网络优先数字出版时间:2017-06-01　　DOI:10.13656/j.cnki.gxkx.20170601.001
网络优先数字出版地址:http://kns.cnki.net/kcms/detail/45.1206.G3.20170601.1109.002.html

Differential Mutation Operator－Based Social Spider Optimization Algorithm*

具有差分进化算子的社会蜘蛛群优化算法*

ZHAO Ruxin,LUO Qifang,ZHOU Yongquan**
赵汝鑫,罗淇芳,周永权

(1. College of Information Science and Engineering,Guangxi University for Nationalities,Nanning,Guangxi,530006,China;2. Guangxi Higher School Key Laboratories of Complex Systems and Intelligent Computing,Nanning,Guangxi,530006,China)
(1.广西民族大学信息科学与工程学院,广西南宁　530006;2.广西高校复杂系统与智能计算重点实验室,广西南宁　530006)

Abstract:【Objective】A social-spider optimization algorithm (SSO) is a novel meta-heuristic optimization algorithm,it has been widely concerned by scholars in this field since it was put forward,and it had been successfully applied in many fields,but the algorithm is still in the early stages of the study,the convergence speed and computational accuracy of the algorithm need to be improved.**【Methods】**In order to enhance the convergence speed and computational accuracy of the algorithm,in this paper,a social-spider optimization algorithm with differential mutation operator (SSO-DM) had been proposed,and was applied to the function optimization problem. The improvement involved differential mutation operator. A social-spider optimization algorithm with differential mutation operator (SSO-DM) was validated by five benchmark functions. **【Results】**Differential mutation operator enhanced the convergence speed and computational accuracy of the algorithm. **【Conclusion】**The results showed that the proposed algorithm was able to obtain accurate solution,and it also had a fast convergence speed and a high degree of stability.

Key words:social-spider optimization algorithm,differential mutation operator,meta-heuristic optimization algorithm,functions optimization

摘要:【目的】社会蜘蛛群优化算法(SSO)是一种新颖的元启发式优化算法,自从它被提出之后就受到该领域学者的广泛关注,并且也被成功应用到许多领域。但是由于社会蜘蛛群优化算法还处在算法的研究初期,该算法的收敛速度与收敛精度还需要进一步提高。**【方法】**将差分进化算子引入到社会蜘蛛群优化算法(SSO-DM)中,并将改进的算法应用于函数优化问题中,通过5个标准测试函数来验证基于差分进化算子的社会蜘蛛群优化算法(SSO-DM)的优化性能。**【结果】**差分进化算子增强了社会蜘蛛群优化算法的收敛速度与收敛精度。**【结论】**本研究中所提出的算法能够获得精确解,并且它也具有较快的收敛速度和较高的算法稳定性。

关键词:社会蜘蛛群优化算法　差分进化算子　元启发式优化算法　函数优化

中图分类号:TP186　　**文献标识码:**A　　**文章编号:**1005-9164(2017)03-0247-11

收稿日期:2017-04-03

修回日期:2017-05-25

作者简介:赵汝鑫(1990－),男,硕士研究生,主要从事智能优化及应用研究。

*国家自然科学基金项目(61463007,61563008)和广西自然科学基金项目(2016GXNSFAA380264)资助。

**通信作者:周永权(1962－),男,教授,博士,博士生导师,主要从事计算智能、智能优化、神经网络及应用研究,E-mail:yongquanzhou@126.com。

0 Introduction

Swarm intelligence optimization algorithm originates from the simulation of various types of biological behavior in nature and has characteristics of simple operation, good optimization performance and strong robustness. Inspired by this idea, there are many bio-inspired swarm intelligent optimization algorithms are proposed, such as, ant colony optimization (ACO)[1], differential evolution (DE)[2], particle swarm optimization (PSO)[3], firefly algorithm (FA)[4], glowworm swarm optimization (GSO)[5], monkey search (MS)[6], harmony search (HS)[7], cuckoo search (CS)[8], bat algorithm (BA)[9], et al. Swarm intelligence optimization algorithm can solve problems which traditional methods cannot handle effectively and it has shown excellent performance in many respects, and its application scope has been greatly expanded.

The social-spider optimization algorithm is proposed by Erik Cuevas in 2013[10], it is a novel metaheuristic optimization algorithm by simulating social-spider behavior. Social-spiders are a representative example of social insects[11]. A social-spider is a spider species whose members maintain a set of complex cooperative behaviors[12]. Whereas most spiders are solitary and even aggressive toward other members of their own species, social-spiders show a tendency to live in groups, forming long-lasting aggregations often referred to as colonies[13]. In a social-spider colony, each member, depending on its gender, executes a variety of tasks such as predation, mating, web design, and social interaction[13-14]. The web is an important part of the colony because it is not only used as a common environment for all members, but also as a communication channel[15] among them. Therefore, important information (such as trapped prays or mating possibilities) is transmitted by small vibrations through the web. Such information, considered as a local knowledge, is employed by each member to conduct its own cooperative behavior, influencing simultaneously the social regulation of the colony[16]. The SSO algorithm is based on the simulation of the cooperative behavior of social-spiders. In SSO algorithm, individuals emulate a group of spiders which interact to each other based on the biological laws of the cooperative colony. The algorithm is divided into two different search agents (spiders): Males and females. Depending on gender, each individual is conducted by a set of different evolutionary operators which mimic different cooperative behaviors which are typical in a colony. Different to most of existent swarm algorithms, most of swarm algorithms model individuals as unisex entities that perform virtually the same behavior. Under such circumstances, algorithms waste the possibility of adding new and selective operators as a result of considering individuals with different characteristics such as sex, task-responsibility, etc. These operators could incorporate computational mechanisms to improve several important algorithm characteristics including population diversity and searching capacities.

In this paper, a social-spider optimization algorithm with differential mutation operator (SSO-DM) has been applied to functions optimization. The improvement involves differential mutation operator. Differential mutation operator enhances the convergence speed and computational accuracy of the algorithm. A social-spider optimization algorithm with differential mutation operator is validated by 5 benchmark functions. The results show that the proposed algorithm is able to obtained accurate solution, and it also has a fast convergence speed and a high degree of stability.

1 The SSO algorithm

The SSO assumes that entire search space is a communal web, where all the social-spiders interact to each other. In this approach, each solution within the search space represents a spider position in the communal web. Every spider receives a weight according to the fitness value of the solution that is symbolized by the social-spider. The algorithm is divided into two different search agents (spiders): Males and females. Depending on gender, each individual is conducted by a set of different evolutionary operators which mimic different cooperative behaviors that are commonly assumed within the colony.

An interesting characteristic of social-spiders is the highly female-biased populations. In order to emulate this fact, the algorithm starts by defining the number of female and male spiders that will be characterized as individuals in the search space. The number of females (N_f) is randomly selected within the range of 65%—90% of the entire population (N). So, N_f is calculated by the following equation:

$$N_f = floor[(0.9 - rand * 0.25) * N], \tag{1}$$

Where $rand$ is a random number between [0, 1], whereas $floor(.)$ maps a real number to an integer number. The number of male spiders (N_m) is computed as the complement between N and N_f . It is calculated as follows:

$$N_m = N - N_f. \tag{2}$$

The group F stands for female population ($F = \{f_1, f_2, \cdots, f_{N_f}\}$), the group M stands for male population ($M = \{m_1, m_2, \cdots, m_{N_m}\}$) and the group S stands for entire population, such that $S = F \cup M$.

1.1 Fitness assignation

In the biological metaphor, the spider size is the characteristic that evaluates the individual capacity to perform better over its assigned tasks. In this approach, every individual (spider) receives a weight (w_i) which represents the solution quality that corresponds to the spider (i) of the population (S). In order to calculate the weight of every spider the next equation is used:

$$w_i = \frac{J(s_i) - worst_s}{best_s - worst_s}, \tag{3}$$

Where $J(s_i)$ is the objective function value of the social-spider individual (s_i). The values of $worst_s$ and $best_s$ are defined as:

$$best_s = \max_{k \in \{1,2,\cdots,N\}} (J(s_k)) \text{ and } worst_s = \min_{k \in \{1,2,\cdots,N\}} (J(s_k)). \tag{4}$$

1.2 Modeling of the vibrations through the communal web

The communal web is used as a mechanism to transmit information among the colony members. This information is encoded as small vibrations that are critical for the collective coordination of all individuals in the population. The vibrations depend on the weight and distance of the spider which has generated them. Since the distance is relative to the individual that provokes the vibrations and the member who detects them, members located near to the individual that provokes the vibrations, perceive stronger vibrations in comparison with members located in distant positions. In order to reproduce this process, the vibrations perceived by the individual i as a result of the information transmitted by the member j are modeled according to the following equation:

$$Vib_{i,j} = w_j * e^{-d_{i,j}^2}, \tag{5}$$

Where $d_{i,j}$ is the Euclidian distance between the spiders i and j, such that $d_{i,j} = \| s_i - s_j \|$. Although it is virtually possible to compute perceived vibrations by considering any pair of individuals, three special relationships are considered within the SSO approach:

①Vibrations ($Vibc_i$) are perceived by the individual $i(s_i)$ as a result of the information transmitted by the member $c(s_c)$ who is an individual that has two characteristics: It is the nearest member to i and possesses a higher weight in comparison to i ($w_c > w_i$).

$$Vibc_i = w_c * e^{-d_{i,c}^2}. \tag{6}$$

②Vibrations ($Vibb_i$) perceived by the individual i as a result of the information transmitted by the member b (s_b), with b being the individual holding the best weight (best fitness value) of the entire population S , such that $w_b = \max_{k \in \{1,2,\cdots,N\}} (w_k)$.

$$Vibb_i = w_b * e^{-d_{i,b}^2}. \tag{7}$$

③Vibrations ($Vibf_i$) perceived by the individual $i(s_i)$ as a result of the information transmitted by the member $f(s_f)$, with f being the nearest female individual to i.

$$Vibf_i = w_f * e^{-d_{i,f}^2}. \tag{8}$$

1.3 Initializing the population

The SSO is an iterative process whose first step is to randomly initialize the entire population (female and male). The algorithm begins by initializing the set S of N spider positions. Each spider position, f_i or m_i, is an n-dimensional vector containing the parameter values to be optimized. Such values are randomly and uniformly distributed between the pre-specified lower initial parameter bound (p_j^{low}) and the upper initial parameter bound (p_j^{high}), just as it is described by the following expressions:

$$f^{0}_{i,j} = p^{low}_{j} + \underset{k=1,2,\cdots,N_m;j=1,2,\cdots,n}{rand(0,1) * (p^{high}_{j} - p^{low}_{j})}, m^{0}_{k,j} = p^{low}_{j} + \underset{k=1,2,\cdots,N_m;j=1,2,\cdots,n}{rand(0,1) * (p^{high}_{j} - p^{low}_{j})}. \quad (9)$$

Where i, j and k are the parameter and individual indexes respectively whereas zero signals the initial population. The function $rand(0,1)$ generates a random number between 0 and 1.

1.4 Cooperative operators

1.4.1 Female cooperative operator

In order to emulate the cooperative behavior of the female spider, a new operator is defined. The operator considers the position change of the female spider i at each iteration process. Such position change, which can be of attraction or repulsion, is computed as a combination of three different elements. The first one involves the change in regard to the nearest member to i that holds a higher weight and produces the vibration ($Vibc_i$). The second one considers the change regarding the best individual of the entire population (S) who produces the vibration ($Vibb_i$). Finally, the third one incorporates a random movement.

Since the final movement of attraction or repulsion depends on several random phenomena, the selection is modeled as a stochastic decision. For this operation, a uniform random number (r_m) is generated within the range [0,1]. If r_m is smaller than a threshold (PF), an attraction movement is generated; otherwise, a repulsion movement is produced. Therefore, such operator can be modeled as follows:

$$f^{k+1}_{i} = \begin{cases} f^{k}_{i} + \alpha * Vibc_i * (s_c - f^{k}_{i}) + \beta * Vibb_i * (s_b - f^{k}_{i}) + \\ \quad \delta * (rand - \frac{1}{2}); r_m < PF \\ f^{k}_{i} - \alpha * Vibc_i * (s_c - f^{k}_{i}) - \beta * Vibb_i * (s_b - \\ \quad f^{k}_{i}) + \delta * (rand - \frac{1}{2}); r_m > PF \end{cases} . \quad (10)$$

Where α, β, δ and $rand$ are random numbers between [0, 1], whereas k represents the iteration number. The individual s_c and s_s represent the nearest member to i that holds a higher weight and the entire population (S), respectively.

Under this operation, each particle presents a movement which combines the past position that holds the attraction or repulsion vector over the local best element (s_c) and the global best individual (s_b) seen so-far. This particular type of interaction avoids the quick concentration of particles at only one point and encourages each particle to search around the local candidate region within its neighborhood (s_c), rather than interacting to a particle (s_b) in a distant region of the domain. The use of this scheme has two advantages. First, it prevents the particles from moving towards the global best position, making the algorithm less susceptible to premature convergence. Second, it encourages particles to explore their own neighborhood thoroughly before converging towards the global best position.

1.4.2 Male cooperative operator

According to the biological behavior of the social-spider, male population is divided into two classes: Dominant and non-dominant male spiders. Dominant male spiders have better fitness characteristics (usually regarding the size) in comparison to non-dominant. Dominant males are attracted to the closest female spider in the communal web. In contrast, non-dominant male spiders tend to concentrate on the center of the male population as a strategy to take advantage of resources that are wasted by dominant males.

For emulating such cooperative behavior, the male members are divided into two different groups (dominant members D and non-dominant members ND) according to their position with regard to the median member. Male members, with a weight value above the median value within the male population, are considered the dominant individuals (D). On the other hand, those under the median value are labeled as non-dominant males (ND). In order to implement such computation, the male population ($M = \{m_1, m_2, \cdots, m_{N_m}\}$) is arranged according to their weight value in decreasing order. Thus, the individual whose weight (w_{N_f+m}) is located in the middle is considered the median male member. Since indexes of the male population in regard to the entire population are increased by the number of female members (N_f), the median weight is indexed by (N_{f+m}). According to this, change of positions for the male spider can be modeled as follows:

$$m^{k+1}_{i} =$$

$$\begin{cases} m_i^k + \alpha * Vibf_i * (s_f - m_i^k) + \delta * (rand - \frac{1}{2}) \\ \qquad w_{N_{f+i}} > w_{N_{f+m}}; \\ m_i^k + \alpha * (\frac{\sum_{h=1}^{N_m} m_h^k * w_{N_{f+h}}}{\sum_{h=1}^{N_m} w_{N_{f+h}}} - m_i^k) \\ \qquad w_{N_{f+i}} \leqslant w_{N_{f+m}}. \end{cases} \qquad (11)$$

Where the individual (s_f) represents the nearest female individual to the male member i , whereas ($\sum_{h=1}^{N_m} m_h^k * w_{N_{f+h}} / \sum_{h=1}^{N_m} w_{N_{f+h}}$) correspond to the weighted mean of the male population (M). By using this operator, two different behaviors are produced. First, the set (D) of particles is attracted to others in order to provoke mating. Such behavior allows incorporating diversity into the population. Second, the set (ND) of particles is attracted to the weighted mean of the male population (M).

1.5 Mating operator

In SSO algorithm, Dominant members (D) and female spiders perform the mating operation. Each male spider has a specific mating radius (r), which is defined as:

$$r = \frac{\sum_{j=1}^{n} p_j^{high} - p_j^{low}}{2n}. \qquad (12)$$

During mating, the weight of spiders involved in mating that can affact the quality of the next generation of spiders. The influence probability (Ps_i) of each spider is assigned by the roulette method, which is defined as follows:

$$Ps_i = \frac{w_i}{\sum_{j \in T^g} w_j}. \qquad (13)$$

If the weight of the newly formed spider is greater than the lightest spider of the previous spider population, the new spider will replace the lightest spider in the spider population. Instead, the new spider will be abandoned and the spider population will not change.

1.6 Computational procedure

The computational procedure for the algorithm can be summarized as follow:

The Social - spider optimization algorithm (SSO)

```
Step 1: Think $N$ as the total number of $n$-dimensional colony members, define the number of male spiders ($N_m$) and female spiders ($N_f$) in the entire population ($S$).
    $N_f = floor[(0.9 - rand * 0.25) * N], N_m = N - N_f$,
Where $rand$ is a random number between [0,1], whereas $floor(.)$ maps a real number to an integer number.
Step 2: Initialize randomly the female ($F = \{f_1, f_2, \cdots, f_{N_f}\}$), male ($M = \{m_1, m_2, \cdots, m_{N_m}\}$) members (where $S = \{s_1 = f_1, s_2 = f_2, \cdots, s_{N_f} = f_{N_f}, s_{N_f+1} = m_1, s_{N_f+2} = m_2, \cdots, S_N = m_{N_m}\}$) and calculate the radius of mating.
    $r = \frac{\sum_{j=1}^{n} p_j^{high} - p_j^{low}}{2n}$.
Step 3: Calculate the weight of every spider of $S$.
    For ( $i = 1; i < N + 1; i$ ++)
        $w_i = \frac{J(s_i) - worst_s}{best_s - worst_s}$, where $best_s = \max_{k \in \{1,2,...,N\}} (J(s_k))$, $worst_s = \min_{k \in \{1,2,...,N\}} (J(s_k))$,
    End for
Step 4: Female spider's movement according to the female cooperative operator.
    For ( $i = 1; i < N_f + 1; i$ ++)
        Calculate $Vibc_i$ and $Vibb_i$
        If ( $r_m < PF$ ); where $r_m \in rand(0,1)$
            $f_i^{k+1} = f_i^k + \alpha * Vibc_i * (s_c - f_i^k) + \beta * Vibb_i * (s_b - f_i^k) + \delta * (rand - \frac{1}{2})$,
        Else if
            $f_i^{k+1} = f_i^k - \alpha * Vibc_i * (s_c - f_i^k) - \beta * Vibb_i * (s_b - f_i^k) + \delta * (rand - \frac{1}{2})$,
        End if
    End for
Step 5: Move the male spiders according to the male cooperative operator.
    Find the median male individual ( $w_{N_f+m}$ ) from $M$.
    For ( $i = 1; i < N_m + 1; i$ ++ )
        Calculate $Vibf_i$
        If ( $w_{N_f+i} > w_{N_f+m}$ )
            $m_i^{k+1} = m_i^k + \alpha * Vibf_i * (s_f - m_i^k) + \delta *$
```

```
$(rand - \frac{1}{2})$,
Else if
    $m_i^{k+1} = m_i^k + \alpha * (\frac{\sum_{h=1}^{N_m} m_h^k * w_{N_f+h}}{\sum_{h=1}^{N_m} w_{N_f+h}} - m_i^k)$,
End If
    End for
    Step 6: Perform the mating operation
    For ( $i = 1$; $i < N_m + 1$; $i$ ++ )
        If ( $m_i \in D$ )
        Find $E^i$
            If ( $E^i$ is not empty)
                From $s_{new}$ using the roulette method
                    If ( $w_{new} > w_{wo}$ )
                    $s_{wo} = s_{new}$
                    End If
        End if
      End if
    End for
    Step 7: if the stop criteria is met, the process is finished; otherwise, go to Step 3.
```

2 Differential mutation operator-based social spider optimization algorithm (SSO-DM)

2.1 Differential Mutation operator (DM)

The differential evolution (DE) developed by Storn and Price, is a stochastic search algorithm based on population cooperation and competition of individuals and has been successfully applied to solve optimization problems particularly involving non-smooth objective functions[17-22]. The optimization process in DE is carried out by combing the simple arithmetic operators with the classical evolution operators of mutation, crossover and selection to evolve from a randomly generated population to a final solution. The differential mutation operation is defined by [23].

$$x_i^{k+1} = x_i^k + (1-Y) \times (x_l^k - x_m^k) + Y \times (x_g^k - x_j^k), \quad (14)$$

Where j, l, g and m are random integers uniformly selected from the set $X = \{x_1, x_2, ..., x_n\}$ and $i \neq j \neq l \neq m \neq g$, in other word the indices are mutually different. Y is defined by

$$Y = \frac{k_{cur}}{k_{\max}}, \quad (15)$$

where k_{cur} is the current iteration and $k_{\max}$ is the number of maximum iterations.

2.2 The flight characteristics of social spiders

The flight (ballooning) is properties of many social spiders. Social spiders can achieve rapid diffusion by means of flight behavior at a certain stage. The flight is an important way to realize long distance diffusion of spiders, the flight distance sometimes can reach hundreds of kilometers. This is also the main reason for the rapid spread of social spiders. Inspired by this feature, assume that social spiders sometimes change their position according to this characteristic.

2.3 The necessary properties and theoretical analysis of the SSO-DM algorithm

(A) The necessary properties (The flight characteristics of social spiders)

The article uses differential mutation operator to simulate the flight characteristics of social spiders. Suppose that each female social spider has flight characteristic, the calculation formula of the position change of the flying characteristic of the female social spider is as follows:

$$f_i^{k+1} = f_i^k + (1-Y) \times (f_l^k - f_m^k) + Y \times (f_g^k - f_j^k), \quad (16)$$

Where j, l, g and m are random integers uniformly selected from the set $F = \{f_1, f_2, \cdots, f_{N_f}\}$ and $i \neq j \neq l \neq m \neq g$, in other word, the indices are mutually different. Y is described in Eq. (15).

Because the flight characteristics of social spiders are shown in a stage, defining a random number- Z and a threshold value- R. Where, Z is a random number between $[0, 1]$, R is a constant with a value of 0.5. When $R < Z$, female social spiders perform the operation of the flight characteristics, otherwise, female social spiders perform the operation of the original algorithm.

(B) Theoretical analysis of the SSO-DM algorithm

In SSO-DM algorithm, the parameter (Z) that controls variables for the flight characteristics of female social spiders, it always obliges the female social spiders to take random executes the operation of the flight characteristics, which greatly enhances population diversity in the algorithm and promotes exploration of the search space that leads to find diverse solutions during optimization. This mechanism is very helpful for resolving local optima stagnation even when the SSO-DM algorithm is in the exploitation phase.

The proposed SSO-DM consists essentially in a strong co-operation of the two evolutionary algorithms. The main difference with SSO is in the mutation operation. This efficient combination strategy of DM and SSO improves the global search capability, avoiding convergence to local minima and accelerates the convergence. The SSO-DM can be summarized as follow (Fig. 1):

Social-spider optimization algorithm with differential mutation operator (SSO-DM)

```
Step 1－Step 3: The same as the original algorithm.
Step 4: Female spider's movement according to the female cooperative operator.
For ( $i=1;i<N_f+1;i$ ++)
    Calculate $Vibc_i$ and $Vibb_i$
    Create $Z$ as a random number in range (0,1)
    If $Z<0.5$
        $f_i^{k+1}=f_i^k+(1-Y)\times(f_l^k-f_m^k)+Y\times(f_g^k-f_j^k)$,
    Else
    If $(r_m<PF)$; where $r_m\in rand(0,1)$
        $f_i^{k+1}=f_i^k+\alpha*Vibc_i*(s_c-f_i^k)+\beta*Vibb_i*(s_b-f_i^k)+\delta*(rand-\frac{1}{2})$,
    Else If
        $f_i^{k+1}=f_i^k-\alpha*Vibc_i*(s_c-f_i^k)-\beta*Vibb_i*(s_b-f_i^k)+\delta*(rand-\frac{1}{2})$,
    End If
    End If
End For
Step 5－Step 7: The same as the original algorithm.
```

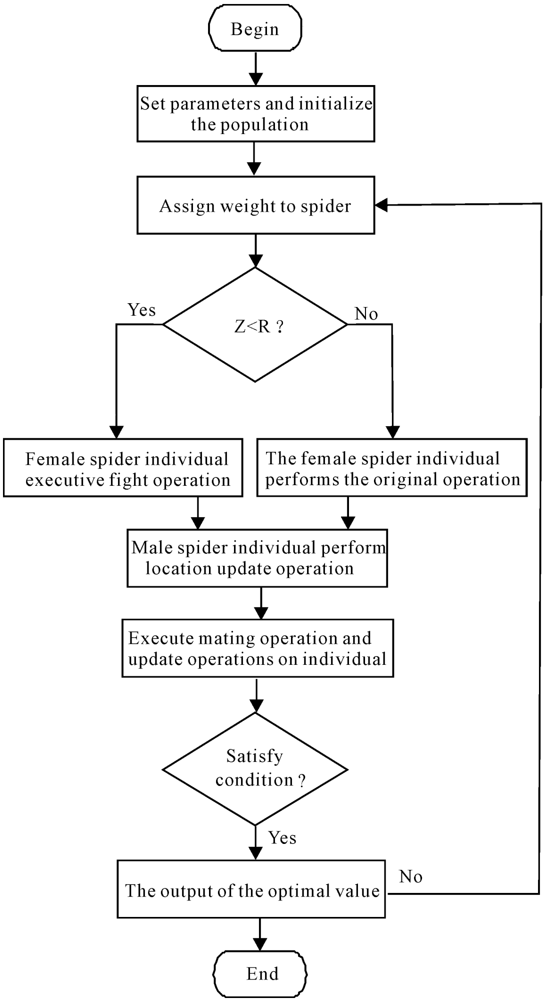
Begin
Set parameters and initialize the population
Assign weight to spider
Yes
Z<R ?
No
Female spider individual executive fight operation
The female spider individual performs the original operation
Male spider individual perform location update operation
Execute mating operation and update operations on individual
Satisfy condition ?
Yes
No
The output of the optimal value
End

Fig. 1 The flow chart of SSO-DM algorithm

3 Simulation experiments and result analysis

In this section, five standard test functions[24] are applied to evaluate the optimal performance of SSO-DM. The space dimension, scope and optimal value of five functions are shown in Table 1. The selected benchmark functions can be divided into two categories (i. e., unimodal benchmark functions and multimodal benchmark functions). They are $f_1 \sim f_3$ for unimodal benchmark functions, f_4 and f_5 for multimodal benchmark functions.

3.1 Experimental setup

All of algorithms were programmed in MATLAB R2012a, numerical experiment was set up on Inter (R) Core (TM) i5-4590 CPU and 4 GB memory.

3.2 Comparison of each algorithm performance

The proposed SSO-DM algorithm is compared with some swarm intelligence optimization algorithms, such as, SSO, ABC, BA, GGSA and DE, which use the mean and standard deviation to compare their optimal performance. The parameters settings of algorithms are as follows. For all the optimization algorithms, Population size $N=50$, and the iteration number is 1 000. 30 independent runs were made for the six algorithms, the results obtained by six algorithms are presented in Table 2, and the evolution curves and the variance diagram of f_1, f_2, f_3, f_4, f_5 obtained by six algorithms are presented in Fig. 2—6 and Fig. 7—11, respectively.

Table 1 Unimodal and multimodal benchmark functions

Functions	Dim	Range	$f_{\min}$
$f_1(x)=\sum_{i=1}^{n} x_i^2$	30	[−100,100]	0
$f_2(x)=\max_i\{\lvert x_i\rvert, 1\leqslant i\leqslant n\}$	30	[−100,100]	0
$f_3(x)=\sum_{i=1}^{n} i x_i^4 + random[0,1)$	30	[−1.28,1.28]	0
$f_4(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	30	[−5.12,5.12]	0
$f_5(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	30	[−32,32]	0

Table 2 Simulation results for test unimodal benchmark functions

Functions	Dim	Algorithm	Best	Worst	Mean	Std.
f_1	30	ABC	3.69E−08	2.32E−06	5.34E−07	5.40E−07
		BA	4.58E+02	5.79E+03	3.13E+03	1.39E+03
		GGSA	4.45E−20	4.17E−19	1.72E−19	9.49E−20
		DE	7.32E+01	1.54E+03	5.50E+02	3.37E+02
		SSO	8.63E−02	8.63E−02	8.63E−02	0
		SSO-DM	3.53E−67	3.53E−67	3.53E−67	2.01E−82
f_2	30	ABC	5.96E+01	7.39E+01	6.78E+01	3.62E+00
		BA	4.29E+01	6.89E+01	5.60E+01	6.54E+00
		GGSA	5.25E−10	1.96E−09	1.24E−09	4.04E−10
		DE	7.38E+00	2.41E+01	1.48E+01	4.39E+00
		SSO	1.47E−01	1.47E−01	1.47E−01	5.56E−17
		SSO-DM	1.20E−37	1.20E−37	1.20E−37	2.12E−53
f_3	30	ABC	4.19E−01	1.02E+01	7.09E−01	1.58E−01
		BA	6.93E−03	2.09E−2	1.23E−02	3.65E−03
		GGSA	3.90E−03	1.39E−01	1.73E−02	2.37E−02
		DE	1.75E−02	3.25E−01	1.27E−01	8.00E−02
		SSO	1.68E−02	1.68E−02	1.68E−02	0
		SSO-DM	1.22E−04	1.22E−04	1.22E−04	5.51E−20
f_4	30	ABC	1.02E+00	1.16E+01	5.47E+00	2.90E+00
		BA	1.48E+02	2.51E+02	1.99E+02	2.63E+01
		GGSA	7.95E+00	2.38E+01	1.60E+01	4.57E+00
		DE	3.43E+01	1.99E+02	1.14E+02	4.99E+01
		SSO	6.33E+01	6.33E+01	6.33E+01	0
		SSO-DM	0	0	0	0
f_5	30	ABC	9.16E−05	1.58E−03	4.79E−04	3.79E−04
		BA	1.85E+01	1.99E+01	1.90E+01	2.50E−01
		GGSA	1.48E−10	5.26E−10	3.09E−10	7.42E−11
		DE	4.44E−15	1.08E+01	6.46E+00	2.01E+00
		SSO	3.12E−01	3.12E−01	3.12E−01	5.65E−17
		SSO-DM	4.44E−15	4.44E−15	4.44E−15	0

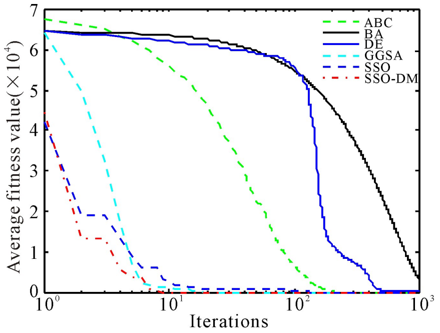

Fig. 2 Dim=30, evolution curves of fitness value for f_1

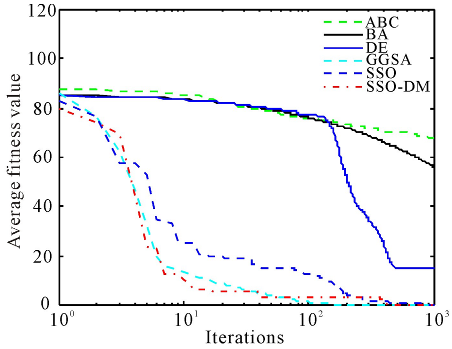

Fig. 3 Dim=30, evolution curves of fitness value for f_2

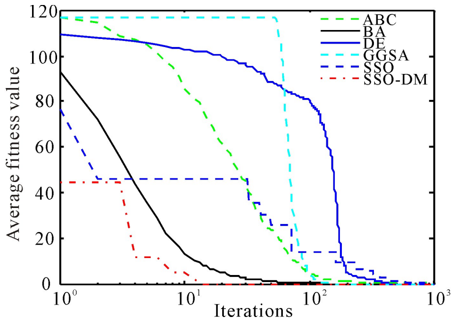

Fig. 4 Dim=30, evolution curves of fitness value for f_3

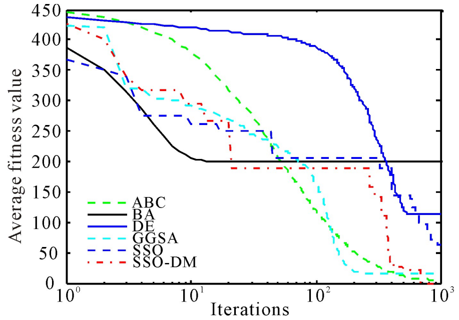

Fig. 5 Dim=30, evolution curves of fitness value for f_4

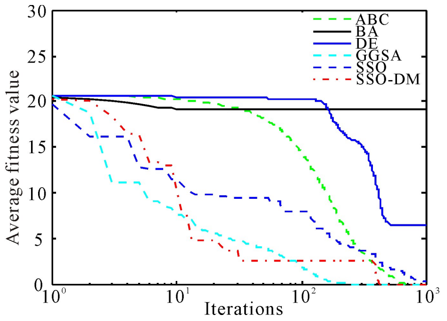

Fig. 6 Dim=30, evolution curves of fitness value for f_5

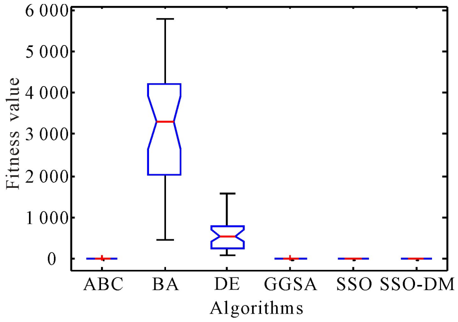

Fig. 7 Dim=30, variance diagram of fitness value for f_1

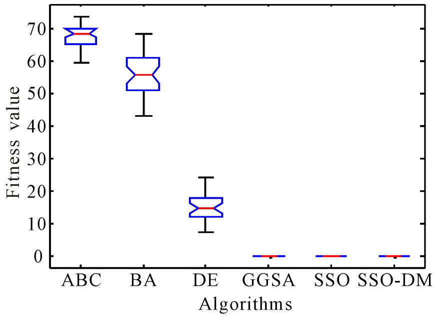

Fig. 8 Dim=30, variance diagram of fitness value for f_2

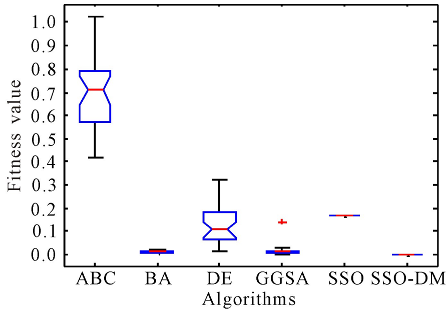

Fig. 9 Dim=30, variance diagram of fitness value for f_3

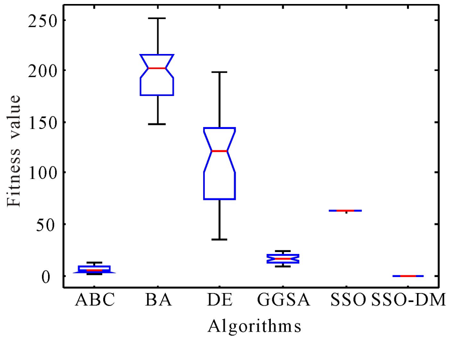

Fig. 10 Dim=30, variance diagram of fitness value for f_4

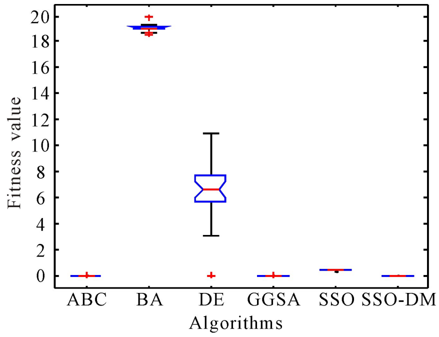

Fig. 11 Dim=30, variance diagram of fitness value for f_5

3.3 Result analysis

Seen from Table 2, in unimodal benchmark functions, SSO-DM can get a better optimal solution for f_1, f_2 and f_3 and has a very strong robustness. For the test benchmark functions, standard deviation of SSO-DM is less than that of other algorithm. And this means that in the optimization of unimodal function, SSO-DM has better stability. Similarly, seen from Table 2, SSO-DM can find the optimal solution for f_4, and the standard deviations are zeros. For f_4, f_5, the precision of mean fitness value, best fitness value, worst fitness value and standard deviation of SSO-DM are better than other algorithms. These results mean that SSO-DM has a strong searching ability and a great stability for solving multimodal function optimization. Fig. 2-6 show the evolution curves of fitness value for f_1, f_2, f_3, f_4, f_5. From these Figs, we can easily find that SSO-DM converges faster than other population based algorithms mentioned above, and the values obtained by SSO-DM are closer to the optimal value of benchmark functions. These show that SSO-DM has a faster convergence speed and a better precision than SSO and other population based algorithms. In sum, proposed SSO-DM is an algorithm with fast convergence speed, high level of precision and a great performance of stability.

Another finding in the results is the poor performance of ABC and BA. These two algorithms belong to the class of swarm-based algorithms. In contrary to SSO-DM algorithm, there is no mechanism for significant abrupt movements in the search space and this is likely to be the reason for the poor performance of ABC and BA. It is also worth discussing the poor performance of the DE algorithm in this subsection. Generally speaking, the DE algorithm has been designed based on mutation mechanism, mutation in evolutionary algorithms maintains the diversity of population and promotes exploitation, but due to the influence of the choice operation, the individual differences will gradually decrease with the increase of the number of iterations, at the same time, the decrease of individual difference also affects the diversity of the variation, which leads to the premature convergence of the algorithm, which is one of the main reasons for the poor performance of DE.

The reason for getting better fitness value provided by the SSO-DM algorithm is that this algorithm is equipped with the parameter (Z) that controls variables for the flight characteristics of female social spiders, it always obliges the female social spiders to take random executes the operation of the flight characteristics. This promotes exploration of the search space that leads to find diverse solutions during optimization. In addition, two thirds of the individuals in the SSO-DM algorithm are devoted to exploration of the search space and the rest to exploitation. This design guarantees the balance between exploration and exploitation. This is also a reason why the proposed algorithm always guides search agents to exploit the most promising regions of the search space, which also assist this algorithm to provide remarkable results.

4 Conclusions

In order to overcome the disadvantage of standard social-spider optimization algorithm, differential mutation operator has been incorporated into the SSO to generate a social-spider optimization algorithm with differential mutation operator for function optimization. Differential mutation operator enhances the diversity of the population, which helps to improve its exploration ability. From the results of the five benchmark functions, the performance of SSO-DM is better than, or at least comparable with other population-based algorithm mentioned in this paper. SSO-DM has a fast convergence speed, a relatively high degree of stability and it is also much more accurate in precision.

References:

[1] SOCHA K, DORIGO M. Ant colony optimization for continuous domains[J]. European Journal of Operational Research, 2008, 185(3): 1155-1173.

[2] STORN R, PRICE K. Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 11(4): 341-359.

[3] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of the IEEE International Conference on Neural Networks. Piscataway: IEEE, 1995: 1942-1948.

[4] YANG X S. Multiobjective firefly algorithm for continuous optimization[J]. Engineering with Computers, 2013, 29(2): 175-184.

[5] ZHOU Y Q, LIU J K, ZHAO G W. Leader glowworm swarm optimization algorithm for solving nonlinear equations systems[J]. Przeglad Elektrotechniczny, 2012, 88(1b): 101-106.

[6] MUCHERINO A, SEREF O. Monkey search: A novel metaheuristic search for global optimization[C]//Proceedings of the American Institute of Physics Conference Gainesville. USA: American Institute of Physics, 2007: 162-173.

[7] ALATAS B. Chaotic harmony search algorithms[J]. Applied Mathematics and Computation, 2010, 216(9): 2687-2699.

[8] YANG X S, DEB S. Cuckoo search via Levy flights [C]//Proceedings of the World Congress on Nature and Biologically Inspired Computing. Coimbatore, India: IEEE, 2009: 210-214.

[9] YANG X S. A new metaheuristic bat-Inspired algorithm[M]//GONZÁLEZ J R, PELTA D A, CRUZ C (eds.). Nature Inspired Cooperative Strategies for Optimization. Berlin, Germany: Springer-Verlag, 2010: 65-74.

[10] CUEVAS E, CIENFUEGOS M, ZALDÍVAR D, et al. A swarm optimization algorithm inspired in the behavior of the social-spider[J]. Expert Systems with Applications, 2013, 40(16): 6374-6384.

[11] LUBIN T, BILDE T. The evolution of sociality in spiders[J]. Advances in the Study of Behavior, 2007, 37: 83-145.

[12] UETZ G W, HIEBER C S. Colonial web-building spiders: Balancing the costs and benefits of group living [M]//CHOU J C, CRESPI B J (eds.). The Evolution of Social Behavior in Insects and Arachnids. Cambridge, England: Cambridge University Press, 1997.

[13] AVILES L. Sex-ratio bias and possible group selection in the social spider *Anelosimus eximius* [J]. The American Naturalist, 1986, 128(1): 1-12.

[14] BURGESS J W, UETZ W G. Social spacing strategies in spiders[M]//WITT P N, ROVNER J S (eds.). Spider Communication: Mechanisms and Ecological Significance. Princeton, New Jersey: Princeton University Press, 1982.

[15] SALOMON M, SPONARSK C, LAROCQUE A, et al. Social organization of the colonial spider *Leucauge* sp. in the Neotropics: Vertical stratification within colonies

[J]. Journal of Arachnology,2010,34(3):46-451.

[16] YIP E C,POWERS K S,AVILS L. Cooperative capture of large prey solves scaling challenge faced by spider societies[J]. Proceedings of the National Academy of Sciences of the United States of America,2008,105(33):11818-11822.

[17] BOLTE J,DANIILIDIS J A,LEWIS A. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems[J]. SIAM Journal on Optimization,2006,17(4):1205-1223.

[18] RAVIKUMAR K,ANAND S,SYDULU M. A novel multi-agent based PSO approaches for security constrained optimal power flows using smooth and non-smooth cost functions[J]. International Journal of Computer Applications,2012,41(3):14-21.

[19] VAISAKH K,SRINIVAS L R. Adaptive PSODV algorithm for OPF with non-smooth cost functions and statistical analysis[J]. Simulation Modelling Practice and Theory,2011,19(9):1824-1846.

[20] SAYAH S,ZEHAR K. Modified differential evolution algorithm for optimal power flow with non-smooth cost functions[J]. Energy Conversion and Management,2008,49(11):3036-3042.

[21] BURGES C J C,RAGNO R,LE Q V. Learning to rank with non-smooth cost functions[C]//Proceedings of the 19th International Conference on Neural Information Processing Systems. Cambridge:MIT Press,2006.

[22] NEYESTANI M,FARSANGI M M,NEZAMABADI-POUR H. A modified particle swarm optimization for economic dispatch with non-smooth cost functions[J]. Engineering Applications of Artificial Intelligence,2010,23(7):1121-1126.

[23] LU S F,SUN C F,LU Z D. An improved quantum-behaved particle swarm optimization method for short-term combined economic emission hydrothermal scheduling[J]. Energy Conversion and Management,2009,51(3):561-571.

[24] MIRJALILI S,MIRJALILI S M,LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software,2014,69:46-61.

（责任编辑:米慧芝）