	342	123	−2.252	2.4E−02	Yes
IGSAPSO	446	19	−4.391	1.1E−05	Yes

signed rank test for D = 10, D = 30 and D = 50. Shown in Tables 12–14 and Figs. 10–12, SGSADE is the best algorithm among the compared algorithms for D = 10, D = 30 and D = 50 on CEC 2017 benchmark functions.

(b) The performance of SGSADE is outstanding on Hybrid Functions among the compared algorithms. Tables 6–8 show that SGSADE stably achieves better results than the compared algorithms out of Function 12 on D = 30 and Function 12, Function 13 on D = 50. For Function 19, the performance of SGSADE is more effective among the compared algorithms for solving the hybrid functions.
(c) The convergence ability of SGSADE is stable. Figs. 2–5 show that SGSADE produces the best results with fast convergence speed with the compared algorithms. The pictures indicate that SGADE can escape from local optimal and obtain better results than the compared algorithms.
(d) The stability of the result for the algorithms is shown in Figs. 6–9. The pictures show that the results of SGSADE are the most stable in the compared algorithms.

5. Conclusions

This paper proposes a new variant algorithm SGSADE based on GSA. In the SGSADE, a new self-adaptive adjusting scheme for two control parameters α and $G0$ is designed to balance exploration and exploitation, and accelerate convergence speed. Meanwhile, the exploitation capacity is significantly strengthened, and

Algorithms	Mean Rank
GSA	6.37
LDGSA	3.75
CROGSA	5.48
CGSA	3.67
CKGSA	6.05
IGSA	3.73
IGSAPSO	5.02
SGSADE	1.93

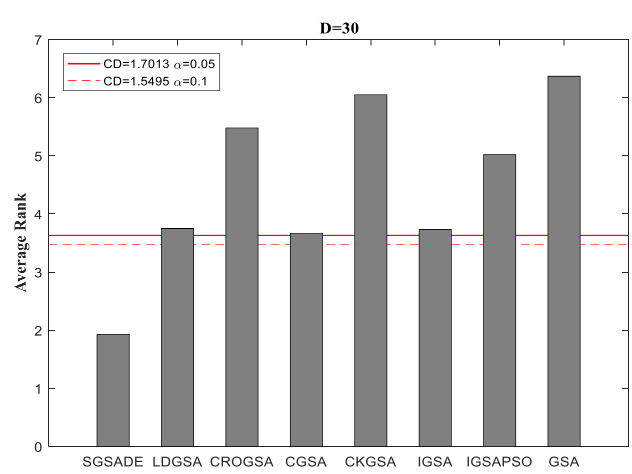

Fig. 11. Rankings for $D = 30$

Algorithms	Mean Rank
GSA	6.5
LDGSA	3.71
CROGSA	4.97
CGSA	3.78
CKGSA	6.21
IGSA	3.79
IGSAPSO	4.93
SGSADE	2.12

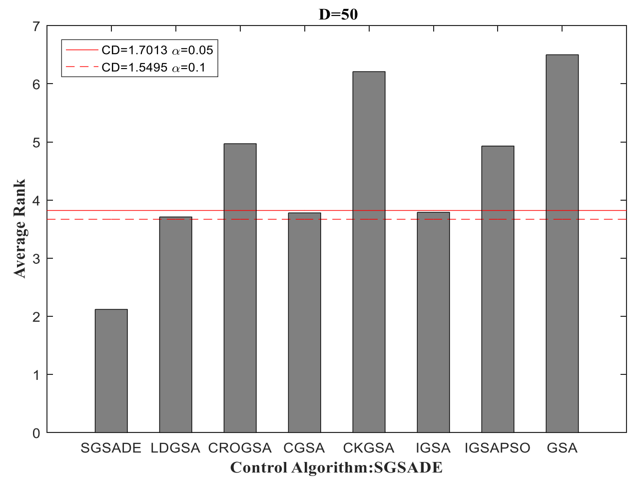

Fig. 12. Rankings for $D = 50$

population diversity is maintained in the late of search process due to the “rand/1” strategy and the “current-to-pbest/1” strategy are employed in the GSA. To further enhance exploitation capacity and solution accuracy, a new disruption based on Levy flight for the global optima is introduced in SGSADE. In addition, CEC 2017 benchmark functions are used for simulation to verify the performance of SGSADE. Experimental results show that the performance of SGSADE is better than the compared algorithms. From the simulation results of statistical test, SGSADE is the best algorithm among the compared algorithms.

In the future research, it is a potential direction of applying SGSADE to solve the combination optimization problem in the real application domain, such as concentric circular antenna arrays design (Biswas, Bose, Das, & Kundu, 2013), sleep scheduling in sensor networks (Kundu, Das, Vasilakos, & Biswas, 2015), unit commitment scheduling problem (Trivedi, Srinivasan, Biswas, & Reindl, 2016), etc. Furthermore, it is another direction that GCO (Biswas, Eita, Das, & Vasilakos, 2014) or some other diversification technique can be integrated with GSA for solving the combination optimization problem in the future.

Acknowledgement

This work was financially supported by the National Natural Science Foundation of China under grant numbers 61663023. It was also supported by the Key Research Programs of Science and Technology Commission Foundation of Gansu Province (2017GS10817), Wenzhou public welfare science and technology project (G20170016), General Program of the Postdoctoral Science Foundation of China (2012M521802), Special Program of the Postdoctoral Science Foundation of China (2013T60889), and the Science Foundation for Distinguished Youth Scholars of Lanzhou University of Technology under grant numbers J201405.

Appendix

This appendix provides sunpplementary matatial for this paper *(A Hybrid Algorithm Based on Self-Adaptive Gravitational Search Algorithm and Differential Evolution).*

Convergence analysis of SGSADE

It is difficult to prove global convergence for SGSADE due to the stochastic nature. In Goryajnov (1996), if the canonical genetic algorithm always maintain the best solution in the population, the algorithm will converge to the global optimum by means of homogeneous finite Markov chain analysis. In this paper, we show the convergence of SGSADE. For convenience, some definitions are produced as follows.

Definition 1. Individual state is determined by the vector x, population state is determined by the individual state of the population, i.e., such that population is **pop** $=\{I_1, I_2, ..., I_N\}$, then population state **S** $=\{x_1, x_2, ..., x_N\}$, where $x_1, x_2, ..., x_N$ is the state of $I_1, I_2, ..., I_N$, respectively. N is the number of the population.

Definition 2. The population state space consists of all possible states of the population, **SP** $=\{S_i = (x_{i1}, x_{i2}, ..., x_{iN}) | i = 1, 2, ..., N\}$.

Definition 3. In SGSADE, the state transition probability of going from x_1 to x_2 in one step is recorded as $x_1 \to x_2$, where x_1 and x_2 are random two states in the individual state space.

Definition 4. In SGSADE, the state transition probability is recorded as $S_i \to S_j$ which going from $S_i = (x_{i1}, x_{i2}, ..., x_{iN})$ to $S_j = (x_{j1}, x_{j2}, ..., x_{jN})$ in one step.

Definition 5. Assume global optimization of optimization problems is g^*, and the optimal population state set is defined as $\Psi = \{S = (x_1, x_2, ..., x_N) | f(x_i) = g^*, \exists x_i, 1 \le i \le N\}$.

Lemma 1. In SGADE, $P(x_1 \to x_2) = P_{x_1 \to u_1} \times P_{(u_1, x_1) \to v_1} \times P_{(v_1, select) \to x_2}$ is the state transition probability of going from the state x_1 to x_2 in one step, where

$$P_{x_1 \to u_1} = 1 \tag{1a}$$

$$P_{(u_1, x_1) \to v_1} = \prod_{i=1}^{D} K(i) \tag{2a}$$

$$(K(i) = \begin{cases} 1 & x_{1i} = u_{1i} \quad or \quad i = q \\ (1 - cr) \times Sel(rand(i) > cr) \\ \quad + cr \times Sel(rand(i) \le cr) & otherwise \end{cases},$$

$Sel(x) = \{\begin{matrix} 1 & x \ is \ true \\ 0 & x \ is \ false \end{matrix}$, $rand(i)$ is random in (0, 1) for ith dimension; D is the number of dimension, q is a integer and $q \in [1, D]$);

$$P_{(v_1, select) \to x_2} = \begin{cases} 1 & f(x_1) \ge f(x_2) \\ 0 & f(x_1) < f(x_2) \end{cases} \tag{3a}$$

Proof..

$$P_{x_1 \to u_1} = (1 - p_1) \times p_2 + (1 - p_1) \times (1 - p_2) + p_1 = 1 \tag{4a}$$

(in this paper, $p_1 = 0.9 - 0.8 \ \times T/\max\_iteration$,$p_2 = 0.1 + 0.8 \ \times T/\max\_iteration$)

According to the cross-operation of DE, the operation of the individual for each dimension as follows:

$$v_1(i) = \begin{cases} u_1(i) & rand(i) \le cr \quad or\ i = q \\ x_1(i) & otherwise \end{cases} \tag{5a}$$

(cr is crossover rate, $rand(i)$ is random in (0,1) for ith dimension; D is the number of dimension, q is an integer and $q \in [1, D]$). The probability of event $rand(i) \le cr$ is $P(rand(i) \le cr) = 1 \times cr = cr$. Due to event $rand(i) \le cr$ and event $rand(i) > cr$ is mutual exclusion event, $P(rand(i) > cr) = 1 - cr$. So,

$$P_{(u_1,x_1)\to v_1} = \prod_{i=1}^{D} K(i) \tag{6a}$$

$$\left(K(i) = \begin{cases} 1 & x_{1i} = u_{1i} \quad or \quad i = q \\ (1-cr) \times Sel(rand(i) > cr) & \\ \quad + cr \times Sel(rand(i) \le cr) & otherwise \end{cases}\right.,$$

$Sel(x) = \{\begin{smallmatrix} 1 & x\ is\ true \\ 0 & x\ is\ false \end{smallmatrix}$, $rand(i)$ is random in (0, 1) for ith dimension; D is the number of dimension, q is an integer and $q \in [1, D]$).

In SGSADE, if the fitness value of x_2 is better than v_1, then the state x_2 is the next state of the x_1, if the fitness value of x_2 is worse than v_1, then the state of x_1 is constant. So,

$$P_{(v_1,select)\to x_2} = \begin{cases} 1 & f(x_1) \ge f(x_2) \\ 0 & f(x_1) < f(x_2) \end{cases} \tag{7a}$$

In summary, according to Eqs. (4a), (6a) and (7a), Lemma 1 is proved.

Lemma 2. In SGSADE, the state transition probability of going from the population state S_i to S_j in one step is

$$P(S_i \to S_j) = \prod_{k=1}^{N} P(x_{ik} \to x_{jk}) \tag{8a}$$

(N is the number of population).

Proof. $S_i \to S_j$ indicates that all individual states of population state S_i are transferred to the corresponding individual states of population state S_j, i.e. $x_{ik} \to x_{jk}$, $k = 1, 2, ..., N$. Due to the state transfer between individuals is independent event, then Eq. (8a) holds. Lemma 2 is proved.

Lemma 3. In SGADE, the population state sequence

$$\{S(t) | t \ge 0\} \tag{9a}$$

is finite homogeneous Markov chains.

Proof. Without loss of generality, we assume that only one variable is included in the optimization problem, its range is $[x_{\min}, x_{\max}]$. If accuracy is ε, then $[x_{\min}, x_{\max}]$ can be expressed by finite discrepancies. A population state $S = (x_1, x_2, ..., x_N)$ consists of the state of N individuals, N is a finite positive integer. So, population state space **SP** is finite.

From Lemma 2, in $\{S(t) | t \ge 0\}$, the state transition probability of going from the state $S(t-1)$ to $S(t)$ is

$$P(S(t-1) \to S(t)) = \prod_{i=1}^{N} P(x_{i,(t-1)} \to x_{i,t}) \tag{10a}$$

From Lemma 1, $P(x_{i,(t-1)} \to x_{i,t})$ is decided by $P_{x_{i,(t-1)} \to u_i}$, $P_{(u_i, x_{i,(t-1)}) \to v_i}$ and $P_{(v_i,select) \to x_{i,t}}$. Due to $P_{x_{i,(t-1)} \to u_i}$, $P_{(u_i, x_{i,(t-1)}) \to v_i}$ and $P_{(v_i,select) \to x_{i,t}}$ are only related to the population state at time $(t-1)$ and independent of time $(t-1)$, and $P(x_{i,(t-1)} \to x_{i,t})$ is in the same way.

From the above analysis, the population state sequence has the characteristics of Markov. The population state space **SP** is discrete and finite, so **SP** is a finite homogeneous Markov chain. Lemma 3 is proved.

Theorem 1. Let $\{S\}$ be a sequence of solutions generated by the SGSADE algorithm, in which $\{S^*\} \in \{S\}$ denotes the optimal solution in the swarm at time t, i.e. $S^* = \arg\min_{1 \le i \le N} f(S_i(t))$.The SGSADE algorithm can converge to global solution with the probability 1. Namely, $\lim_{t\to\infty} P\{f(S^*) = g^*\} = 1$.

Proof. Suppose $p_i(t)$ is the probability of ith individual of the population S at time t, then $p_t = \sum_{S_i(t) \ne S^*} p_i(t)$

According to the Lemma 3,

$$p_{t+1} = \sum_{S(t)} \sum_{S_j(t+1) \ne S^*} p_i(t) p_{ij}(t)$$
$$= \sum_{S_i(t) = S^*} \sum_{S_j(t+1) \ne S^*} p_i(t) p_{ij}(t) + \sum_{S_i(t) \ne S^*} \sum_{S_j(t+1) \ne S^*} p_i(t) p_{ij}(t)$$

According to the properties of Markov chain,

$$\sum_{S_i(t) = S^*} \sum_{S_j(t+1) \ne S^*} p_i(t) p_{ij}(t) + \sum_{S_i(t) \ne S^*} \sum_{S_j(t+1) \ne S^*} p_i(t) p_{ij}(t)$$
$$= \sum_{S_i(t) \ne S^*} p_i(t) = p_t$$

After formula conversion,

$$\sum_{S_i(t) \ne S^*} \sum_{S_j(t+1) \ne S^*} p_i(t) p_{ij}(t) = p_t - \sum_{S_i(t) = S^*} \sum_{S_j(t+1) \ne S^*} p_i(t) p_{ij}(t)$$

Then,

$$0 \le p_{t+1} = \sum_{S_i(t) = S^*} \sum_{S_j(t+1) \ne S^*} p_i(t) p_{ij}(t) + p_t$$
$$- \sum_{S_i(t) = S^*} \sum_{S_j(t+1) \ne S^*} p_i(t) p_{ij}(t)$$
$$< \sum_{S_i(t) = S^*} \sum_{S_j(t+1) \ne S^*} p_i(t) p_{ij}(t) + p_t$$

In the above formula,

$$\sum_{S_i(t) = S^*} \sum_{S_j(t+1) \ne S^*} p_i(t) p_{ij}(t) = 0$$

Therefore,

$$0 \le p_{t+1} < p_t$$

According to the above formula,

$$\lim_{t\to\infty} p_t = 0$$

That is,

$$\lim_{t\to\infty} P\{f(S^*) = g^*\} = 1 - \lim_{t\to\infty} \sum_{S_i(t) \ne S^*} p_i(t) = 1 - \lim_{t\to\infty} p_t \ = 1$$

According to the above analysis, SGSADE can converge to the global solution with the probability 1.

References

Awad, N. H., Ali, M. Z., Liang, J. J., Qu, B. Y., & Suganthan, P. N. (2016). Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization.

Biswas, S., Bose, D., Das, S., & Kundu, S. (2013). Decomposition-based evolutionary multi-objective optimization approach to the design of concentric circular antenna arrays. *Progress in Electromagnetics Research B, 52*, 185–205.

Biswas, S., Eita, M. A., Das, S., & Vasilakos, A. V. (2014). Evaluating the performance of group counseling optimizer on CEC 2014 problems for computational expensive optimization. *Evolutionary Computation*, 1076–1083.

Biswas, S., Kundu, S., Bose, D., Das, S., & Suganthan, P. N. (2013). Synchronizing differential evolution with a modified affinity-based mutation framework. *Differential Evolution*, 61–68.

Biswas, S., Kundu, S., & Das, S. (2015). Inducing niching behavior in differential evolution through local information sharing. *IEEE Transactions on Evolutionary Computation, 19*, 246–263.
Brest, J., Maučec, M. S., & Bošković, B. (2017). Single objective real-parameter optimization: Algorithm jSO. *Evolutionary Computation*, 1311–1318.
Dorigo, M., Birattari, M., & Stutzle, T. (2004). Ant colony optimization. *Computational Intelligence Magazine IEEE, 1*, 28–39.
Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2013). Erratum to: Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Engineering with Computers, 29* 245-245.
Goel, L., Singhal, S., Mishra, S., & Mohanty, S. (2017). Hybridization of gravitational search algorithm and biogeography based optimization and its application on grid scheduling problem. In *International Conference on Contemporary Computing* (pp. 1–6).
Goryajnov, V. V. (1996). Evolutionary families of analytic functions and time-nonhomogeneous Markov branching processes. *Doklady Mathematics, 53*.
Gouthamkumar, N., Sharma, V., & Naresh, R. (2015). Disruption based gravitational search algorithm for short term hydrothermal scheduling. *Expert Systems with Applications, 42*, 7000–7011.
Han, X., Quan, L., Xiong, X., Almeter, M., Xiang, J., & Lan, Y. (2017). A novel data clustering algorithm based on modified gravitational search algorithm. *Engineering Applications of Artificial Intelligence, 61*, 1–7.
Haupt, R. L., & Haupt, S. E. (2004). *Practical Genetic Algorithms* (2nd Edition).
Ismail, A. M., Mohamad, M. S., Majid, H. A., Abas, K. H., Deris, S., Zaki, N., Szm, H., Ibrahim, Z., Shamsir, M. S., & Remli, M. A. (2017). An improved hybrid of particle swarm optimization and the gravitational search algorithm to produce a kinetic parameter estimation of aspartate biochemical pathways. *Biosystems, 162*.
Ji, J., Gao, S., Wang, S., Tang, Y., Yu, H., & Todo, Y. (2017). Self-adaptive gravitational search algorithm with a modified chaotic local search. *IEEE Access, PP* 1-1.
Jiang, S., Wang, Y., & Ji, Z. (2014). Convergence analysis and performance of an improved gravitational search algorithm. *Applied Soft Computing, 24*, 363–384.
Jordehi, A. R. (2017). Gravitational search algorithm with linearly decreasing gravitational constant for parameter estimation of photovoltaic cells. *Evolutionary Computation*, 37–42.
Kang, K., Bae, C., Yeung, H. W. F., & Chung, Y. Y. (2018). A hybrid gravitational search algorithm with swarm intelligence and deep convolutional feature for object tracking optimization. *Applied Soft Computing, 66*.
Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *IEEE international conference on neural networks, 1995. proceedings: 1944* (pp. 1942–1948).
Khatibinia, M., & Khosravi, S. (2014). A hybrid approach based on an improved gravitational search algorithm and orthogonal crossover for optimal shape design of concrete gravity dams. *Applied Soft Computing, 16*, 223–233.
Kundu, S., Biswas, S., Das, S., & Suganthan, P. N. (2013). Crowding-based local differential evolution with speciation-based memory archive for dynamic multimodal optimization. In *Proceedings of the 15th annual conference on genetic and evolutionary computation* (pp. 33–40).
Kundu, S., Das, S., Vasilakos, A. V., & Biswas, S. (2015). A modified differential evolution-based combined routing and sleep scheduling scheme for lifetime maximization of wireless sensor networks. *Soft Computing, 19*, 637–659.
Li, L. L., Lin, G. Q., Tseng, M. L., Tan, K., & Ming, K. L. (2018). A maximum power point tracking method for PV system with improved gravitational search algorithm. *Applied Soft Computing, 65*, 333–348.
Mirjalili, S., & Gandomi, A. H. (2017). Chaotic gravitational constants for the gravitational search algorithm. *Applied Soft Computing, 53*, 407–419.
Mittal, H., Pal, R., Kulhari, A., & Saraswat, M. (2017). Chaotic Kbest gravitational search algorithm (CKGSA). In *International conference on contemporary computing* (pp. 1–6).
Molina, D., Lozano, M., & Herrera, F. (2009). A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization. *Journal of Heuristics, 15*, 617–644.
Montgomery, D. C. (2006). Design and analysis of experiments. *Technometrics, 48* 158-158.
Naik, M. K., & Panda, R. (2015). A new hybrid CS-GSA algorithm for function optimization. *International conference on electrical, electronics, signals, communication and optimization*.
PANT, M., THANGARAJ, R., & ABRAHAM, A. (2011). DE-PSO: A new hybrid meta-heuristic for solving global optimization problems. *New Mathematics & Natural Computation, 7*, 363–381.
Pelusi, D., Mascella, R., Tallini, L., Nayak, J., Naik, B., & Abraham, A. (2018). Neural network and fuzzy system for the tuning of gravitational search algorithm parameters. *Expert Systems with Applications, 102*.
Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). *GSA: A gravitational search algorithm*. Elsevier Science Inc.
Soleimanpour-Moghadam, M., Nezamabadi-Pour, H., & Farsangi, M. M. (2014). A quantum inspired gravitational search algorithm for numerical function optimization. *Information Sciences, 267*, 83–100.
Storn, R., & Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*, 341–359.
Sun, G., Ma, P., Ren, J., Zhang, A., & Jia, X. (2018). A stability constrained adaptive alpha for gravitational search algorithm. *Knowledge-Based Systems, 139*.
Sun, G., Zhang, A., Wang, Z., Yao, Y., Ma, J., & Couples, G. D. (2016). Locally informed gravitational search algorithm. *Knowledge-Based Systems, 104*, 134–144.
Tanabe, R., & Fukunaga, A. (2013). Success-history based parameter adaptation for differential evolution. *Evolutionary Computation*, 71–78.
Trivedi, A., Srinivasan, D., Biswas, S., & Reindl, T. (2016). A genetic algorithm – Differential evolution based hybrid framework: Case study on unit commitment scheduling problem. *Information Sciences, 354*, 275–300.
Xiao, J., Niu, Y., Chen, P., Leung, S. C. H., & Xing, F. (2016). An improved gravitational search algorithm for green partner selection in virtual enterprises. *Neurocomputing, 217*, 103–109.
Yang, X. S. (2018). Mathematical analysis of nature-inspired algorithms.
Yin, B., Guo, Z., Liang, Z., & Yue, X. (2017). Improved gravitational search algorithm with crossover. *Computers & Electrical Engineering*.
Zhang, A., Sun, G., Ren, J., Li, X., Wang, Z., & Jia, X. (2016). A dynamic neighborhood learning-based gravitational search algorithm. *IEEE Transactions on Cybernetics*, 1–12.
Zhang, X., Zou, D., & Shen, X. (2017). A simplified and efficient gravitational search algorithm for unconstrained optimization problems. 11–17.
Zhao, F., Liu, H., Zhang, Y., Ma, W., & Zhang, C. (2017). A discrete water wave optimization algorithm for no-wait flow shop scheduling problem. *Expert Systems with Applications, 91*.
Zhao, F., Liu, Y., Shao, Z., Jiang, X., Zhang, C., & Wang, J. (2016). A chaotic local search based bacterial foraging algorithm and its application to a permutation flow-shop scheduling problem. *International Journal of Computer Integrated Manufacturing, 29*, 962–981.
Zhao, F., Liu, Y., Zhang, C., & Wang, J. (2015). A self-adaptive harmony PSO search algorithm and its performance analysis. *Expert Systems with Applications, 42*, 7436–7455.
Zhao, F., Liu, Y., Zhang, Y., Ma, W., & Zhang, C. (2017). A hybrid harmony search algorithm with efficient job sequence scheme and variable neighborhood search for the permutation flow shop scheduling problems. *Engineering Applications of Artificial Intelligence, 65*, 178–199.
Zhao, F., Shao, Z., Wang, J., & Zhang, C. (2015). A hybrid optimization algorithm based on chaotic differential evolution and estimation of distribution. *Computational and Applied Mathematics, 36*, 433–458.
Zhao, F., Shao, Z., Wang, J., & Zhang, C. (2017). A hybrid differential evolution and estimation of distribution algorithm based on neighbourhood search for job shop scheduling problems. *International Journal of Production Research, 54*, 1–22.
Zhao, F., Zhang, J., Zhang, C., & Wang, J. (2015). An improved shuffled complex evolution algorithm with sequence mapping mechanism for job shop scheduling problems. *Expert Systems with Applications, 42*, 3953–3966.
Zheng, Y. J. (2015). Water wave optimization: A new nature-inspired metaheuristic. *Computers & Operations Research, 55*, 1–11.