Soft Computing
https://doi.org/10.1007/s00500-018-3662-3

METHODOLOGIES AND APPLICATION

Memetic frog leaping algorithm for global optimization

Deyu Tang[1,2] · Zhen Liu[1,4] · Jin Yang[1,2] · Jie Zhao[3]

© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract
Developing an effective memetic algorithm that integrates leaning units and achieves the synergistic coordination between exploration and exploitation is a difficult task. In this paper, we propose a memetic algorithm based on the shuffled frog leaping algorithm, which is fulfilled by three units: memetic diffusion component, memetic evolutionary component and memetic learning component. Memetic diffusion component enhances the diversity of population by the shuffled process. Memetic evolutionary component accomplishes the exploitation task by integrating the frog leaping rule, geometric center, Newton's gravitational force-based gravitational center and Lévy flight operator. Memetic learning component improves the exploration by an adaptive learning rule based on the individual selection and the dimension selection. In order to evaluate the effectiveness of the proposed algorithm, 30 benchmark functions and a real-world optimization problem are used to compare our algorithm against 13 well-known heuristic methods. The experimental results demonstrate that the performance of our algorithm is better than others for the continuous optimization problems.

Keywords Memetic algorithm · Shuffled frog leaping algorithm · Gravity search algorithm · Lévy flight · Continuous optimization

1 Introduction

Optimization problems are frequently involved in the areas of engineering and science. Many mathematical programming methods, such as fast steepest method, conjugate gradient method and quasi-Newton method, have been used to solve optimization problems. However, most of real-world optimization problems exhibit multimodal or non-continuous, etc., which are difficultly solved by traditional methods. Each optimization problem is similar to a 'black box,' in which we can only know the input variables. To this end, many population-based algorithms were developed rapidly in the past decades (Vasan 2014; Vasant et al. 2016). Kennedy and Eberhart (1995) proposed the particle swarm optimization (PSO), which was inspired by the food foraging behavior of bird flocks or fish school. After that, differential evolution (DE) was proposed by Storn and Price (1997), which is a simple and powerful evolutionary algorithm (EA) solving the continuous optimization problems by the mutation operator, crossover operator and selection operator. Owning to its excellent performance, adaptive DE algorithms have been developed in Brest et al. (2006), Qin et al. (2009) and Zhang and Sanderson (2009). And then, shuffled frog leaping algorithm (SFLA) (Eusuff and Lansey 2003) was proposed to simulate the frog foraging behaviors. Artificial bee colony algorithm (ABC) (Karaboga 2005) was presented to simulate the process of exploiting food sources by the employed bees, onlookers and scout bees. Biogeography-based optimization algorithm (BBO) (Simon 2008) is an evolutionary algorithm motivated by the optimality perspective of natural biogeography and was initially developed by Simon (2008). Cuckoo Search algorithm (CS) (Yang and Deb 2009) emulates the brood parasitism behavior of cuckoos. Gravitational search algorithm (GSA) (Rashedi et al. 2009) was inspired from the Newton's law

Communicated by V. Loia.

✉ Deyu Tang
scutdy@126.com

[1] School of Medical Information and Engineering, Guangdong Pharmaceutical University, Guangzhou 510006, China

[2] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

[3] Department of Information Management Engineering, School of Management, Guangdong University of Technology, Guangzhou 510520, China

[4] Department of Computer Science, American University, Washington, DC 20016, USA

Published online: 03 December 2018

of gravity and motion. Teaching–learning-based optimization (TLBO) (Rao et al. 2012) simulates the process of teaching and learning of people to achieve a good score by teaching operator and learning operator. Gray wolf optimizer (GWO) (Mirjalili et al. 2014) was inspired from the social hunting behavior of gray wolves usually depending on their leadership hierarchy and hunting activities. Artificial algae algorithm (AAA) (Uymaz et al. 2015) was inspired by the living behaviors of microalgae. And invasive tumor growth optimization algorithm (ITGO) (Tang et al. 2015) simulates the invasive behaviors of tumor by the proliferative cells, quiescent cells and dying cells.

Shuffled frog leaping algorithm is a simple and efficient algorithm. It has been widely applied to many real-world problems such as the traveling salesman problem (TSP) (Luo et al. 2008), grid task scheduling problem (Ou and Sun 2011), economic dispatch problem (Narimani 2011; Roy et al. 2013), resource-constrained project scheduling problem (Wang and Fang 2011; Fang and Wang 2012), flow shop scheduling problem (Lei and Guo 2015; Li et al. 2012a), vehicle routing problem (Luo et al. 2015), 0/1 knapsack problem (Bhattacharjee and Sarmah 2014). However, the basic shuffled frog leaping algorithm is easy to fall into the local optimum, especially for the continuous optimization problem. To handle this problem, some improved methods have been presented. Li et al. (2012b) tuned the parameters of leaping operator and preserved the historical information, so as to enhance the global search ability. In addition, they utilized an extremal optimization operator with gradient information to enhance the local search ability. Ahandani and Alavi-Rad (2015) used the opposition-based learning (OBL) strategy for the SFLA, which employs the OBL to accelerate the search and to improve the diversity of population. Ahandani (2014) proposed a diversified shuffled frog leaping algorithm, in which a differential operator is used for evolutionary process. Cheng et al. (2014) used quantum bit as a position of frog and obtained a quantum operator by the local best frog and the center of population. Sharma et al. (2015) embedded a centroid mutation strategy into the leaping operator, so as to enhance the diversity of population. The literature (Wang et al. 2015) maintained the historical information and appended the global information for the leaping operator and used crossover and mutation operator meanwhile. The literature (Liu et al. 2018a) combined the chaotic operator and opposition-based learning operator to produce the initial population. In addition, an adaptive nonlinear inertia weight and a perturbation operator strategy based on Gaussian mutation for global (local) best frog were performed to ensure the balance of exploration and exploitation. The literature (Liu et al. 2016) combined with the excellent characteristics of cloud model that enhances the accuracy and convergence speed of SFLA.

As mentioned above, the basic SFLA is easy to fall into the local optimum due to the weakness of exploitation and loss of exploration. The key of population-based algorithm is to balance exploration and exploitation for achieving high accuracy and fast convergence speed. However, the current improvements of SFLA focus on the frog leaping rules relying on the PSO framework. To this end, we attempt to propose a novel framework that can achieve the synergistic coordination between exploitation and exploration.

The contribution of this study is as follows.

1. We propose a new memetic framework of SFLA, which fulfills the balance of exploration and exploitation by the collaboration of multi-agents and multi-operators.
2. We devise a strong local searcher with multi-'attractors' and adaptive selection strategy due to the weakness of exploitation in the old versions of SFLA.
3. We devise a global searcher with an adaptive learning strategy due to the loss of exploration in the old versions of SFLA. The proposed approach is synergistic coordination between exploitation and exploration.

The rest of the paper is organized as follows. Section 2 reviews the related work on memetic algorithm. Section 3 represents the shuffled frog leaping algorithm. Section 4 describes the proposed approach. Section 5 shows the experimental simulation, results, and analysis. Finally, Sect. 6 concludes this paper.

2 Related work

The word 'meme' was firstly proposed by the philosophical theory of Richard Dawkins (Dawkins 1976), that can be duplicated in human brains, modified, and combined with other memes in order to generate a new meme. Different from 'gene,' it is also called the 'culture gene.' Inspired by Darwin's principles of natural evolution and Richard Dawkins's principles of cultural evolution, the first memetic algorithm (MA) was developed by Moscato and Norman (1989), and Moscato (1989). It is a modification of genetic algorithms (GAs) employing a local search operator for addressing the traveling salesman problem (TSP). And then, a massive diffusion of MAs occurred only ten years due to the diffusion of the No Free Lunch Theorem (NFLT) (Wolpert and Macready 1997). NFLT proves that there is no a meta-heuristic method that can solve all the optimization problems.

Nowadays, MAs focus on the synergistic coordination between exploration and exploitation (Le et al. 2012). MA can be seen as a population-based algorithm, which is coupled with one or more individual learning units (local

search) (Tang et al. 2014), coevolution (Smith 2007a, b), local surrogate models (Zhou et al. 2007), as it includes several algorithmic schemes characterized by various levels of adaptation and decision-making (Ong et al. 2006a; Nguyen et al. 2009), or purposely simple cascades of single-solution search units (Iacca et al. 2012). The framework of the MA is proposed in Ong and Keane (2004) so that the best local search method can be chosen adaptively from local search pool according to past merits in generating local improvements. Samma et al. (2016) presented a memetic algorithm, which was integrated by the particle swarm optimization (PSO) algorithm and a reinforcement learning (RL) algorithm as a local search method. Wang et al. (2012) proposed a memetic algorithm that hybridizes particle swarm optimization (PSO) with two different local search operators in a cooperative way for locating multiple global and local optimal solutions in the fitness landscape. Li et al. (2014) proposed a memetic algorithm with double mutation operators (MADM) to deal with the problem of global optimization. It combines two meta-learning systems to improve the ability of global and local exploration. In Bambha et al. (2004), the parameters of local search are systematically adjusted. Detailed taxonomy and comparative study on adaptive choice of local search methods or memes in MA can be found in Ong et al. (2006b). A multirestart MA framework was proposed in Sun et al. (2013). In this framework, an EA and a local optimizer are employed as separated building blocks.

The meme refers to cultural evolution that was capable of local/individual refinement (Ong et al. 2010). Kóczy et al. (2017) proposed a novel memetic algorithm, named discrete bacterial memetic evolutionary algorithm (DBMEA). It is based on the combination of the bacterial evolutionary algorithm and local search techniques. A hybrid global–local heuristic search methodology was proposed, in which genetic algorithm is adapted as the global search and the proposed two-level learning strategy as the local search (Liu et al. 2018b). Some hybridization approaches in MA have been proposed and proved to be effective. In Müller et al. (2009), MA was combined with particle swarm optimizer (PSO) as the global search and covariance matrix adaptation evolution strategy (CMA-ES) as the local search. The hybridization of barebones PSO with differential evolution (DE) is shown in Omran et al. (2007). Meanwhile, co-evolving MA was developed where rule-based representation of local search is co-adapted alongside the problem representation (Smith 2007a, b).

3 Shuffled frog leaping algorithm

Shuffled frog leaping algorithm (Eusuff and Lansey 2003) is a heuristic search approach that mimics the foraging behavior of frogs. It can be seen as a multiple subpopulations evolutionary algorithm based on a simple search rule. It is similar to the PSO. In the frog population, each frog can communicate and discuss different ideas with other frogs which corresponds to the global search operator, and the worst frog can jump to find the best foods guided by the best frog which corresponds to the local search operator. For the convenience of communication and discussion, the whole population is divided into m subpopulations (or groups). The frogs in each subpopulation have almost the same ability, which is fulfilled by a shuffled process for the multiple subpopulations. In a predefined number of local search steps, only the worst frog can be updated by the best frog in each subpopulation, which can be the frog jumping rule similar to the PSO. The shuffled operator and the frog jumping operator are used alternately until the defined convergence criterion is satisfied. Actually, the shuffled operator can only divide the whole population into different subpopulations but cannot update the position of each frog. It can be described as follows:

$$P = \{Q_k | Q_k = Q_{i+m(j-1)},\ i = 1, 2, \ldots m,\ j = 1, 2, \ldots, n,\ k = 1, 2, \ldots, \text{popsize}\ \} \quad (1)$$

The population P is divided into m subpopulations according to the fitness of each frog sorted in descending order. n is the number of frogs in each subpopulation and popsize $= m \times n$.

After the shuffled process, the worst frog can be updated according to the guidance of the best frog in each subpopulation, which is described as the following equation:

$$Q' = Q_w + rand.(Q_{\text{best}} - Q_w) \quad (2)$$

where Q_w and Q_{best} represent the position of the worst frog and the best frog, respectively, and *rand* denotes the random number of uniform distribution $rand \sim U(0,1)$. Q_{best} is the local best solution in each subpopulation. If the old Q_w cannot be updated with the worst fitness value, then the Q_{best} can be replaced by the global best frog Q_g. If the old Q_w still cannot be updated, then it can be replaced by a new solution generated in the whole search space. The search operator of Q_w is continued until the threshold value reaches the predefined number of iterations in each subpopulation. The shuffled operator and the frog jumping operator run alternately until a predefined convergence criterion is satisfied.

According to the presentation of SFLA, we can find that it can be seen as a local search optimizer. For SFLA, the search of each frog depends on some best frogs in different

groups. The shuffled process, called the global search step, is not the real global search operator. It cannot fulfill the exploration process but can enhance the diversity of population for exploitation. This means that it has the best local search ability, but it is weak for the global search. Through these analyses, the shuffled process can be designed as the memetic diffusion component and other units are required to be combined with shuffled process. To solve these problems, a new MA based on SFLA can be considered for design.

4 The proposed approach

In this section, we summarize the advantages and disadvantages of SFLA and MA, and propose a novel MA called memetic frog leaping algorithm (MFLA) based on the SFLA. It is integrated by the three different units: memetic diffusion component (MD), memetic evolutionary component (ME) and memetic learning component (ML). MD is fulfilled by a shuffled strategy according to the meme communication, which can enhance the diversity of population for the exploitation process. ME is fulfilled by a local-center-based frog leaping rule which has more 'attractors' provided by MD, mutation characteristic and adaptive ability. It is responsible for the exploitation task for MFLA. ML is the third important component, and it is fulfilled by an adaptive learning strategy for the exploration. MFLA can be seen as a 'small system' for solving optimization problems. MD, ME and ML are not independent but cooperative, which achieve the synergistic coordination between exploitation and exploration.

In MFLA, each frog in the population has the idea similar to people. To find better food, they can communicate to each other by the meme diffusion, evolution in some memetic scene and learn in meme transformation.

4.1 Memetic diffusion process (MD)

In real life of people, the meme (an important information unit) diffusion is common in many fields, such as culture, entertainment, education, academia and so on. For instance, students in school are able to lean knowledge rapidly by discussing or exchanging information (meme) among different kinds of groups. Similar to people, the frogs in a population are very clever and can learn by the meme diffusion. We adopt the shuffled process of SFLA in a simple way. In each iteration, the population can be divided into different memeplexes (groups) according to the fitness values and the frogs in each memeplex have almost equal ability. Memes are diffused in each memeplex by periodically redividing the population. It is achieved as Eq. (1) in Sect. 2. An example of the shuffled process is shown in Fig. 1. First, all the frogs are divided according to their fitness values. Then, the number of memeplexes and frogs in each memeplex is given. In this example, there are m ($m = 4$) memeplexes shown by different colors (yellow, blue, green and red) and n ($n = 4$) frogs in each memeplex. There are 16 ($ps = m*n = 16$) frogs in the population. Q_3 is a local best frog in a memeplex (green), and Q_{15} is a local worst frog in a memeplex (green). Q_{11} is an ordinary frog, and it can be arranged into the green memeplex according to Eq. (1), i.e., $Q_{11} = Q_{3+4*(3-1)}$ (i = 3, j = 3).

4.2 Memetic evolution (ME)

Memetic evolution is achieved by the frog leaping operator. After the meme diffusion process, the worst frog in each memeplex hopes that it can leap to the position of the best frog. In order to enhance the exploitation ability, the local-centers-based frog leaping rules (adaptive and mutational) are as follows.

4.2.1 Search strategy by geometric center and gravitational center

In this subsection, we propose the concept of geometric center and gravitational center.

Definition 1 Geometric center is the average position of the n individuals.

$$Q_c = \frac{1}{n}\sum_{j=1}^{n} Q_j \tag{3}$$

where n is the number of frogs in subpopulation and Q_c is the geometric center.

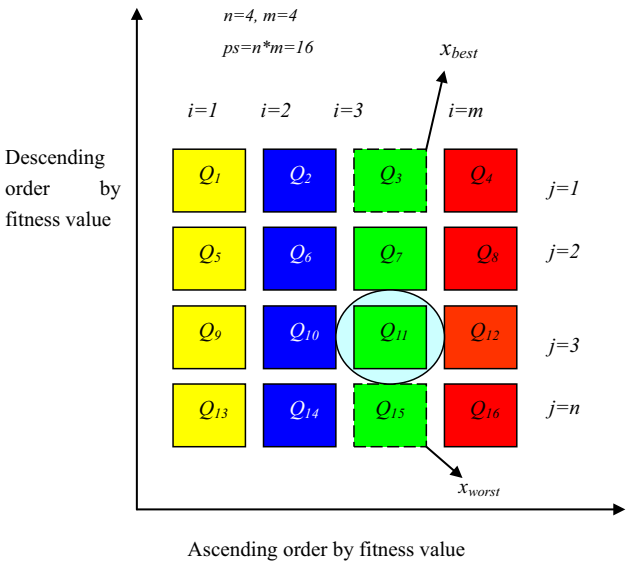

Fig. 1 Shuffled process

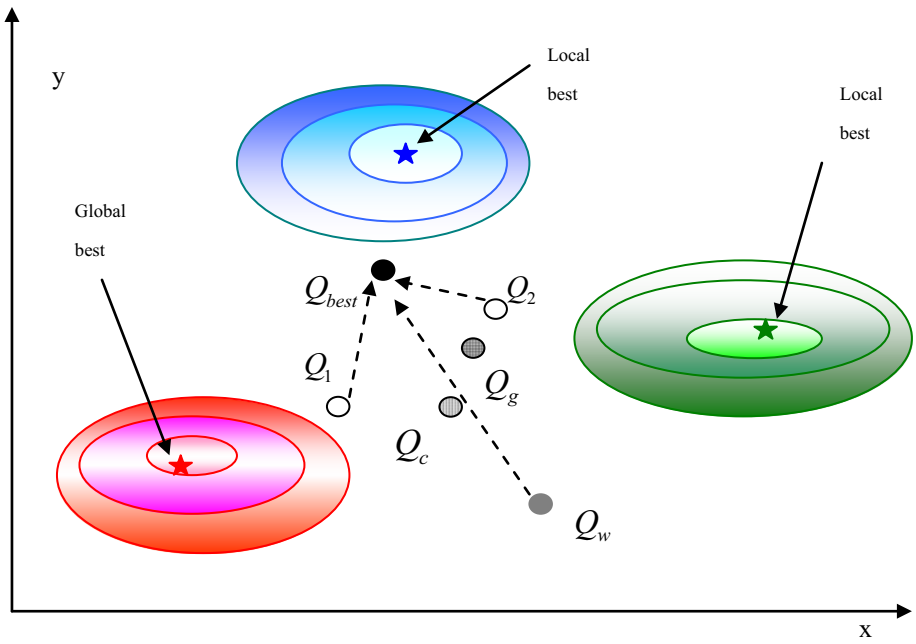

Fig. 2 Search with geometric center and gravitational center

In the Newton law of gravity (Holliday et al. 1993; Schutz, 2003), each particle attracts every other particle with a 'gravitational force.' The gravitational force between two particles is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. In high-dimensional space, it can be represented as follows:

$$F_{ij} = G_0 \frac{M_i M_j}{R_{ij}^2 + \varepsilon} \tag{4}$$

$$R_{ij} = ||x_i - x_j||_2 \tag{5}$$

where M_i and M_j indicate the gravitational mass of particle i and particle j, respectively, x_i and x_j represent the position of the particle i and particle j in high-dimensional space, respectively, R_{ij} is the Euclidean distance between x_i and x_j, and ε is a small constant. The fitness of particle is used for the gravitational mass of particle.

$$m_i = \frac{\text{fit}_i - \text{worstfit}}{\text{bestfit} - \text{worstfit}} \tag{6}$$

$$M_i = \frac{m_i}{\sum_{i=1}^{n} m_i} \tag{7}$$

where fit_i represents the fitness value of the particle i, and worstfit_i and bestfit_i represent the worst and the best fitness value in the population of n particles, respectively. m_i can be seen as the score of fitness value, and the mass value M_i can be represented by the normalized m_i in [0, 1] as Eq. (7).

Definition 2 Gravitational center is the product of gravity ratio and the position of n individuals

$$Gf_{ij} = \frac{G_0 \frac{M_i M_j}{R_{ij}^2+\varepsilon}}{\sum_{j=1}^{n} G_0 \frac{M_i M_j}{R_{ij}^2+\varepsilon}} (j \neq i) = \frac{\frac{M_j}{R_{ij}^2+\varepsilon}}{\sum_{j=1}^{n} \frac{M_j}{R_{ij}^2+\varepsilon}} (j \neq i) \tag{8}$$

where G_0 is the value of the gravitational constant, Gf_{ij} represents the ratio of the magnitude of the gravitational force between frog i and frog j. n is the number of frogs in subpopulation.

$$Q_g = \sum_{j=1, j\neq i}^{n} Gf_{ij} \cdot Q_j \tag{9}$$

where Q_j represents the position of frog j, n is the number of frogs in subpopulation, and Q_g represents the gravitational center.

Then, the new search leaping rule with gravitational center and geometric center can be represented as follows:

$$Q_m = \begin{cases} Q_g, & \text{if } rand < 0.5 \\ Q_c, & \text{else} \end{cases} \tag{10}$$

$$Q'_w = Q_w + rand.(Q_{\text{best}} - Q_w) + rand.(Q_m - Q_w) \tag{11}$$

where Q_w represents the worst frog and Q_{best} is the best frog in subpopulation.

Suppose there are three local best solutions for a multimodal optimization problem in two-dimensional space. The search strategy by geometric center and gravitational center is shown in Fig. 2. It shows that the 'red star' is the global best solution, and the 'blue star' and the 'green star' are the two local best solutions, respectively. At this time, four frogs (Q_w, Q_{best}, Q_1 and Q_2) in the population are used. The best frog Q_{best} obtains the best fitness value in the

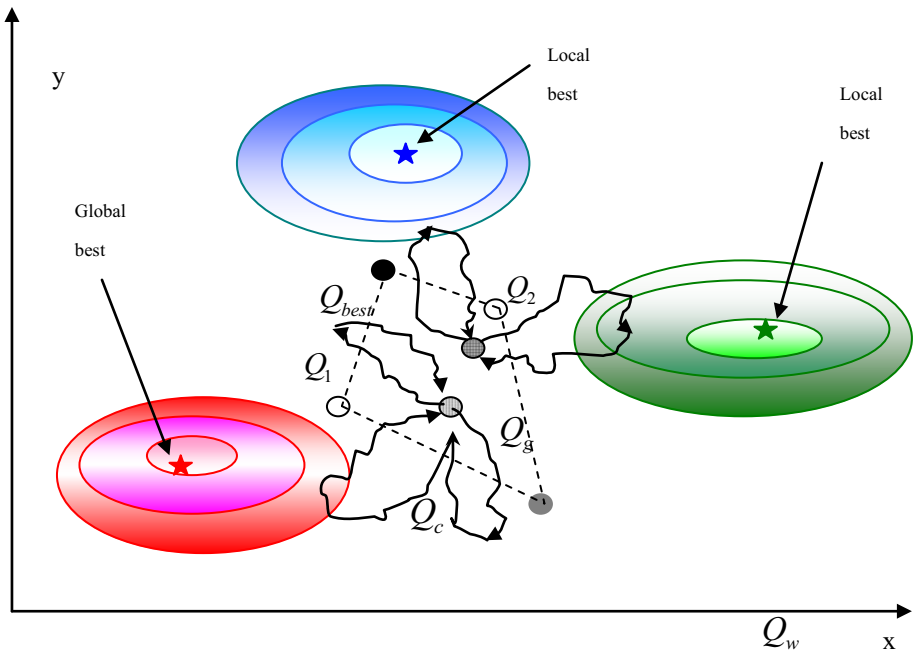

Fig. 3 Lévy flight for geometric center and gravitational center

population. It is unfortunate that the frog Q_{best} (the black ball) is near the local best position (the blue star) that is not the global best position (the red star). It is obvious that the worst frog Q_w is very easy to fall into the local optimum due to the traditional leaping rule only guided by the best frog Q_{best}. Hence, we create two new frogs as new 'attractors' called geometric center (Q_c) and gravitational center (Q_g), respectively. The gravitational center (the grid ball, Q_g) is produced by the gravitational force between the best frog Q_{best} and the other three frogs (Q_w, Q_1, Q_2) according to the Newton law of gravity. The dotted line means the gravitational force between the best frog Q_{best} and the other three frogs (Q_w,Q_1, Q_2). Another 'attractor' (the vertical line ball, Q_c) is produced by the geometric center. We can see that the geometric center Q_c is near the global best solution and the gravitational center Q_g is near another best solution (the green star). In our search strategy, Eq. (10) shows that the two 'attractors' are chose alternately to guide the worst frog Q_w to move for the 'red star' and 'the green star' not only move for 'the blue star.' We can see that it can avoid falling into the local optimum.

4.2.2 Lévy flight for mutation of the two centers

In real life, the behavior of Lévy flight occurs in many natural phenomena, such as the flying of albatross, the flying of Drosophila, the feeding way of other animals, and even the hunting mode of human. Lévy flight shows the characteristic of the search in short distance occasionally and in long distance. So, it is suited to solve some optimization problems and has been widely used in some heuristic algorithms, such as CS (Yang and Deb 2009). So, the Lévy flight is considered as the mutation operator for the two 'attractors': geometric center and gravitational center.

$$\text{LevyFlight} = rand(0,1) \cdot \text{normal}(0,1) \cdot \text{Levy} \tag{12}$$

where $rand(0,1)$ represents the random number of uniform distribution $rand \sim U(0,1)$. $\text{normal}(0,1)$ means the random number of normal distribution, and $Levy$ represents the random number of Lévy distribution in high-dimensional space(Yang and Deb 2009).

$$Q_w^{'} = Q_w + rand.(Q_{best} - Q_w) + rand.(\text{LevyFlight} \otimes Q_m - Q_w) \tag{13}$$

where $\otimes$ is an element-wise multiplication.

In Fig. 3, we can see that geometric center and gravitational center are limited in the population space. In this case, geometric center (Q_c) and gravitational center (Q_g) are limited in the quadrangle space by the four positions of two frogs (Q_w, Q_{best}, Q_1, Q_2). The dotted lines represent the quadrangle space (shown in Fig. 2), which reduces the diversity of population. This indicates another weakness of local search. The geometric center and the gravitational center cannot guarantee that the worst frog (Q_w) can jump out the local best due to the smaller quadrangle space. To this end, Lévy flight is applied for the mutation of geometric center (Q_c) and gravitational center (Q_g). In Lévy flight, the local search of short walking distance and the occasionally global search of longer walking distance occur, so some solutions of the near optimum value can be

FES=0;//fitness evaluation number
Randomized initialization *(Q^1_d Q^2_d ,..., Qx^k_d)* and evaluate fitness values *$f(Q_k)$*, for *k=1,2,...,ps*.
FES=FES+ps;
While *FES<=MAX_FES//the max fitness evaluation number*
// exploitation and shuffled process
Sort and arrange population **(ps=m*n)** according to the fitness, **n** is the number of memeplexes, and **m** is the number of frogs in each memeplex.
For *i =1 to* ***n*** **do**// ***n*** is the iteration number, which is equal to the number of frogs in each memeplex
For *j=1 to* ***m*** do // ***m*** memeplexes
Get the worst and best frog $\boldsymbol{Q_{worst}}$, $\boldsymbol{Q_{best}}$ in the $\boldsymbol{j_{th}}$ memeplex;
r_1=rand;
r_2=rand;
Q_c =(1/m)sum($Q_{j+m(k-1)}$); k=1,2,..n, Geometric center
$R_{j+m(k-1)}$ =sqrt(sum($Q_{best,d}$ –$Q_{j+m(k-1),d}$)^2);// k=1,2,..n ,Euclidean distance
$m_{j+m(k-1)}$ =($fit_{j+m(k-1)}$ -worst)/(bestfit-worstfit); // k=1,2,..n, compute mass
$M_{j+m(k-1)}$ = $m_{j+m(k-1)}$ /sum($m_{j+m(k-1)}$); // k=1,2,..n , normalization
$Gf_{j+m(k-1)}$ =($M_{j+m(k-1)}$ / ($R_{j+m(k-1)}$ +eps))/(sum($M_{j+m(k-1)}$ / ($R_{j+m(k-1)}$ +eps))); // k=1,2,..n-1
// j+m(k-1)≠best, 'best' is the index number of $\boldsymbol{Q_{best}}$, eps *is a minimum floating-point number*
Q_g =sum($Gf_{j+m(k-1)}$.* $Q_{j+m(k-1)}$); //k=1,2,..n-1.
If *rand<0.5*
Q_m = Q_c; //Geometric center
Else
Q_m = Q_g; //Gravitational center
End if
sigma=(gamma(1+beta)*sin(pi*beta/2)/(gamma((1+beta)/2)*beta*2^((beta-1)/2)))^(1/beta);
//***beta*** *is a parameter,* ***gamma()*** *produces the random number of gamma distribution*
u=randn(D)*sigma; //D is the dimension ,randn() produces the random
vw=randn(D);
levy=u./abs(vw).^(1/beta);
LevyFligh= rand .*(randn) .* levy;

For *d=1 to* **D do**
$Q'_{worst,d}$= (1- r_1)Q_{worst} + r_1 .Q_{best}+r2.(LevyFligh .*Q_m -Q_{worst});
End For
If f($\boldsymbol{Q'_{worst}}$) <f($\boldsymbol{Q_{worst}}$)
Q_{worst} = $Q'_{worst;}$
End if
FES=FES+1;
End for //m memeplexes
End for //n frogs
// exploration process
For *i=1* to *ps*
map=zeros(1, D); //D is dimension
u=randperm (D); // produces D random integers
map(i,u(1:ceil(rand*M)))=1;//dimension selection
A= randperm(ps);// produces ps random integers
B= randperm(ps); // produces ps random integers
j =A(i);//select an individual from population
k=A(i);// //select an individual from population
Q'_i= Q_i +rand.*map.*(Q_j - Q_k);
If f(Q'_i) <f(Q_i)
Q_i = Q'_i ;
End if
FES=FES+1;
End
End while

◀**Fig. 4** MFLA algorithm

searched and local search ability can be enhanced. Figure 2 shows that the possible search tracks the geometric center(Q_c) and the gravitational center(Q_g) by the Lévy flight, in which the two 'attractors' (Q_c and Q_g) can walk out of the designated quadrangle space. That is, Lévy flight ensures the two 'attractors' search in an irregular search space, which can enhance the search ability and the diversity of population.

4.3 Memetic learning method (ML)

Memetic learning method is a global learning process with the informal communication and discussion for frog population. Contrast to MD, meme transmission of the frogs in population is not required to be divided in ML. In ML, each frog can learn knowledge by the communication and discussion of the optional two frogs in the population. And it is an adaptive learning strategy without any parameter.

The motivation comes from the following two points:

1. The search should depend on the interaction among all the frogs in population.
2. Considering the exploration and diversity, how to search in more wide space including individual's selection and dimension's selection?

Based on this idea, we consider the approximate discrete uniform distribution for the individual selection and dimension selection. Though it is similar to the DE algorithm, it is an adaptive learning strategy without any parameters. The detail is as follows:

$$u_i = randperm(D) \tag{14}$$

$$popA = randperm(popSize) \tag{15}$$

Table 1 Details of benchmark problem

Fun	Benchmark problem	Type	Low	Up	Dim	Optimum values
F1	Sphere function	U	− 100	100	30	0
F2	Schwefel's problem 2.22	U	− 10	10	30	0
F3	Schwefel's problem 1.2	U	− 100	100	30	0
F4	Quartic function	U	− 1.28	1.28	30	0
F5	Rastrigrin function	M	− 5.12	5.12	30	0
F6	Ackley function	M	− 32	32	30	0
F7	Griewank function	M	− 600	600	30	0
F8	Rosenbrock function	M	− 10	10	30	0
F9	Penalized function	M	− 50	50	30	0
F10	Weierstrass function	M	− 0.5	0.5	30	0
F11	Zakharov function	M	− 5	10	30	0
F12	Alpine function	M	− 10	10	30	0
F13	Salomon problem	M	− 100	100	30	0
F14	Periodic problem	M	− 10	10	30	0.9
F15	Inverted cosine mixture problem	M	− 1	1	30	0
F16	Rotated Bent Cigar function	M	− 100	100	30	200
F17	Rotated discus function	M	− 100	100	30	300
F18	Shifted and Rotated Rosenbrock function	M	− 100	100	30	400
F19	Shifted and Rotated Weierstrass function	M	− 100	100	30	600
F20	Shifted and Rotated Griewank's function	M	− 100	100	30	700
F21	Shifted and Rotated Katsuura function	M	− 100	100	30	1200
F22	Shifted and Rotated HappyCat function	M	− 100	100	30	1300
F23	Hybrid function 2 (N = 3)	M	− 100	100	30	1800
F24	Hybrid function 4 (N = 4)	M	− 100	100	30	2000
F25	Composition function 1 (N = 5)	M	− 100	100	30	2300
F26	Composition function 2 (N = 3)	M	− 100	100	30	2400
F27	Composition function 3 (N = 3)	M	− 100	100	30	2500
F28	Composition function 4 (N = 5)	M	− 100	100	30	2600
F29	Composition function 5 (N = 5)	M	− 100	100	30	2700
F30	Composition function 6 (N = 5)	M	− 100	100	30	2800

Table 2 Parameters setting

No.	Algorithm	Parameter setting
1	SPSO2011	w = 1/(2*log(2));
		c1 = 0.5 + log(2); c2 = c1;
2	DE/best/2/bin	F = 0.5,CR = 0.9
3	GbABC	SN = 12,lf = 1.12,C = 1.507
4	SFLA	c = 1, le = 5
5	BBO	pmodify = 1, PMutate = 0.01, and elitism parameter = 2
6	GSA	Rpower = 2, Rnorm = 2, ElitistCheck = 1
7	GWO	–
8	AAA	
9	CS	beta = 1.5,pa = 0.25
10	TLBO	–
11	SaDE	–
12	JADE	c = 1/10; p = 0.05;
		CRm = 0.5;Fm = 0.5;
		Afactor = 1;
13	jDE	F = 0.5,CR = 0.9
14	MFLA	m = 4, beta = 0.6

$$popB = randperm(popSize) \tag{16}$$

where u_i is the row vectors containing different random permutations of the integers in the range of 1 to D, and D is the dimension. *popA* and *popB* are the two row vectors containing the different random permutations of the integers in the range of 1 to *popsize*, respectively.

$$U_i = u_j(1 : ceil(rand * D)) \tag{17}$$

U_i performs the dimension selection based on the approximate discrete uniform distribution.

*rand*D* is an uniform distribution in [0, D], *ceil* function toward the positive direction of rounding.

$$Q'_{ij} = \begin{cases} Q_{ij} + rand.(Q_{popA(u),j} - Q_{popB(v),j}), & j \in U_i \\ Q_{ij,} & j \notin U_i \end{cases} \tag{18}$$

$Q_{popA(u)}$ and $Q_{popA(v)}$ perform individual selection based on the approximate discrete uniform distribution. j is an integer in [1,D]; i, u and v are three different integers in [1,popsize], respectively. The source code of MFLA can be downloaded at https://ww2.mathworks.cn/matlabcentral/fileexchange/68233-memetic-frog-leaping-algorithm.

Pseudo code of our proposed algorithm (called MFLA) is as follows (Fig. 4).

5 Experimental results and analysis

To evaluate the performance of the proposed algorithm (MFLA), we prepare three groups of experiments as follows: (1) experiment on the continuous optimization problems in low-dimensional space, (2) experiment on the continuous optimization problems in high-dimensional space and (3) experiment on the parameter optimization for the support vector machine. In the first experiment, the well-known 13 algorithms are compared with the proposed algorithm including standard particle swarm optimization (SPSO2011) (Tang et al. 2016), differential evolution (DE/best/2/bin) (Veček et al. 2014), global best guided artificial bee colony (GbABC) (Liao et al. 2013), shuffled frog leaping algorithm (SFLA) (Eusuff and Lansey 2003), biogeography-based optimization (BBO) (Simon, 2008), gravitational search algorithm (GSA) (Rashedi et al. 2009), gray wolf optimizer (GWO) (Mirjalili et al. 2014), self-adaptive differential evolutionary algorithm (SaDE) (Tang et al. 2016), adaptive differential evolution (JADE) (Tang et al. 2016), self-adapting control parameters in differential evolution (jDE) (Tang et al. 2016), cuckoo search (CS) (Yang and Deb 2009), teaching–learning-based optimization (TLBO) (Rao et al. 2012) and artificial algae algorithm (AAA) (Uymaz et al. 2015), in which 30 benchmark problems are adopted. In the second experiment, we evaluate and compare the performance of our proposed algorithm against the well-known 13 algorithms including SPSO2011, DE/best/2/bin, GbABC, SFLA, BBO, GSA, GWO, SaDE, JADE, jDE, CS, TLBO and AAA for the four high-dimensional problems. In the third experiment, our proposed algorithm is compared against the other 4 methods including GbABC, JADE, jDE and TLBO for the parameter optimization problem of support vector machine, which is a real-world continuous optimization problem.

On the experimental datasets, 15 basic benchmark problems are recommended by the literature (Chow and

Table 3 Comparison results for parameter tuning of MFLA

Para	F4	F8	F14	F18	F23	F30
Beta = 0.3						
Means	3.6730e−004	**3.9215e+000**	1.4807e−001	**2.7949e−001**	**3.3401e+001**	8.7200e+002
Std.	2.6891e−004	**9.7476e−001**	2.7901e−002	**6.6451e−001**	**1.1789e+001**	3.7664e+001
Beta = 0.4						
Means	1.1218e−004	4.3428e+000	4.7201e−002	1.9975e−001	4.3241e+001	8.6800e+002
Std.	7.7765e−005	1.3438e+000	6.9372e−002	3.3907e−001	1.6991e+001	2.1839e+002
Beta = 0.5						
Means	5.7445e−005	4.8307e+000	8.5893e−003	4.1197e+000	4.5785e+001	8.6957e+002
Std.	3.6316e−005	1.0139e+000	3.2692e−002	1.5027e+001	2.8207e+001	2.3890e+002
Beta = 0.6						
Means	2.7956e−005	5.2892e+000	3.8705e−003	2.4518e+000	4.2251e+001	6.8081e+002
Std.	1.6633e−005	1.2182e+000	2.1199e−002	1.2350e+001	1.4547e+001	3.4792e+002
Beta = 0.7						
Means	1.5782e−005	5.5333e+000	**0.0000e+000**	4.6331e+000	5.4658e+001	5.3399e+002
Std.	9.6846e−006	1.0906e+000	**0.0000e+000**	1.6990e+001	2.1481e+001	3.6630e+002
Beta = 0.8						
Means	1.2196e−005	6.6177e+000	**0.0000e+000**	5.0885e+000	4.8244e+001	4.9890e+002
Std.	9.7593e−006	1.0796e+000	**0.0000e+000**	1.5242e+001	1.9207e+001	4.7245e+002
Beta = 0.9						
Means	9.8180e−006	7.4068e+000	**0.0000e+000**	3.9423e+000	6.5442e+001	3.3537e+002
Std.	6.7219e−006	1.3893e+000	**0.0000e+000**	1.3388e+001	4.1265e+001	3.2497e+002
Beta = 1.0						
Means	9.9590e−006	7.5406e+000	**0.0000e+000**	1.4532e+000	5.8184e+001	2.9443e+002
Std.	7.7189e−006	1.1817e+000	**0.0000e+000**	2.6333e+000	2.4020e+001	2.4557e+002
Beta = 1.1						
Means	7.4274e−006	7.8883e+000	**0.0000e+000**	3.9840e+000	5.1973e+001	3.0706e+002
Std.	5.4702e−006	1.4353e+000	**0.0000e+000**	1.3153e+001	1.8526e+001	2.8070e+002
Beta = 1.2						
Means	7.6633e−006	7.9304e+000	**0.0000e+000**	4.4238e+000	6.6156e+001	2.2764e+002
Std.	6.3173e−006	1.3338e+000	**0.0000e+000**	1.3304e+001	5.4476e+001	1.5139e+002
Beta = 1.3						
Means	5.3327e−006	9.0930e+000	**0.0000e+000**	7.1207e+000	4.7933e+001	2.4900e+002
Std.	4.8962e−006	1.1720e+000	**0.0000e+000**	2.0475e+001	1.8755e+001	1.8718e+002
Beta = 1.4						
Means	5.1073e−006	8.6706e+000	**0.0000e+000**	4.4957e+000	5.7523e+001	2.7642e+002
Std.	2.9612e−006	1.4717e+000	**0.0000e+000**	1.4036e+001	3.0614e+001	2.3328e+002
Beta = 1.5						
Means	6.5735e−006	9.3913e+000	**0.0000e+000**	3.6176e+000	5.9242e+001	2.6991e+002
Std.	5.7141e−006	1.1896e+000	**0.0000e+000**	1.3242e+001	4.1776e+001	2.6521e+002
Beta = 1.6						
Means	6.0449e−006	9.2871e+000	**0.0000e+000**	7.1445e+000	5.4422e+001	2.1863e+002
Std.	5.4165e−006	1.6746e+000	**0.0000e+000**	1.9137e+001	2.3178e+001	1.0202e+002
Beta = 1.7						
Means	6.2249e−006	9.8639e+000	**0.0000e+000**	3.9739e+000	6.2957e+001	**2.0000e+002**
Std.	5.5213e−006	1.2563e+000	**0.0000e+000**	1.3502e+001	4.3863e+001	2.4209e−013
Beta = 1.8						
Means	5.7917e−006	1.0021e+001	**0.0000e+000**	7.0927e+000	6.4582e+001	**2.0000e+002**
Std.	5.7276e−006	1.3561e+000	**0.0000e+000**	2.0173e+001	4.5877e+001	2.2725e−013

Table 3 (continued)

Para	F4	F8	F14	F18	F23	F30
Beta = 1.9						
Means	**4.9309e−006**	1.0025e+001	**0.0000e+000**	5.3278e+000	6.0527e+001	**2.0000e+002**
Std.	**3.7786e−006**	1.3181e+000	**0.0000e+000**	1.8044e+001	5.6973e+001	**2.2329e−013**

Bold fonts denote the best fitness values

Table 4 Friedman test for parameter tuning of MFLA

Order	Algorithm	Averages ranks
1	Beta = 1.2	6.92
2	Beta = 1.1	7.25
3	Beta = 1.0	8.25
4	Beta = 1.4	8.25
5	Beta = 1.3	8.42
6	Beta = 1.7	8.58
7	Beta = 1.5	8.75
8	Beta = 0.6	8.83
9	Beta = 0.4	9.00
10	Beta = 1.9	9.08
11	Beta = 0.3	9.33
12	Beta = 0.8	9.58
13	Beta = 0.9	9.58
14	Beta = 1.8	9.92
15	Beta = 0.7	9.92
16	Beta = 0.5	10.50
17	Beta = 1.6	10.83

Table 5 Friedman test for parameters tuning of MFLA

Method	Statistical value	*p* value
Friedman test	4.3074	0.9982

Yuen 2011; Huang et al. 2012), which include 4 unimodal functions and 11 multimodal functions. To evaluate the performance of the shifted, rotated and composite benchmark problems, the 15 problems in CEC2014 suites (Liang et al. 2013) are adopted. Usually, unimodal functions (F1–F4) are used to verify the local search ability and multimodal functions (F5–F30) are used to verify the global search ability. Details of the 30 benchmark problems are shown in Table 1. To better validate the effectiveness of our proposed algorithm, 13 well-known algorithms are chosen and the parameter setting is shown in Table 2.

Each benchmark problem is solved 30 times by a different initial population each time in order to avoid any negative effects of the structure of the initial population during the tests. For a fair comparison, the population size is set the same as ps = 20. There are 4 memeplexes in the population, and each memeplex has 5 frogs. In addition, the number of the fitness evaluations (FES) is used to compare all the algorithms in these tests. The max fitness evaluation (MAX_FES) is set as D*1E4 (Liang et al. 2013). The mean values, standard deviation and median values of fitness are computed as a result of the test for the detailed statistical analysis. The Wilcoxon signed-rank test (Derrac et al. 2011) with Bonferroni–Holm correction method and the Friedman's test (Derrac et al. 2011) are used to statistic the comparison results.

5.1 Parameter tuning

There are two parameters (*beta* and *m*), which influence the performance of the proposed algorithm. Parameter tuning problem can be considered as a combination optimization problem, and exhaustive all the combination of parameters is unrealistic. Therefore, we adopt the parameter tuning strategy from literatures (Lim and Isa 2014; Lam et al. 2012). As we known, parameter *beta* is the scale factor of Lévy flight. It is a real number and should be set in a region as [0.3, 1.9] according to Mantegna (1991, 1994). Parameter *m* is the number of memeplexes, and it is an integer. Only three integers (2, 4, 5) can be selected for *m* when population size is 20(population size = *m***n*, *n* is the number of frogs in a memeplex). We choose six problems (F4, F8, F14 belong to Type 1, and F18, F23, F30 belong to Type 2) from benchmark dataset in Table 1, and MFLA is ran in the same way as mentioned above. We record the mean values and standard deviations for comparison. First, we choose two parameter values randomly to initialize the parameter combination as *beta* = 0.6 and *m* = 4. Then, we fix the value of *m* (*m* = 4) and vary the value of *beta*. In Table 3, it can be observed that we obtain better performance for MFLA when beta = 0.3(F8, F18, F23) and beta = 1.9(F4, F14, F30). Table 4 shows the comparison results of MAFAL with different parameter values using Friedman test. Though Table 5 shows that there is no significant difference among the performance of MALA with different parameter values, we obtain the best *beta* value (*beta* = 1.2) (shown in Table 4). Second, we fix the better values of *beta* (0.3, 1.2, 1.9) and vary the values of *m.* Table 6 shows the comparison results of MFLA with different parameter combination(*beta* and *m*). It can be

Table 6 Comparison results for parameter tuning of MFLA

para		F4	F8	F14	F18	F23	F30
Beta = 0.3	Means	3.3474e−004	**3.8339e+000**	1.3178e−001	3.4031e+000	4.1651e+001	8.9778e+002
m = 2	Std.	2.3476e−004	**8.8503e−001**	4.3435e−002	1.7385e+001	4.5645e+001	6.6969e+001
Beta = 0.3	Means	3.6730e−004	3.9215e+000	1.4807e−001	**2.7949e−001**	**3.3401e+001**	8.7200e+002
m = 4	Std.	2.6891e−004	9.7476e−001	2.7901e−002	**6.6451e−001**	**1.1789e+001**	3.7664e+001
Beta = 0.3	Means	4.7231e−004	3.8680e+000	1.5633e−001	6.7990e+000	3.5487e+001	9.0182e+002
m = 5	Std.	2.8988e−004	7.3852e−001	2.1297e−002	2.0282e+001	1.3035e+001	5.1138e+001
Beta = 1.2	Means	7.4842e−006	1.0055e+001	**0.0000e+000**	7.6912e+000	5.9838e+001	2.6036e+002
m = 2	Std.	5.7370e−006	1.4308e+000	**0.0000e+000**	2.0575e+001	3.2360e+001	2.3031e+002
Beta = 1.2	Means	7.6633e−006	7.9304e+000	**0.0000e+000**	4.4238e+000	6.6156e+001	2.2764e+002
m = 4	Std.	6.3173e−006	1.3338e+000	**0.0000e+000**	1.3304e+001	5.4476e+001	1.5139e+002
Beta = 1.2	Means	7.4560e−006	7.5189e+000	**0.0000e+000**	5.9012e+000	5.0381e+001	**2.0000e+002**
m = 5	Std.	6.4976e−006	1.5494e+000	**0.0000e+000**	1.6953e+001	1.9333e+001	2.4574e−013
Beta = 1.9	Means	4.9845e−006	1.2499e+001	**0.0000e+000**	1.6386e+001	6.8011e+001	**2.0000e+002**
m = 2	Std.	4.5275e−006	1.1020e+000	**0.0000e+000**	2.7019e+001	3.4229e+001	2.1722e−013
Beta = 1.9	Means	**4.5347e−006**	1.0494e+001	**0.0000e+000**	1.1273e+001	5.7956e+001	**2.0000e+002**
m = 4	Std.	**3.1911e−006**	1.5070e+000	**0.0000e+000**	2.4798e+001	2.4739e+001	**2.2329e−013**
Beta = 1.9	Means	4.9309e−006	1.0025e+001	**0.0000e+000**	5.3278e+000	6.0527e+001	**2.0000e+002**
m = 5	Std.	3.7786e−006	1.3181e+000	**0.0000e+000**	1.8044e+001	5.6973e+001	**2.2329e−013**

Bold fonts denote the best fitness values

Table 7 Comparison results for parameter tuning of MFLA

Order	Algorithm	Averages ranks
1	Beta = 1.2 m = 5	3.83
2	Beta = 1.9 m = 5	4.17
3	Beta = 0.3 m = 2	4.67
3	Beta = 0.3 m = 2	4.67
3	Beta = 1.9 m = 4	4.67
4	Beta = 1.2 m = 4	5.08
5	Beta = 1.2 m = 2	5.75
6	Beta = 1.9 m = 2	6.00
7	Beta = 0.3 m = 5	6.17

Table 8 Friedman test for parameter tuning of MFLA

Method	Statistical value	p value
Friedman test	4.539	0.805

observed that these parameter combinations (*beta* = 0.3, m = 2; *beta* = 0.3 m = 4; *beta* = 1.9, m = 4) can be considered. The parameter combination of *beta* = 1.2 and m = 5 is the best according to the comparison results by the Friedman test (shown in Table 7). Though there is no significant difference among the performance of MFLA with different parameter combinations (shown in Table 8), the performance of MAFLA is better in many cases when the number of memeplex (m = 4 or 5) is approximately equal to the number of frogs (n = 5 or n = 4) in each memeplex. For parameter *beta*, we know that the worst frog can search more long distance when *beta* is larger. In contrary, the worst frog can search more short distance when *beta* is smaller. Therefore, *beta* can be set as a mean value (such as, 1.1 = (0.3 + 1.9)/2) in many cases. The *beta* can be set to a smaller value (*beta* = 0.3) or a larger value (*beta* = 1.9) for the particular problems (such as F8, F30). In general speaking, the parameter combination (*beta* = 1.2, m = 5) is suggested to be chosen for the most of the optimization problems.

5.2 Experiments on the benchmark datasets (dimension = 30)

In this section, the proposed algorithm is compared against the 13 well-known algorithms for the 30 benchmark problems as dimension = 30. The experimental results of the minimum error (the best fitness value or the optimum value) are saved as records. We compute the mean, standard deviation and the nonparametric statistics by Wilcoxon signed-rank test with Bonferroni–Holm correction and Friedman test. In Table 9, we compare MFLA to SPSO2011, DE/best/2/bin, GbABC, SFLA, BBO, GSA and GWO.

Table 9 Comparison with MFLA and other algorithms (dimension = 30)

Func	SPSO2011	DE/best/2/bin	GbABC	SFLA	BBO	GSA	GWO	MFLA
F1								
Means	**0.0000e+000**	2.8701e−074	2.7460e−016	4.3432e−020	4.5100e−001	1.4272e+004	**0.0000e+000**	**0.0000e+000**
Std	**0.0000e+000**	1.5483e−073	5.2730e−017	2.3338e−019	3.0572e−001	2.8377e+003	**0.0000e+000**	**0.0000e+000**
F2								
Means	**0.0000e+000**	3.3333e+000	2.9242e−016	2.1021e−023	**0.0000e+000**	5.6281e−009	**0.0000e+000**	**0.0000e+000**
Std	**0.0000e+000**	1.8257e+001	5.3029e−017	5.8569e−023	**0.0000e+000**	1.4021e−009	**0.0000e+000**	**0.0000e+000**
F3								
Means	1.5684e−050	4.0859e−042	1.4987e+003	3.0377e+000	9.8819e+002	4.5782e+004	2.0256e−035	**0.0000e+000**
Std	4.2478e−050	1.7054e−041	9.1510e+002	1.4583e+000	4.1004e+002	1.9172e+004	9.3086e−035	**0.0000e+000**
F4								
Means	4.5553e+001	1.8357e−002	2.3416e−002	2.0122e−003	1.7807e+000	3.9340e−002	2.1025e−004	**2.2105e−005**
Std	2.2411e+001	1.6386e−002	6.1977e−003	7.7237e−004	1.1868e+000	7.9060e−003	8.2651e−005	**1.6063e−005**
F5								
Means	4.5087e+001	7.0277e+001	**0.0000e+000**	2.3615e+001	**0.0000e+000**	3.1109e+001	1.1420e+000	**0.0000e+000**
Std	2.1266e+001	2.3651e+001	**0.0000e+000**	5.7064e+000	**0.0000e+000**	5.9061e+000	3.0974e+000	**0.0000e+000**
F6								
Means	2.1257e+000	9.9328e+000	2.8659e−014	9.7619e−003	2.3005e−001	5.7284e−005	1.0125e+001	**0.0000e+000**
Std	4.6498e−001	1.8963e+000	2.6279e−015	5.2080e−002	5.5162e−002	4.9927e−006	1.0298e+001	**0.0000e+000**
F7								
Means	8.8635e−003	4.5506e−001	**0.0000e+000**	4.7425e−002	4.7836e−001	6.0559e+002	**0.0000e+000**	**0.0000e+000**
Std	8.8035e−003	8.2830e−001	**0.0000e+000**	3.7248e−002	1.3534e−001	8.4554e+001	**0.0000e+000**	**0.0000e+000**
F8								
Means	7.0038e+000	1.8118e+001	**3.3141e−001**	3.6819e+001	7.8163e+001	7.1520e+000	2.5859e+001	4.9959e+000
Std	1.6556e+000	3.1175e+001	**8.5346e−001**	2.1688e+001	3.2330e+001	5.5899e−001	7.8261e−001	1.1763e+000
F9								
Means	6.1597e−001	4.4518e+000	2.7095e−016	3.4291e−007	3.7222e−003	1.6242e−001	5.0867e−003	**1.5705e−032**
Std	8.5917e−001	5.1683e+000	4.5486e−017	1.8461e−006	3.2115e−003	2.1562e−001	5.6084e−003	**5.5674e−048**
F10								
Means	1.0127e+001	1.2328e+001	**0.0000e+000**	4.6229e+000	**0.0000e+000**	9.6955e−002	**0.0000e+000**	**0.0000e+000**
Std	1.7835e+000	3.1011e+000	**0.0000e+000**	2.5280e+000	**0.0000e+000**	1.0225e−002	**0.0000e+000**	**0.0000e+000**
F11								
Means	3.1791e−066	2.6031e+000	1.5849e+002	3.5950e+000	9.6477e+000	5.5096e−008	3.7492e−067	**0.0000e+000**
Std	1.6280e−065	7.9454e+000	3.9217e+001	1.3982e+000	3.4704e+000	1.4900e−008	1.8912e−066	**0.0000e+000**
F12								
Means	1.4060e+000	1.6550e−011	1.8093e−008	1.2096e−004	**0.0000e+000**	2.6219e−005	5.7609e−163	**0.00e+000**
Std	1.4776e+000	4.8857e−011	9.8918e−008	2.8663e−004	**0.0000e+000**	2.7702e−006	3.1435e−162	**0.00e+000**
F13								
Means	2.1321e−001	2.8132e+000	7.0654e−001	4.0654e−001	1.1233e+000	1.2476e+001	1.5321e−001	**0.00e+000**
Std	3.4575e−002	1.2247e+000	1.0807e−001	6.3968e−002	1.8125e−001	1.4900e+000	5.0742e−002	**0.00e+000**
F14								
Means	2.0570e+000	9.1882e−001	1.0000e−001	1.1389e−001	**0.0000e+000**	1.0000e−001	5.6742e+000	**4.2755e−003**
Std	7.0822e−001	6.8943e−001	1.1643e−006	4.1471e−002	**0.0000e+000**	1.2047e−009	1.0822e+000	**2.3418e−002**
F15								
Means	6.0592e−001	1.5459e+000	5.1434e−017	4.9264e−003	**0.0000e+000**	7.3680e−008	**0.0000e+000**	**0.0000e+000**
Std	2.4653e−001	4.4250e−001	4.9059e−018	2.6982e−002	**0.0000e+000**	1.3623e−008	**0.0000e+000**	**0.0000e+000**
F16								
Means	7.6356e+003	2.0350e+008	5.3919e+001	9.7541e+007	1.1731e+006	9.2554e+010	5.8266e+008	**6.1580e−014**
Std	6.4773e+003	5.7947e+008	6.6506e+001	4.2439e+007	4.6807e+005	1.6355e+010	4.0784e+008	**1.9865e−014**

Table 9 (continued)

Func	SPSO2011	DE/best/2/bin	GbABC	SFLA	BBO	GSA	GWO	MFLA
F17								
Means	7.2116e+002	3.1798e+003	8.6012e+002	7.3001e+003	1.1958e+004	1.1202e+007	6.3166e+003	**3.4106e−013**
Std	2.5269e+002	7.0714e+003	2.6001e+003	3.4359e+003	1.1566e+004	2.0185e+007	4.5548e+003	**2.6113e−013**
F18								
Means	1.0514e+001	7.7237e+001	1.5100e+001	2.1465e+002	1.1812e+002	2.0196e+004	2.0327e+002	**4.2598e−001**
Std	2.7454e+001	3.2457e+001	2.4405e+001	2.9351e+001	3.7984e+001	5.6739e+003	8.9852e+001	**1.2578e+000**
F19								
Means	1.7497e+001	2.2305e+001	1.2767e+001	2.9343e+001	1.3911e+001	4.6647e+001	1.5320e+001	**1.1859e+001**
Std	3.1373e+000	3.1409e+000	1.5907e+000	2.3236e+000	2.7869e+000	1.8490e+000	2.7941e+000	**3.3629e+000**
F20								
Means	1.1482e−002	1.6597e+000	**1.8049e−005**	3.1543e+000	8.2042e−001	9.7299e+002	7.3026e+000	2.4655e−004
Std	1.1561e−002	4.5618e+000	**6.3638e−005**	7.3016e−001	1.0571e−001	1.5067e+002	5.7162e+000	1.3504e−003
F21								
Means	1.9111e+000	2.1876e+000	**2.3242e−001**	6.7584e−001	1.8181e−001	3.8418e+000	2.4577e+000	4.7841e−001
Std	3.5183e−001	6.3483e−001	**7.6841e−002**	2.3011e−001	4.0184e−002	5.6399e−001	2.3990e−001	8.5903e−002
F22								
Means	2.2687e−001	5.8647e−001	**2.1863e−001**	2.8342e−001	5.3249e−001	9.3019e+000	3.6680e−001	3.7483e−001
Std	5.6668e−002	1.4803e−001	**3.2112e−002**	3.7749e−002	1.3984e−001	9.3140e−001	9.3248e−002	6.4734e−002
F23								
Means	2.2821e+003	1.1973e+004	4.6654e+003	7.3246e+002	9.0527e+003	8.8189e+009	2.3635e+006	**4.6653e+001**
Std	2.6307e+003	1.0511e+004	6.3184e+003	6.9427e+002	4.4876e+003	2.8553e+009	5.3815e+006	**1.8399e+001**
F24								
Means	2.8532e+002	5.2792e+002	5.1039e+003	5.8737e+003	9.3853e+003	1.0333e+007	8.9043e+002	**5.8040e+001**
Std	8.5442e+001	8.0383e+002	2.8967e+003	1.8822e+003	6.6136e+003	2.7567e+007	1.4866e+003	**2.4531e+001**
F25								
Means	3.1525e+002	3.2237e+002	3.1574e+002	3.2165e+002	3.1594e+002	1.6719e+003	3.2522e+002	**2.00e+002**
Std	8.6067e−004	1.0693e+001	7.8335e−001	9.6218e−001	5.7088e−001	4.1548e+002	4.0370e+000	**0.00e+000**
F26								
Means	2.4127e+002	2.5639e+002	2.2076e+002	2.2551e+002	2.3162e+002	3.8090e+002	2.0001e+002	**2.0000e+002**
Std	6.3887e+000	1.0815e+001	1.3972e+001	3.2047e+000	4.5167e+000	4.8473e+001	3.6529e−003	**8.2065e−007**
F27								
Means	2.1657e+002	2.1004e+002	2.0715e+002	2.1840e+002	2.0799e+002	3.3090e+002	2.0983e+002	**2.0000e+00**
Std	3.0207e+000	6.0012e+000	1.1725e+000	4.0354e+000	2.1521e+000	5.0842e+001	2.1525e+000	**0.00e+000**
F28								
Means	1.5013e+002	1.6609e+002	**1.0032e+002**	1.0046e+002	1.0722e+002	2.8706e+002	1.0044e+002	1.9003e+002
Std	5.0753e+001	8.9550e+001	**4.0311e−002**	1.0985e−001	2.5382e+001	5.1758e+001	9.3121e−002	3.0421e+001
F29								
Means	7.1343e+002	9.2991e+002	4.0659e+002	6.9998e+002	6.9876e+002	2.2773e+003	6.9563e+002	**2.9269e+002**
Std	1.4209e+002	2.6039e+002	2.8211e+000	3.5707e+002	1.4102e+002	5.0784e+002	1.0936e+002	**1.0914e+002**
F30								
Means	1.2323e+003	1.8945e+003	8.5704e+002	4.0857e+003	1.0671e+003	9.0653e+003	1.0282e+003	**6.2981e+002**
Std	2.7882e+002	5.2140e+002	5.1347e+001	6.8771e+002	2.0477e+002	9.3441e+002	1.5372e+002	**3.2926e+002**

On the unimodal functions, Sphere function (F1), Schwefel's problem 2.22(F2), Schwefel's problem 1.2 (F3) and Quartic function (F4) are adopted for comparison. It can be observed that MFLA, GWO and SPSO2011 obtained the global optimum for Sphere function and Schwefel's problem 2.22. It means that MFLA, GWO and SPSO2011 have stronger local search ability for F1 and F2. However, it is difficult to obtain the global optimum for Schwefel's

Table 10 Comparison with MFLA and other algorithms (dimension = 30)

Func	AAA	CS	TLBO	SaDE	JADE	jDE	MFLA
F1							
Means	6.8985e−169	2.8674e−038	**0.0000e+000**	1.1766e−101	1.2598e−222	1.1162e−209	**0.0000e+000**
Std	**0.0000e+000**	6.5675e−038	**0.0000e+000**	6.4432e−101	**0.0000e+000**	**0.0000e+000**	**0.0000e+000**
F2							
Means	2.3886e−170	3.9956e−039	**0.0000e+000**	2.0779e−103	4.1123e−220	4.9837e−207	**0.0000e+000**
Std	**0.0000e+000**	7.4256e−039	**0.0000e+000**	1.1170e−102	**0.0000e+000**	**0.0000e+000**	**0.0000e+000**
F3							
Means	1.3355e−002	2.8718e−005	**0.0000e+000**	2.4352e−002	2.2678e−055	7.3716e−012	**0.0000e+000**
Std	1.1644e−002	3.5276e−005	**0.0000e+000**	8.8827e−002	1.0334e−054	2.0936e−011	**0.0000e+000**
F4							
Means	8.5641e−003	9.3236e−003	7.2848e−005	8.8467e−003	2.9540e−003	7.7073e−003	**2.2105e−005**
Std	2.3630e−003	5.0437e−003	2.9388e−005	4.6034e−003	1.6924e−003	9.0739e−003	**1.6063e−005**
F5							
Means	3.5487e+000	1.1226e+001	1.3965e+001	3.9467e+000	1.9899e−001	1.0613e+000	**0.0000e+000**
Std	2.3902e+000	3.7571e+000	7.0856e+000	1.6209e+000	4.0479e−001	1.2783e+000	**0.0000e+000**
F6							
Means	1.4566e−014	3.1043e−002	4.0264e−015	1.7970e+000	5.8322e−001	7.4129e−001	**0.0000e+000**
Std	3.1432e−015	1.7003e−001	1.2283e−015	5.7557e−001	7.9550e−001	1.4254e+000	**0.0000e+000**
F7							
Means	**0.0000e+000**	1.2324e−003	**0.0000e+000**	8.5658e−002	5.9041e−003	1.2170e−002	**0.0000e+000**
Std	**0.0000e+000**	3.2844e−003	**0.0000e+000**	1.7779e−001	1.0159e−002	1.9348e−002	**0.0000e+000**
F8							
Means	3.2218e+000	3.7629e+000	1.7363e+001	2.2823e+001	**1.5943e+000**	4.1072e+000	4.9959e+000
Std	5.4762e+000	1.8908e+000	**8.4136e−001**	1.4933e+001	4.5133e+000	5.1192e+000	1.1763e+000
F9							
Means	**1.5705e−032**	5.1095e−030	3.9352e−030	8.9871e−002	1.0725e−001	5.8771e−002	**1.5705e−032**
Std	**5.5674e−048**	1.8687e−029	1.2239e−029	1.5344e−001	2.9190e−001	1.5332e−001	**5.5674e−048**
F10							
Means	**0.0000e+000**	1.8513e−001	**0.0000e+000**	7.6034e−001	6.9189e−001	1.5328e−001	**0.0000e+000**
Std	**0.0000e+000**	3.1651e−001	**0.0000e+000**	5.6642e−001	8.1363e−001	2.5964e−001	**0.0000e+000**
F11							
Means	5.5718e−004	6.6002e−007	5.5342e−214	1.9606e−003	2.8179e−009	8.8999e−020	**0.0000e+000**
Std	5.8238e−004	6.1023e−007	**0.0000e+000**	5.2562e−003	1.5435e−008	1.7576e−019	**0.0000e+000**
F12							
Means	4.1115e−015	5.9142e−001	**0.0000e+000**	4.7751e−016	3.7192e−016	2.5234e−016	**0.0000e+000**
Std	2.9802e−015	5.3010e−001	**0.0000e+000**	6.7927e−016	6.1714e−016	7.4449e−016	**0.0000e+000**
F13							
Means	6.3987e−001	3.1990e−001	9.9873e−002	5.9654e−001	4.8654e−001	4.0654e−001	**0.0000e+000**
Std	1.4039e−001	6.6430e−002	6.6665e−011	1.6914e−001	1.3060e−001	2.3916e−001	**0.0000e+000**
F14							
Means	1.0000e−001	1.4849e−001	2.0679e−001	1.0000e−001	1.0000e−001	1.0000e−001	**4.2755e−003**
Std	1.0811e−0166	1.1596e−002	4.6927e−001	7.0575e−017	1.0286e−007	1.3934e−015	**2.3418e−002**
F15							
Means	5.6740e−173	9.8523e−003	**0.0000e+000**	1.0838e−001	1.9705e−001	5.9114e−002	**0.0000e+000**
Std	**0.0000e+000**	3.7494e−002	**0.0000e+000**	1.2232e−001	1.6158e−001	9.1841e−002	**0.0000e+000**
F16							
Means	1.4532e+003	**1.0421e−014**	2.6528e−002	7.7794e+003	3.4153e−012	1.3358e−013	6.1580e−014
Std	2.3385e+003	2.2985e−014	1.4060e−001	6.2096e+003	1.8170e−011	1.9890e−013	**1.9865e−014**

Table 10 (continued)

Func	AAA	CS	TLBO	SaDE	JADE	jDE	MFLA
F17							
Means	2.7031e+003	1.1833e−010	3.9461e+000	1.9235e+003	3.1878e−011	1.0768e−011	**3.4106e−013**
Std	4.2522e+003	2.9500e−010	1.1407e+001	2.1475e+003	8.0224e−011	4.0099e−011	**2.6113e−013**
F18							
Means	2.2671e+001	1.4393e+001	6.9300e+001	4.3740e+001	4.3881e+000	1.2332e+001	**4.2598e−001**
Std	3.5987e+001	2.7722e+001	2.5717e+001	4.1463e+001	1.6062e+001	2.5079e+001	**1.2578e+000**
F19							
Means	1.4786e+001	2.3317e+001	2.2002e+001	1.1920e+001	**7.9167e+000**	8.8387e+000	1.1859e+001
Std	**1.6181e+000**	3.1472e+000	3.0434e+000	2.2695e+000	2.3186e+000	3.0460e+000	3.3629e+000
F20							
Means	9.8581e−004	3.8666e−003	5.5683e−001	3.9148e−002	1.0563e−002	4.9079e−002	**2.4655e−004**
Std	3.0767e−003	6.2021e−003	2.4214e+000	6.4627e−002	1.4596e−002	8.7936e−002	**1.3504e−003**
F21							
Means	1.6684e−001	6.8326e−001	2.4712e+000	6.2450e−001	**1.1328e−001**	2.4374e−001	4.7841e−001
Std	1.1275e−001	1.2944e−001	2.5779e−001	8.0847e−002	**1.8821e−002**	3.7638e−002	8.5903e−002
F22							
Means	3.8732e−001	3.5674e−001	4.8771e−001	3.0766e−001	3.0311e−001	**2.6083e−001**	3.7483e−001
Std	1.1529e−001	5.6859e−002	1.0793e−001	7.1721e−002	7.3082e−002	**4.1616e−002**	6.4734e−002
F23							
Means	4.4482e+003	1.0826e+002	1.9591e+003	1.5423e+003	1.3439e+003	1.5422e+003	**4.6653e+001**
Std	3.7493e+003	2.2886e+001	2.2286e+003	3.0345e+003	4.6543e+003	2.9300e+003	**1.8399e+001**
F24							
Means	9.9249e+003	7.3042e+001	3.0221e+002	4.7442e+003	3.3960e+003	**5.2829e+001**	5.8040e+001
Std	7.0083e+003	1.9188e+001	1.1428e+002	3.8276e+003	4.6957e+003	3.1161e+001	**2.4531e+001**
F25							
Means	3.1524e+002	3.1524e+002	3.1525e+002	3.1524e+002	3.1524e+002	3.1524e+002	**2.0000e+002**
Std	1.0722e−007	5.7815e−014	9.4009e−003	1.5443e−005	1.3830e−012	1.0462e−012	**0.00e+000**
F26							
Means	2.2791e+002	2.2505e+002	**2.0000e+002**	2.3648e+002	2.3883e+002	2.3605e+002	**2.0000e+002**
Std	3.7658e+000	1.3770e+000	4.5431e−004	6.9769e+000	5.9235e+000	7.2387e+000	**8.2065e−007**
F27							
Means	2.0591e+002	2.0439e+002	2.0000e+002	2.1262e+002	2.0755e+002	2.0803e+002	**2.0000e+002**
Std	1.9721e+000	9.4831e−001	1.8882e−013	2.4769e+000	4.0129e+000	4.6792e+000	**0.00e+000**
F28							
Means	1.0040e+002	1.0036e+002	1.4694e+002	1.4022e+002	1.2030e+002	**1.0031e+002**	1.9003e+002
Std	**9.2162e−002**	6.9330e−002	5.0494e+001	4.9681e+001	4.0542e+001	1.0239e−001	3.0421e+001
F29							
Means	4.4751e+002	4.0454e+002	7.0491e+002	5.4629e+002	4.9371e+002	4.5369e+002	**2.9269e+002**
Std	1.1843e+002	**2.4065e+000**	2.6471e+002	1.2740e+002	9.4038e+001	7.4796e+001	1.0914e+002
F30							
Means	9.1209e+002	9.8627e+002	1.5567e+003	9.9204e+002	9.2917e+002	8.8268e+002	**6.2981e+002**
Std	**5.1179e+001**	9.4492e+001	3.8900e+002	6.4839e+001	1.4847e+002	3.2631e+001	3.2926e+002

problem 1.2 (F3) and Quartic function (F4). It can be observed that only MFLA obtains the global optimum for Schwefel's problem 1.2 (F3). For Quartic function (F4), MFLA obtains a better accuracy than SPSO2011, DE/best/2/bin, GbABC, SFLA, BBO, GSA and GWO. We can see that MFLA has stronger local search ability than other 7 algorithms. In Table 10, we compare MFLA to the AAA, CS, TLBO, SaDE, JADE and jDE for the four unimodal functions (F1–F4). It can be observed that MFLA and TLBO obtain the global optimum for Sphere function (F1),

Table 11 Comparison with MFLA and other algorithms (means)

	SPSO	TLBO	JADE	jDE	SaDE	AAA	GWO	CS	GSA	SFLA	BBO	GbABC	DE
F1	=	=	+	+	+	+	=	+	+	+	+	+	+
F2	=	=	+	+	+	+	=	+	+	+	=	+	+
F3	+	=	+	+	+	+	+	+	+	+	+	+	+
F4	+	+	+	+	+	+	+	+	+	+	+	+	+
F5	+	+	+	+	+	+	+	+	+	+	=	=	+
F6	+	+	+	+	+	+	+	+	+	+	+	+	+
F7	+	=	+	+	+	=	=	+	+	+	+	=	+
F8	+	+	−	−	+	−	+	+	+	+	+	−	+
F9	+	+	+	+	+	=	+	+	+	+	+	+	+
F10	+	=	+	+	+	=	=	+	+	+	=	=	+
F11	+	+	+	+	+	+	+	+	+	+	+	+	+
F12	+	=	+	+	+	+	+	+	+	+	=	+	+
F13	+	+	+	+	+	+	+	+	+	+	+	+	+
F14	+	+	+	+	+	+	+	+	+	+	=	+	+
F15	+	=	+	+	+	+	=	+	+	+	=	+	+
F16	+	+	+	+	+	+	+	−	+	+	+	+	+
F17	+	+	+	+	+	+	+	+	+	+	+	+	+
F18	+	+	+	+	+	+	+	+	+	+	+	+	+
F19	+	+	−	−	+	+	+	+	+	+	+	+	+
F20	+	+	+	+	+	+	+	+	+	+	+	+	+
F21	+	+	−	−	+	−	+	+	+	+	−	−	+
F22	−	+	−	−	−	+	−	−	+	−	+	−	+
F23	+	+	+	+	+	+	+	+	+	+	+	+	+
F24	+	+	+	−	+	+	+	+	+	+	+	+	+
F25	+	+	+	+	+	+	+	+	+	+	+	+	+
F26	+	=	+	+	+	+	=	+	+	+	+	+	+
F27	+	=	+	+	+	+	+	+	+	+	+	+	+
F28	−	−	−	−	−	−	−	−	+	−	−	−	−
F29	+	+	+	+	+	+	+	+	+	+	+	+	+
F30	+	+	+	+	+	+	+	+	+	+	+	+	+
w/e/l	**26/2/2**	**20/9/1**	**25/0/5**	**24/0/6**	**28/0/2**	**24/3/3**	**23/5/2**	**27/0/3**	**30/0/0**	**28/0/2**	**22/6/2**	**23/3/4**	**29/0/1**

Schwefel's problem 2.22(F2) and Schwefel's problem 1.2 (F3). It means that they have the same local search ability. For Quartic function (F4), TLBO also obtains better accuracy than CS, AAA, SaDE, JADE and jDE except for MFLA. The mean and standard deviation show that MFLA performs better than TLBO.

On the multimodal functions, 26 optimization problems are selected for comparison. Table 9 shows that MFLA obtains smaller error than SPSO2011, DE/best/2/bin, GbABC, SFLA, BBO, GSA and GWO for most multimodal optimization problems. Ackley function (F6) has many local optima, on which all methods cannot obtain the global optimum except for MFLA. The mean and standard deviation values show that MFLA obtains the global optimum. On Zakharov function (F11), only MFLA obtains the global optimum. GWO obtains smaller errors than SPSO2011, DE/best/2/bin, GbABC, SFLA, BBO, GSA. For Rotated discus function (F17), MFLA obtains smaller errors than SPSO2011, DE/best/2/bin, GbABC, SFLA, BBO, GSA and GWO. And the second best is GbABC. For Composition function 5 ($N = 5$) (F29), it is a complex multimodal optimization problem. The mean and standard deviation values show that MFLA obtains smaller error than SPSO2011, DE/best/2/bin, GbABC, SFLA, BBO, GSA and GWO. In Table 10, MFLA also obtains better solutions for many multimodal optimization problems. For Salomon problem (F13), only MFLA obtains the global optimum. The second best is GWO, and the third is the SPSO2011. For Penalized function (F9), there is no algorithm who obtains the global optimum. However, MFLA

Table 12 Friedman test for MFLA and other art-state algorithms

Order	Algorithm	Averages ranks
1	**MFLA**	3.07
2	jDE	6.37
3	JADE	6.58
4	TLBO	6.67
5	AAA	6.93
6	GbABC	7.08
7	CS	7.23
8	GWO	8.53
9	SaDE	9.32
10	BBO	9.40
11	SPSO2011	9.50
12	SFLA	10.55
13	DE	12.37
14	GSA	13.18

Table 13 Statistical value of Friedman test for MFLA and other art-state algorithms

Method	Statistical value	p value
Friedman test	175.104	6.35E−30

achieves a smaller error than CS, TLBO, SaDE, JADE and jDE. For Shifted and Rotated Griewank's function (F20), MFLA obtains a smaller error than AAA, CS, TBLO, SaDE, JADE and jDE. The second best is AAA, and the third is CS. For Hybrid function 2 ($N = 3$) (F23), MFLA also obtains smaller error than AAA, CS, TBLO, SaDE, JADE and jDE. The results demonstrate that MFLA has stronger global search ability.

In addition, F1–F15 are the basic optimization problems (Type 1) and F16–F30 are the Shifted or Rotated optimization problems (Type 2). They can be considered as different types. Some algorithms can only suit to Type 1 or Type 2. For instance, in Tables 9 and 10, SPSO2011, GWO, TLBO suit to solve Type 1 problems, whereas SaDE, JADE and jDE suite to solve Type 2 problems. MFLA suites to solve both Type 1 and Type 2. It means that MFLA is able to solve more kinds of optimization problems.

In Tables 9, 10 and 11, the winner is highlighted in bold. In Table 11, we statistic the results for all comparisons. The '+' means the win of MFLA, '-' means the loss of MFLA, and '=' means that another algorithm has the equal minimum errors with MFLA. The 'w/e/l' denotes 'win/equal/loss.'

The results show that MFLA achieves outstanding performance. Tables 12 and 13 include the comparison results of Friedman test. The average rank of MFLA is 3.07, that is smaller than others. It means that the performance of MFLA is better than other algorithms. Table 14 shows the statistic results of the Wilcoxon signed-rank test with Bonferroni–Holm correction. In this method, R+ is the sum of ranks for the problems in which the first algorithm outperforms the second, and R- is the sum of ranks in the opposite. The p-values are computed according to the R+ and R- values. Bonferroni–Holm correction is based on the correction of p value. The null hypothesis H_0 was used for the Wilcoxon signed-rank tests for purposes in this paper. The statistical significance value used to test H_0 hypothesis is $p = 0.05$. If a p value that is smaller than or equal to p occurs in any test, then the H_0 hypothesis is rejected and the alternative hypothesis is selected. In Table 14, H_0 of the comparison between MFLA and other algorithms is

Table 14 Wilcoxon signed-rank test and Bonferroni–Holm correction (means values)

i	Algorithm	R+	R−	p value	a/i	$w/t/l$	Hypothesis
12	MFLA- GSA	0.00	15.50	1.73E−6	0.0042	30/0/0	REJECT
11	MFLA-DE	20.00	15.34	1.23E−5	0.0045	29/0/1	REJECT
10	MFLA- SFLA	15.50	15.50	3.40E−5	0.0050	28/0/2	REJECT
9	MFLA-SaDE	16.00	15.46	3.72E−5	0.0056	28/0/2	REJECT
8	MFLA-SPSO2011	11.50	14.73	4.15E−5	0.0063	26/2/2	REJECT
7	MFLA-GWO	11.50	13.13	1.74E−4	0.0071	23/2/5	REJECT
7	MFLA-BBO	7.67	13.73	1.74E−4	0.0071	22/3/5	REJECT
6	MFLA- CS	15.50	15.50	4.53E−4	0.0083	26/0/4	REJECT
5	MFLA-TLBO	15.00	10.80	4.77E−4	0.010	20/9/1	REJECT
4	MFLA-AAA	15.33	13.83	5.91E−4	0.0125	24/3/3	REJECT
3	MFLA-GbABC	12.20	14.41	0.0021	0.0167	22/3/5	REJECT
2	MFLA-JADE	18.60	14.88	0.0041	0.025	25/5/0	REJECT
1	MFLA-jDE	19.67	14.46	0.0185	0.05	24/0/6	REJECT

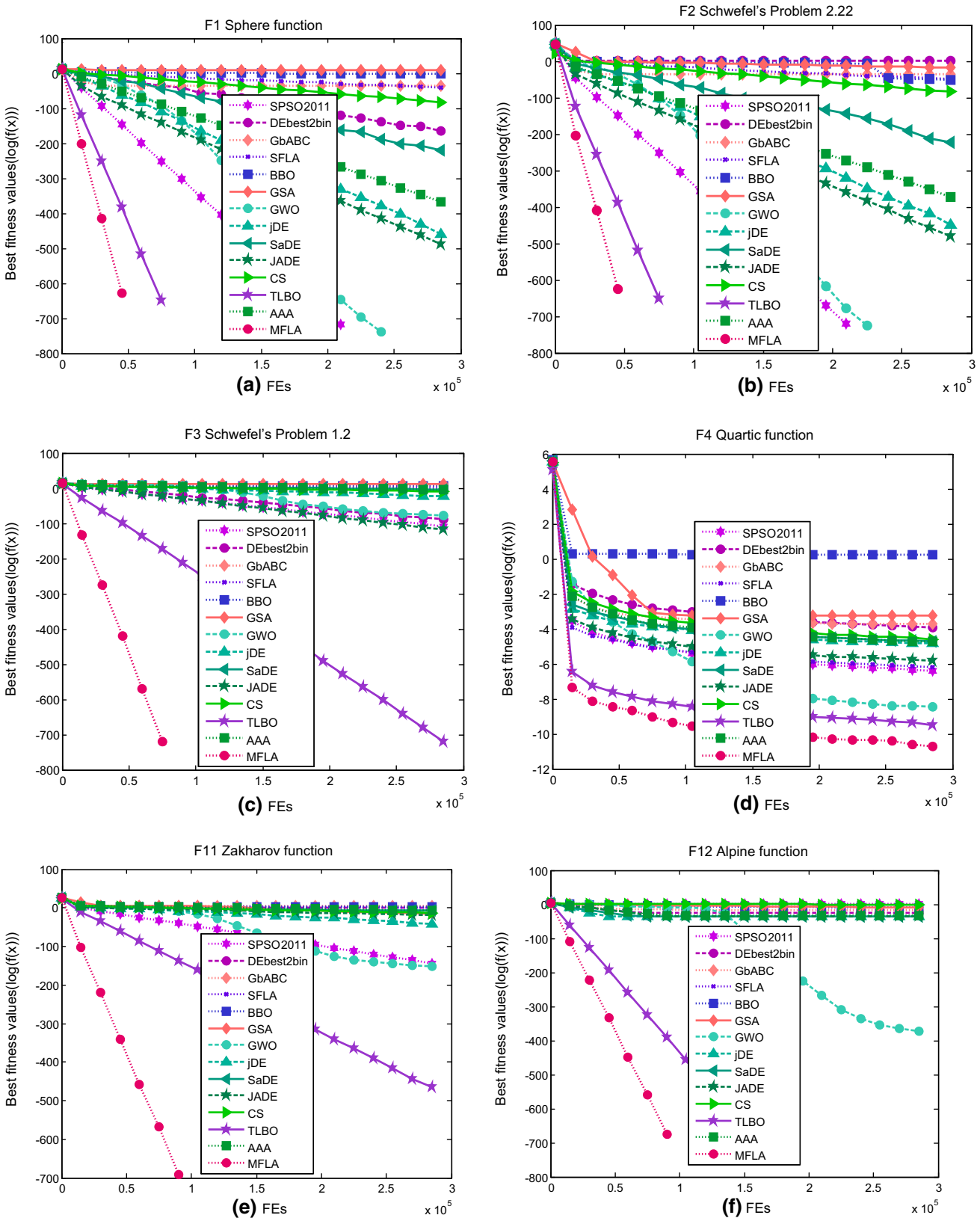

Fig. 5 Convergence curve of 14 algorithms

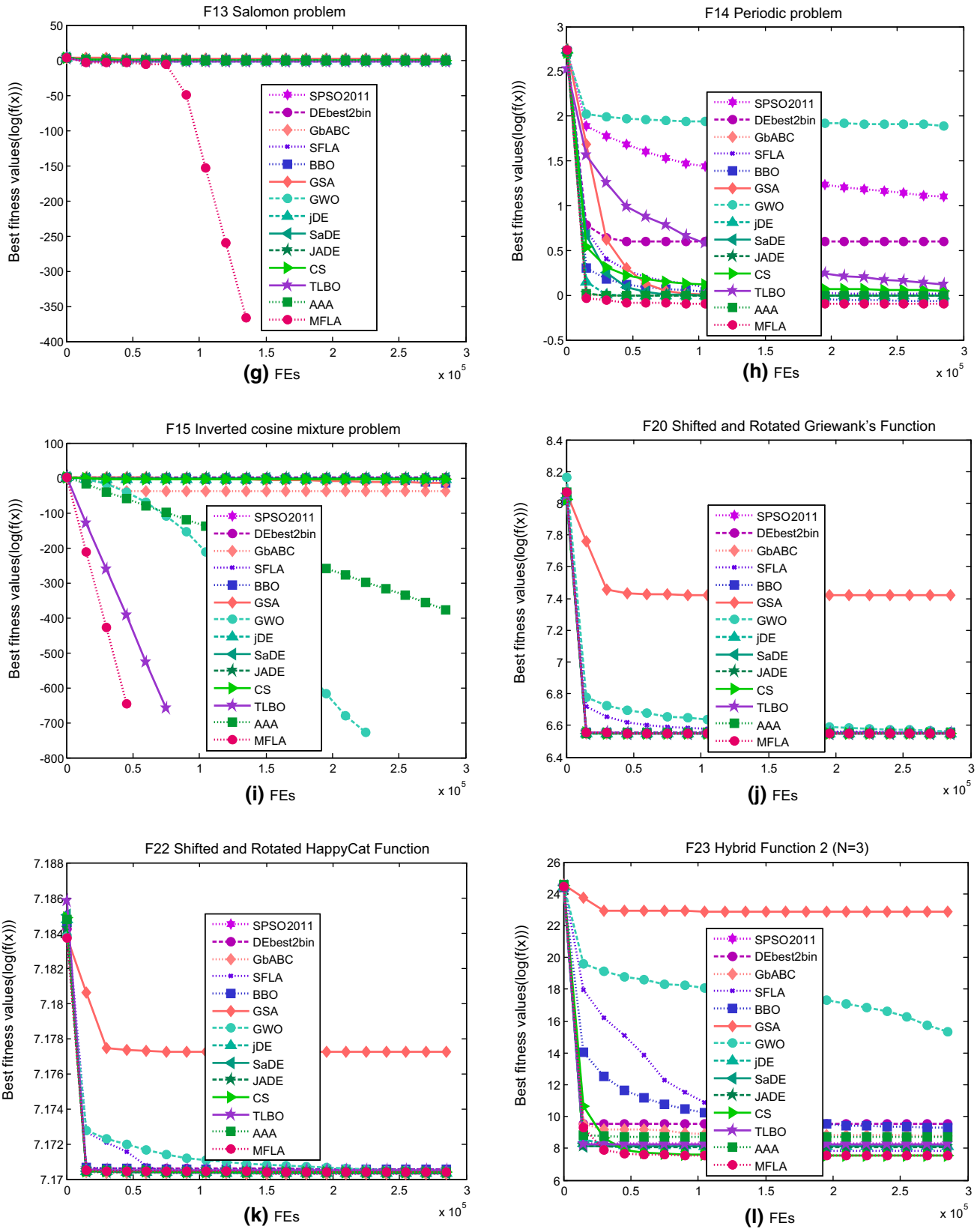

Fig. 5 continued

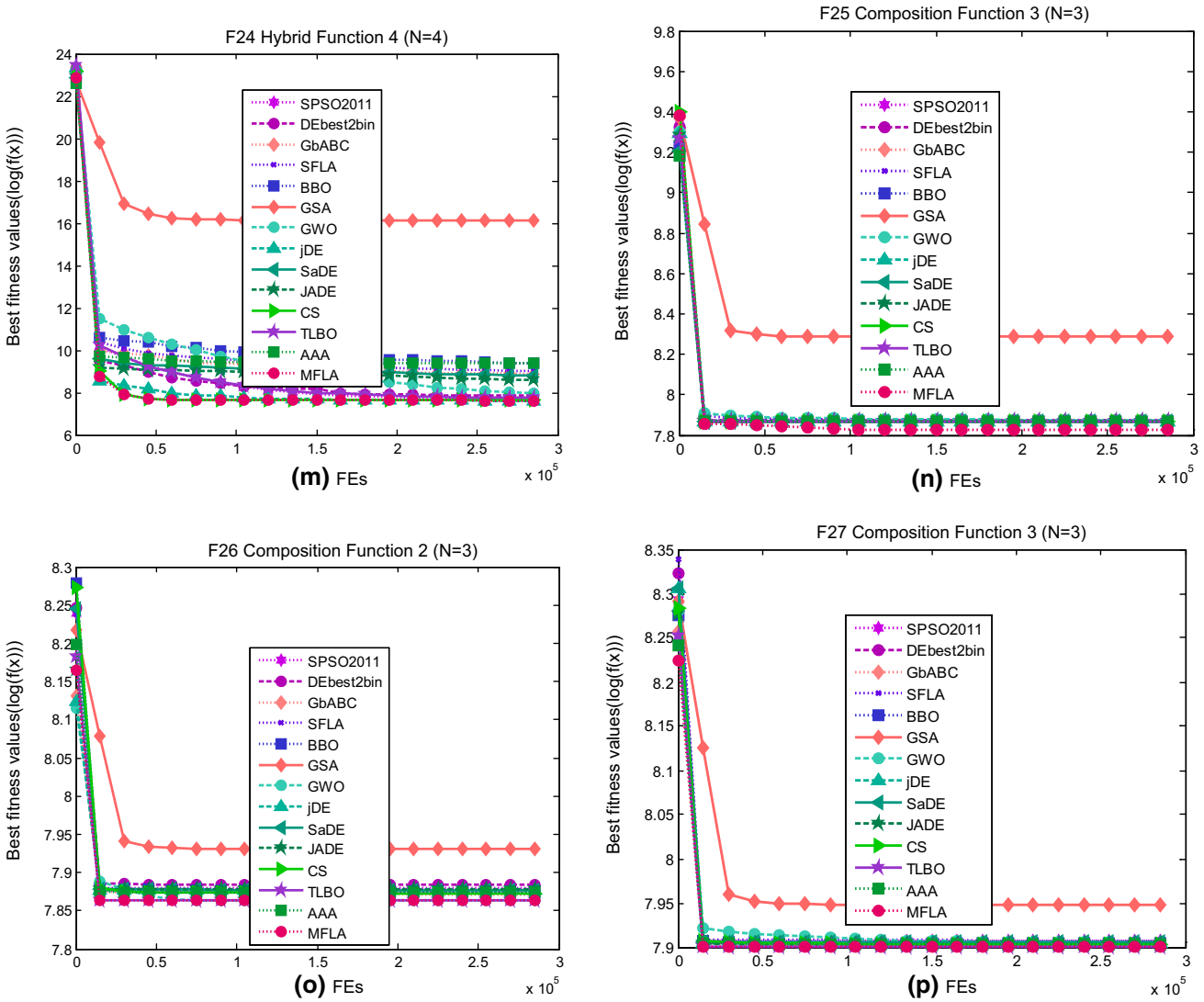

Fig. 5 continued

rejected. It means that the performance of MFLA is significantly better than that of other algorithms.

Figure 5 shows the convergence curve of the 14 algorithms including SPSO2011, DE/best/2/bin, SaDE, JADE, jDE, CS, TLBO, GbABC, SFLA, BBO, GSA, GWO, AAA and MFLA for the unimodal and multimodal optimization problems. We can see that our proposed algorithm obtains better solution and faster convergence speed. On unimodal optimization problems, the convergent cures for Sphere function (F1), Schwefel's problem 2.22 (F2), Schwefel's problem 1.2 (F3) and Quartic function (F4) can be shown in Fig. 5. It shows that MFLA converges faster than all other 13 algorithms. TLBO has better convergent speed than SPSO2011, DE/best/2/bin, GbABC, SFLA, BBO, GSA, GWO, SaDE, JADE, jDE, CS, AAA. On the multimodal optimization problems, MFLA also obtains better accuracy and convergent speed. It can be observed in Fig. 5e, f that MFLA obtains the global optimum when the search rate (FES/MAX_FES $<$ 100,000/300,000) is less than 0.5 for Zakharov function (F11) and Alpine function (F12). For Salomon problem (F13), Fig. 5g shows that MFLA obtains the global optimum when the search rate (FES/MAX_FES $<$ 150,000/300,000) is less than 0.5. It suggests that MFLA obtains the global optimum in the earlier search stage for Zakharov function (F11), Alpine function (F12) and Salomon problem (F13). For the shifted or rotated multimodal problems, MFLA also has fast convergence. For Hybrid function 4 ($N = 4$) (F24), it can be observed that GSA has the slower convergent speed. In the earlier search stage, jDE obtains the better fitness values than MFLA. However, MFLA exceeds it in later.

5.3 Experiments on the benchmark datasets (dimension = 100)

In this section, we choose four benchmark problems (F13, F14, F20 and F25) including two shifted and rotated problems from Table 1 to evaluate test the performance of our proposed algorithm for high-dimensional data (dimension = 100). With the increase in dimensionality, the performance of some algorithms may decrease significantly, especially for multimodal optimization problems. When the dimension is increased, more local optima will appear. Table 15 shows that MFLA obtains smaller errors than all others except for F20. For the basic multimodal optimization problems of Salomon problem (F13) and Periodic problem (F14), MFLA can also obtain the global optimum when the dimension is increased from 30 to 100. It suggests that MFLA has better scalability than other 13 algorithms for basic optimization problems (Type 1). For Shifted and Rotated Griewank's function (F20), MFLA obtains better accuracy than other algorithms except for AAA. For Composition function 1 ($N = 5$) (F25), MFLA obtains stable fitness values when the dimension is increased from 30 to 100. Table 16 shows the comparison results. '+' means the win, and '−' means the loss. It is obviously that our proposed algorithm is better than all the other algorithms except AAA for F20. It means that AAA is more suitable to solve optimization problem of Shifted and Rotated Griewank's function (F20) in high-dimensional space ($D = 100$) using the spiral search style. It also confirms that no heuristic algorithm can solve all the optimization problems according to the NFL theorems (Wolpert and Macready 1997). Tables 17 and 18 show the static results by Friedman test. The average rank of MFLA (1.25) is smaller than that of other 13 algorithms. In addition, Fig. 6 shows the best fitness values on the 30 times running in box figure. The English letters A, B, C, D, E, F, G, H, I, J, K, L, M and Q) denote the 14 algorithms AAA, BBO, CS, GWO, TLBO, SFLA, SPSO2011, GSA, GbABC, DEbest2bin, SaDE, JADE, jDE and MFLA, respectively. For Salomon problem (F13), GSA (H), DEbest2bin (J), SaDE (K), JADE (L) and jDE (M) obtain more different fitness values on 30 times running. For Periodic problem (F14), GWO (D), SPSO2011 (G) and DEbest2bin (J) obtain more different fitness values on 30 times running. For Shifted and Rotated Griewank's function and Composition function 1 ($N = 5$) (F25), the performance of GSA (H) is worst with more different fitness values on 30 times running. We can see that our proposed algorithm (MFLA) can obtain more stable accuracy.

5.4 Experiment on the parameter optimization of SVM

Vapnik (1995) proposed a support vector machine (SVM) learning method based on the VC (Vapnik–Chervonenkis) statistical learning theory and the structural risk minimization principle. It has the advantage for solving pattern recognition problems with small samples in nonlinear and high dimensional space, and has been applied in many fields such as novelty detection (Zhou and Yang 2006), protein subcellular localization classification (Bo et al. 2005), fault diagnosis (Shin et al. 2005) and so on. In SVM, the input vectors should be transformed from low-dimensional space to high-dimensional space using some nonlinear transformation method. Then, a hyper plane is constructed in high-dimensional space to classify the two different data. SVM can be seen as a quadratic optimization problem:

$$\begin{aligned} &\min \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}\xi_i \\ &\text{s.t.} \quad y_i(w\cdot\phi(x_i)) + b) + \xi_i \geq 1 (\xi_i \geq 0, i = 1, 2, \ldots, l) \end{aligned} \tag{19}$$

$x_i \in x \subset R^n$ is the input vector, $y_i = \{-1, +1\}$ is the class number, and C is the penalty parameter which imposes a trade-off between training error and generalization, ξ_i is a slack variable, and w is an orthogonal vector to the hyper plane.

The optimal problem of Eq. (19) can be converted to the dual problem as follows by Lagrange method:

$$\begin{aligned} &\min \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(x_i \cdot x_j) - \sum_{j=1}^{l}\alpha_j \\ &\text{s.t.} \quad \sum_{i=1}^{l} y_i \alpha_i = 0 \, (0 \leq \alpha_i \leq C, i = 1, 2, \ldots, l) \end{aligned} \tag{20}$$

$K(\cdot, \cdot)$ denotes the kernel function. The generally used kernel function is given as follows:

RBF kernel $K(x_i, x_j) = \exp(-||x_i - x_j||^2/2\sigma^2)$; σ is parameter. A large experiments have confirmed that the setting of parameters (C, σ) directly influences the performance of SVM.

Support vector machine-based disease diagnosis approaches have attracted considerable research interest in recent years. People can know if he or she has some disease by these methods according to the obtained dataset of physical examination. These problems can be considered as a classification problem, and the accuracy of prediction is important. However, SVM is sensitive to the two parameters (C, σ). It is a black box optimization problem and can be solved by heuristic algorithms. In this paper, we attempt to increase the accuracy of prediction of SVM for the

Table 15 Comparison with MFLA and other algorithms (dimension = 100)

Func	F13	F14	F20	F25
AAA				
Median	1.9119e+000	1.0000e−001	**2.7285e−012**	3.4834e+002
Means	1.9838e+000	1.0000e−001	**2.8043e−012**	3.4845e+002
Std	2.9278e−001	2.1213e−016	**7.6894e−013**	3.0986e−001
BBO				
Median	3.5999e+000	1.0113e−001	1.1317e+000	3.5083e+002
Means	3.6269e+000	1.0104e−001	1.1313e+000	3.5086e+002
Std	3.7431e−001	3.8098e−004	3.3096e−002	1.4842e+000
CS				
Median	1.7999e+000	1.0673e+000	8.4015e−011	3.4823e+002
Means	1.8432e+000	9.6783e−001	9.1739e−003	3.4823e+002
Std	5.3090e−001	3.6153e−001	2.2861e−002	1.0202e−012
GWO				
Median	1.9987e−001	2.8293e+001	2.5699e+002	5.0371e+002
Means	2.1987e−001	2.8188e+001	2.7602e+002	5.0985e+002
Std	4.0684e−002	7.7853e−001	8.9874e+001	3.3286e+001
TLBO				
Median	9.9873e−002	1.3201e−001	8.5117e−002	**2.0000e+002**
Means	1.2654e−001	6.4957e−001	2.4542e−001	2.1482e+002
Std	4.4978e−002	1.2018e+000	4.2915e−001	4.5233e+001
SFLA				
Median	1.1999e+000	2.4663e−001	6.9146e+001	4.0530e+002
Means	1.2299e+000	5.2767e−001	6.7739e+001	3.9173e+002
Std	8.7691e−002	6.3107e−001	6.2390e+000	5.2233e+001
SPSO2011				
Median	6.9987e−001	1.3083e+001	3.6380e−012	3.4899e+002
Means	6.7654e−001	1.2665e+001	4.4351e−003	3.4905e+002
Std	8.1720e−002	2.2888e+000	7.8549e−003	2.2915e−001
GSA				
Median	2.7700e+001	1.0000e−001	3.8788e+003	4.6176e+003
Means	2.7623e+001	1.0000e−001	3.8785e+003	4.5644e+003
Std	1.1133e+000	6.3523e−009	3.3107e+002	7.1314e+002
GbABC				
Median	2.8999e+000	1.0000e−001	3.1069e−004	3.6059e+002
Means	2.8332e+000	1.0000e−001	5.4291e−003	3.6002e+002
Std	2.1867e−001	9.7954e−008	2.1194e−002	8.2171e+000
DEbest2bin				
Median	1.6600e+001	3.6515e+000	2.5113e+001	3.8003e+002
Means	1.6240e+001	3.7003e+000	3.7019e+001	4.1087e+002
Std	2.5849e+000	1.2283e+000	4.4641e+001	8.9692e+001
SaDE				
Median	3.5999e+000	1.0000e−001	1.9678e−002	3.4840e+002
Means	3.6632e+000	1.0000e−001	3.5765e−001	3.4847e+002
Std	6.5784e−001	7.0575e−017	9.5469e−001	2.7113e−001
JADE				
Median	5.4999e+000	1.0000e−001	3.9449e−011	3.4823e+002
Means	5.3765e+000	1.0000e−001	1.4168e−002	3.4823e+002
Std	1.8547e+000	5.7886e−006	3.3184e−002	7.3113e−010
jDE				
Median	5.1999e+000	1.0000e−001	7.3960e−003	3.4823e+002

Table 15 (continued)

Func	F13	F14	F20	F25
Means	4.9399e+000	1.0000e−001	1.3319e−001	3.4823e+002
Std	1.4658e+000	1.0819e−014	3.4184e−001	5.8993e−010
MFLA				
Median	**0.0000e+000**	**0.0000e+000**	3.5243e−012	**2.0000e+002**
Means	**0.0000e+000**	**0.0000e+000**	2.4653e−004	**2.0000e+002**
Std	**0.0000e+000**	**0.0000e+000**	1.3503e−003	**0.0000e+000**

Bold fonts denote the best fitness values

Table 16 Static results (dimension = 100)

	F13	F14	F20	F25
AAA	+	+	−	+
BBO	+	+	+	+
CS	+	+	+	+
GWO	+	+	+	+
TLBO	+	+	+	+
SFLA	+	+	+	+
SPSO2011	+	+	+	+
GSA	+	+	+	+
GbABC	+	+	+	+
DEbest2bin	+	+	+	+
SaDE	+	+	+	+
JADE	+	+	+	+
jDE	+	+	+	+

Table 17 Friedman test for MFLA and other art-state algorithms (100D)

Order	Algorithm	Averages ranks
1	**MFLA**	**1.25**
2	AAA	4.63
3	TLBO	5.50
4	CS	6.50
5	jDE	6.63
5	JADE	6.63
5	GbABC	6.63
6	SPSO2011	7.00
7	SaDE	7.63
8	BBO	9.00
9	SFLA	9.25
10	GWO	10.75
11	GSA	11.63
12	DE	12.00

Bold font denote the winner with the minimal averages ranks

disease diagnosis of breast cancer and heart disease through optimizing the parameter values by our proposed algorithm.

Table 18 Statistical value of Friedman test for MFLA and other art-state algorithms (100D)

Method	Statistical value	p value
Friedman test	25.234	0.022

In this section, four benchmark datasets with different dimensions are chosen from the repository of the machine learning databases (Merz and Blake 2015), that is, prognostic breast cancer Wisconsin dataset (PBCW), original breast cancer Wisconsin dataset (OBCW), heart disease dataset and Wisconsin diagnostic breast cancer dataset (WDBC).

In order to evaluate the performance of MFLA, four state-of-the-art algorithms including JADE, jDE, GbABC and TLBO are chosen for comparison. The LIBSVM tool provided by Chih-Jen Lin (Tang et al. 2015) is used. The parameter of C is set as [1E−1, 1E3], and σ is set as [1E−2, 1E3]. 10-fold cross validation is adopted, and the accuracy of prediction (%) is used as the fitness value of the population-based heuristic algorithm. Population size is 20, and the max fitness evaluation is set as MAX_FES = 1000. The other parameter settings of JADE, jDE, GbABC, TLBO and MFLA are adopted as Table 2. Each algorithm is run 10 times, and the median, mean and standard deviation values are used for comparison.

Table 19 shows the details of the four datasets with different dimensions. Table 20 shows the median, mean and standard deviation of four datasets by the four algorithms. We can see that MFLA obtains the best accuracy than JADE, jDE, GbABC and TLBO on all datasets except for WDBC dataset. The mean value on WDBC dataset shows that GbABC is better than others. As we all known, the parameter optimization problem of SVM is a 'black box' optimization problem. In other words, this problem can be seen as an optimization problem of a complex function. The dataset is the constant of this function. The WDBC dataset changes the shape of this function so that it is suit to be solved by GbABC due to the search style of GbABC. Tables 21 and 22 show the comparison results of

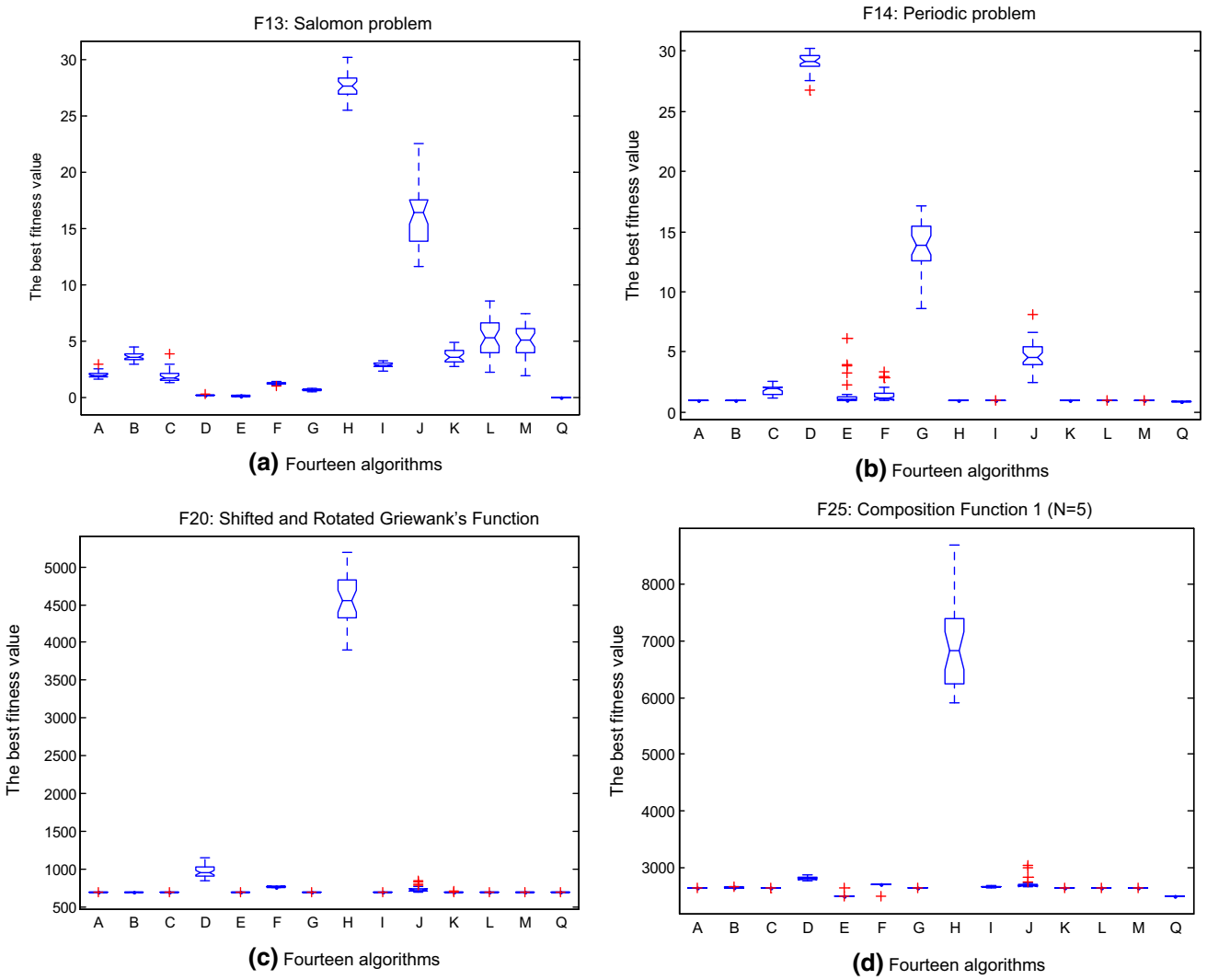

Fig. 6 Comparison results of 14 algorithms by box figure

Friedman test. Though the comparison of MFLA and other algorithms is not significant, MFLA is better than other algorithms with average rank of 3.07 that is larger than others.

In addition, we plot the convergence curve for the fair comparison including the search speed and the accuracy (shown in Fig. 7). We can see that MFLA obtains faster convergence speed than JADE, jDE, GbABC and TLBO.

Table 19 The details of four dataset

Dataset	Number of class	Dimension	Number of samples
OBCW	2	10	699
PBCW	2	33	198
WDBC	2	31	569
Heart	2	13	270

6 Conclusions

Shuffled frog leaping algorithm has been widely applied to many fields, especially the discrete optimization problems. However, the improved versions of SFLA for the continuous optimization are relatively less researched according to literature reviews. In this paper, we propose a memetic algorithm based on the shuffled frog leaping algorithm (called MFLA) for continuous optimization problem, which is integrated by three different units: memetic diffusion component (MD), memetic evolutionary component (ME) and memetic learning component (ML). The design of our proposed algorithm is based on the principle of synergistic coordination between exploitation and

Table 20 Comparison with MFLA and GbABC, JADE, jDE and TLBO for SVM

	GbABC(%)	JADE(%)	jDE(%)	TLBO(%)	MFLA(%)
1 Median	96.853	96.853	96.853	96.853	**96.996**
Mean	96.867	96.881	96.867	96.910	**96.967**
Std.	4.5240e−002	6.0320e−002	**4.5240e−002**	7.3877e−002	6.0320e−002
2 Median	81.313	82.323	82.323	82.323	**82.828**
Mean	81.919	82.323	82.323	82.323	**82.828**
Std.	6.6493e−001	**0.00**	**0.00e+000**	**0.00e+000**	1.4980e−014
3 Median	**98.067**	**98.067**	**98.067**	**98.067**	**98.067**
Mean	**98.155**	98.014	98.049	98.067e+001	98.120
Std.	1.2427e−001	1.6673e−001	5.5576e−002	1.4980e−014	**8.4894e−002**
4 Median	70.00	69.259	69.630	69.630	**70.370**
Mean	70.00	67.926	69.630	69.667	**70.370**
Std.	**0.00**	4.0953e+000	5.2378e−001	4.7655e−001	1.4980e−014

Bold fonts denote the best fitness values

Table 21 Friedman test for MFLA and other art-state algorithms

Order	Algorithm	Averages ranks
1	**MFLA**	4.50
2	TLBO	2.75
3	jDE	2.75
4	GbABC	2.63
5	JADE	2.38

Table 22 Statistical value of Friedman test for MFLA and other art-state algorithms (100D)

Method	Statistical value	p value
Friedman test	8.267	0.082

exploration considering the diversity of population. In MD unit, a shuffled process is used for the diversity of population. In ME unit, we improved the leaping rule by gravitational center and geometric center for enhancing the exploitation ability of SFLA. Considering the diversity of population, we use the Lévy flight method for the mutation of the two 'attractors'. In ML unit, we propose an adaptive leaping rule for improving the exploration ability of MFLA, which needs a new search method for searching in more wide space. For which, we consider the approximate random distribution for the individual selection and the dimension selection.

Finally, we design three groups of experiments on the benchmark datasets. In experiment (1) and (2), we compare the performance of MFLA to the 13 well-known algorithms including SPSO2011, JADE, jDE, CS, TLBO, DE/best/2/bin, GbABC, BBO, SFLA, GSA, GWO, SaDE, and AAA in low-dimensional space and high-dimensional space. Then, a typical real-world optimization problem, the parameter optimization problem of SVM, is used to compare MFLA to JADE, jDE, GbABC and TLBO. The experimental results and statistics show that the proposed algorithm outperforms existing well-known algorithms.

In the future, three aspects can be considered. Firstly, the 'attractors' of gravitational center and the geometric center can be used widely for other population-based heuristic algorithms. Secondly, the adaptive leaping rule can be improved with better adaptive selection method for individuals and dimensions. Thirdly, the proposed algorithm can be used widely for the other real-world optimization problems.

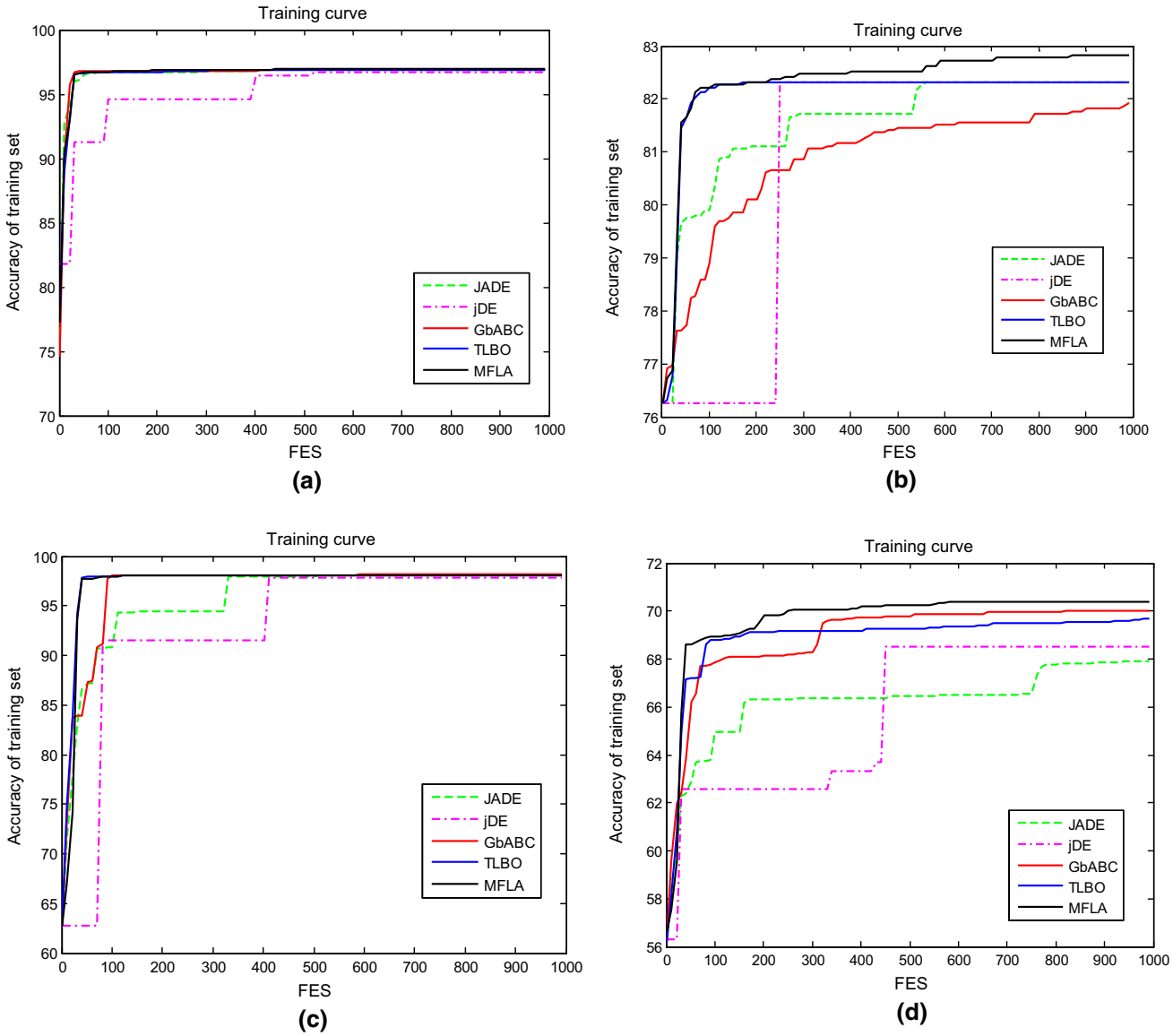

Fig. 7 Convergence curve of 5 algorithms for SVM. **a** OBCW, **b** PBCW, **c** WDB, **d** Heart

Acknowledgements The authors would like to thank the reviewers and editor for their very useful and constructive comments that helped to improve the quality of the paper. This work is supported by the Guang Dong Provincial Natural fund project, Drug-target interaction prediction method based on collaborative intelligent optimization (2016A030310300); the Natural Science Foundation of China under Grant (No. 61501128); NSFC, Research on reasoning of behavior trust for resisting collusive reputation attack (71401045); Guangdong province precise medicine and big data engineering technology research center for traditional Chinese medicine, Guang Dong Provincial Natural fund (2014A030313585, 2015A030310267, 2015A030310483). Major scientific research projects of Guangdong, Research of Behavioral Trust resisting collusion reputation attack based on implicit and explicit big behavior data analysis (2017WTSCX021). Philosophy and Social Sciences of Guangzhou '13th Five-Year' program (2018GZGJ48).

Compliance with ethical standards

Conflict of interest The author declares that there is no conflict of interest.

Ethical approval The work of this article does not involve use of human participants or animals.

References

Ahandani MA (2014) A diversified shuffled frog leaping: an application for parameter identification. Appl Math Comput 239:1–16

Ahandani MA, Alavi-Rad H (2015) Opposition-based learning in shuffled frog leaping: an application for parameter identification. Inf Sci 291:19–42

Bambha NK, Bhattacharyya SS, Teich J, Zitzler E (2004) Systematic integration of parameterized local search into evolutionary algorithms. IEEE Trans Evolut Comput 8(2):137–155

Bhattacharjee KK, Sarmah SP (2014) Shuffled frog leaping algorithm and its application to 0/1 knapsack problem. Appl Soft Comput 19:252–263

Bo J, Yuchun T, Yang-Qing Z, Chung-Dar L, Weber I (2005) Support vector machine with the fuzzy hybrid kernel for protein subcellular localization classification. In: Proceedings of IEEE international conference on fuzzy systems (FUZZ'05), Reno, NV, pp 420–423

Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evolut Comput 10(6):646–657

Cheng C, Zhang Y, Song M, Cheng G, Guo D, Cao J, Bao X (2014) Quantum-inspired shuffled frog leaping algorithm for spectrum sensing in cooperative cognitive radio network. In: International conference on human centered computing. Springer International Publishing, pp 80–92

Chow CK, Yuen SY (2011) An evolutionary algorithm that makes decision based on the entire previous search history. IEEE Trans Evolut Comput 15:741–768

Dawkins R (1976) The selfish gene. Clarendon Press, Oxford

Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evolut Comput 1:3–18

Eusuff MM, Lansey KE (2003) Optimization of water distribution network design using the shuffled frog leaping algorithm. J Water Resour Plan Manag 129(3):210–225

Fang C, Wang L (2012) An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem. Comput Oper Res 39(5):890–901

Holliday D, Resnick R, Walker J (1993) Fundamentals of physics. Wiley, New York

Huang H, Qin H, Hao ZF, Lim A (2012) Example-based learning particle swarm optimization for continuous optimization. Inf Sci 182:125–138

Iacca G, Neri F, Mininno E, Ong Y-S, Lim M-H (2012) Ockham's Razor in memetic computing: three stage optimal memetic exploration. Inf Sci 188:17–43

Karaboga D (2005) An idea based on honey bee swarm for numerical optimization, technical report TR06, Erciyes University

Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceeding IEEE international conference neural network, Perth, Western Australia, pp 1942–1948

Kóczy LT, Földesi P, Tűű-Szabó B (2017) Inf Sci 000: 1–12

Lam AYS, Li VOK, Yu JJQ (2012) Real-coded chemical reaction optimization. IEEE Trans Evol Comput 16:339–353

Le MN, Ong YS, Jin Y, Sendhoff B (2012) A unified framework for symbiosis of evolutionary mechanisms with application to water clusters potential model design. IEEE Comput Intell Mag 7(1):20–35

Lei D, Guo X (2015) A shuffled frog-leaping algorithm for hybrid flow shop scheduling with two agents. Expert Syst Appl 42(23):9333–9339

Li J, Pan Q, Xie S (2012a) An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems. Appl Math Comput 218(18):9353–9371

Li X, Luo J, Chen M-R, Wang N (2012b) An improved shuffled frog-leaping algorithm with extremal optimisation for continuous optimisation. Inf Sci 192:143–151

Li Y, Jiao L, Li P, Bo W (2014) A hybrid memetic algorithm for global optimization. Neurocomputing 134:132–139

Liang JJ, Qu BY, Suganthan PN (2013) Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Comput Intell Lab, Zhengzhou Univ., Zhengzhou, China, and Nanyang Technol. Univ., Singapore, Tech. Rep. 201311, Dec

Liao T, Aydın D, Stützle T (2013) Artificial bee colonies for continuous optimization: experimental analysis and improvements. Swarm Intel 7(4):327–356

Lim WH, Isa NAM (2014) Bidirectional teaching and peer-learning particle swarm optimization. Inf Sci 280:111–134

Liu H, Yi F, Yang H (2016) Adaptive grouping cloud model shuffled frog leaping algorithm for solving continuous optimization problems. Comput Intell Neuro 2016:25

Liu C, Niu P, Li G et al (2018a) Enhanced shuffled frog-leaping algorithm for solving numerical function optimization problems. J Intell Manuf 29:1133–1153

Liu W, Gong M, Wang S, Ma L (2018b) A two-level learning strategy based memetic algorithm for enhancing community robustness of networks. Inf Sci 422:290–304

Luo X-H, Ye Y, Xia L (2008) Solving TSP with shuffled frog-leaping algorithm. In: Eighth international conference on intelligent system design and application, ISDA'08, 3

Luo J et al (2015) A novel hybrid shuffled frog leaping algorithm for vehicle routing problem with time windows. Inf Sci 316:266–292

Mantegna RN (1991) Levy walks and enhanced diffusion in Milan stock exchange. Phys A 179:232–242

Mantegna RN (1994) Fast, accurate algorithm for numerical simulation of Levy stable stochastic process. Phys Rev E 5(49):4677–4683

Merz CJ, Blake CL (2015) UCI repository of machine learning databases. http://www.ics.uci.edu/-mlearn/MLRepository.html

Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61

Moscato P (1989) On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms, technical reports. 826, Caltech concurrent computation program

Moscato P, Norman M (1989) A competitive and cooperative approach to complex combinatorial search, technical reports. 790, Caltech Concurrent Computation Program

Müller CL, Baumgartner B, Sbalzarini IF (2009) Particle swarm CMA evolution strategy for the optimization of multi-funnel landscapes. In: Proceedings of IEEE congress evolutionary computation, pp 18–21

Narimani MR (2011) A new modified shuffle frog leaping algorithm for non-smooth economic dispatch. World Appl Sci J 12(6):803–814

Nguyen QH, Ong Y-S, Lim MH (2009) A probabilistic memetic framework. IEEE Trans Evolut Comput 13(3):604–623

Omran MGH, Engelbrecht AP, Salman A (2007) Differential evolution based particle swarm optimization. In: Proceedings of swarm intelligence symposium, Honolulu, HI, USA, pp 112–119

Ong YS, Keane AJ (2004) Meta-Lamarckian learning in memetic algorithm. IEEE Trans Evol Comput 8(2):99–110

Ong Y-S, Lim M-H, Zhu N, Wong K-W (2006a) Classification of adaptive memetic algorithms: a comparative study. IEEE Trans Syst Man Cybern B Cybern 36(1):141–152

Ong Y-S, Lim M-H, Zhu N, Wong K-W (2006b) Classification of adaptive memetic algorithms: a comparative study. IEEE Trans Syst Man Cybern B Cybern 36(1):141–152

Ong YS, Lim MH, Chen X (2010) Memetic computation—past, present & future [research frontier]. IEEE Comput Intell Mag 5(2):24–31

Ou Y, Sun Y (2011) Grid task scheduling strategy based on improved shuffled frog leaping algorithm. Comput Eng 37(21):146–151
Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evolut Comput 13(2):398–417
Rao RV, Savsani VJ, Vakharia DP (2012) Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. Inf Sci 183:1–15
Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248
Roy P, Roy P, Chakrabarti A (2013) Modified shuffled frog leaping algorithm with genetic algorithm crossover for solving economic load dispatch problem with valve-point effect. Appl Soft Comput 13(11):4244–4252
Samma H, Lim CP, Saleh JM (2016) A new reinforcement learning-based memetic particle swarm optimizer. Appl Soft Comput 43:276–297
Schutz B (2003) Gravity from the ground up. Cambridge University Press, Cambridge
Sharma S, Sharma TK, Pant M et al (2015) Centroid mutation embedded shuffled frog-leaping algorithm. Proc Comput Sci 46:127–134
Shin HJ, Eom DH, Kim SS (2005) One-class support vector machine-an application in machine fault detection and classification. Comput Ind Eng 48(2):395–408
Simon D (2008) Biogeography-based optimization. IEEE Trans Evolut Comput 12:702–713
Smith J (2007a) Coevolving memetic algorithms: a review and progress report. IEEE Trans Syst Man Cybern B Cybern 37(1):6–17
Smith JE (2007b) Coevolving memetic algorithms: a review and progress report. IEEE Trans Syst Man Cybern B Cybern 37(1):6–17
Storn RM, Price KV (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359
Sun J, Garibaldi JM, Krasnogor N, Zhang Q (2013) An intelligent multi-restart memetic algorithm for box constrained global optimization. Evolut Comput 21(1):107–147
Tang D, Cai Y, Zhao J, Xue Y (2014) A quantum behaved particle swarm optimization with memetic algorithm and memory for continuous non-linear large scale problems. Inf Sci 289:162–189
Tang D, Dong S, Jiang Y, Li H, Huang Y (2015) ITGO: invasive tumor growth optimization algorithm. Appl Soft Comput 36:670–698
Tang D, Yang J, Dong S, Liu Z (2016) A Lévy flight-based shuffled frog-leaping algorithm and its applications for continuous optimization problems. Appl Soft Comput 49:641–662
Uymaz SA, Tezel G, Yel E (2015) Artificial algae algorithm (AAA) for nonlinear global optimization. Appl Soft Comput 31:153–171
Vapnik V (1995) The nature of statistical learning theory. Springer, New York
Vasan P (2014) Handbook of research on artificial intelligence techniques and algorithms (2 volumes). https://doi.org/10.4018/978-1-4666-7258-1
Vasant P, Weber G-W, Dieu VN (2016) Handbook of research on modern optimization algorithms and applications in engineering and economics. https://doi.org/10.4018/978-1-4666-9644-0
Veček N, Mernik M, Č repinšek M (2014) A chess rating system for evolutionary algorithms: a new method for the comparison and ranking of evolutionary algorithms. Inf Sci 277:656–679
Wang L, Fang C (2011) An effective shuffled frog-leaping algorithm for multi-mode resource-constrained project scheduling problem. Inf Sci 181(20):4804–4822
Wang H, Moon I, Yang S, Wanga D (2012) A memetic particle swarm optimization algorithm for multimodal optimization problems. Inf Sci 197:38–52
Wang H, Zhang K, Tu X (2015) A mnemonic shuffled frog leaping algorithm with cooperation and mutation. Appl Intell 43(1):32–48
Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evolut Comput 1(1):67–82
Yang XS, Deb S (2009) Cuckoo search via Ĺevy flights. Proceedings of world congress on nature and biologically inspired comput IEEE Publications, USA, pp 210–214
Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. IEEE Trans Evolut Comput 13(5):945–958
Zhou HG, Yang CD (2006) Using immune algorithm to optimize anomaly detection based on SVM. In: Proceedings of IEEE international machine learning and cybernetics conference, Dalian, China, pp 4257–4261
Zhou Z, Ong Y-S, Nair P, Keane A, Lum K-Y (2007) Combining global and local surrogate models to accelerate evolutionary optimization. IEEE Trans Syst Man Cybern C Appl Rev 37(1):66–76

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.