Abstract: As a new swarm intelligence methods inspired by biological evolution and a global search algorithm, cuckoo search algorithm (CS) simulated the behavior of baby bearing and Levy flights. In order to tackle with multi-dimension function optimization problems, this strategies, as a result of taking the same step and random walk, may reduce the convergence speed and the quality of the solution on the algorithm due to different search capability of every individual. An improved CS algorithm named Dynamic Adaptive Multi-Cuckoo Search Algorithm (DAMCS), was proposed to overcome this shortage. On the basis of the difference of fitness value of the individual, the population of the proposed algorithm consists of one elite sub-population, one ordinary sub-population and one developing sub-population. Each sub-population was evolved with different steps of Levy flights. The step was changed adaptively according to different sub-populations and calculation times of fitness. The population of the proposed algorithm consists of one elite sub-population and one developing sub-population by the fitness value of the population after they were spotted. Elite sub-population learns from the best individual to strengthen the local search ability, developing the sub-population evolved with Mutation operator of Differential Evolution (DE) algorithm to overstep the local optimum. Sub-population will be transformed basing on the fitness value in the next iteration, and each sub-population communicate information well. The experimental results of 8 standard test functions indicate that DAMCS algorithm behaves stronger performance on convergence as well as adaptation and confirms the effectiveness when compared with segmental improved cuckoo search algorithms and other swarm intelligence algorithms.

Key words: Cuckoo search algorithm, multi-population, differential evolution algorithm, Levy flights, diversity, function optimization.

1. Introduction

YANG X S and DEB S proposed a novel random optimization method called cuckoo search algorithm, by simulating the cuckoo’s living habits in 2009 [1]. Its main thought is relation to two basic strategies: Levy flights and preference random walk, which can balance the global and local search ability of the cuckoo search algorithm to attain an efficient optimization model [2].

CS is a superior global search algorithm in virtue of few parameters, simple operation, easy implementation and strong global search capability, etc. It also has been successfully applied to various optimization problems such as traveling salesman problem (TSP) [3], proportion integration differentiation (PID) controller [4] and so on. Owing to completely depending on the mechanism of random walking in the process of search, the algorithm has poor local search ability. Aimed at correcting the flaw, many CS variants have been proposed by scholars, which are mainly the improvement of the parameters of step and
detection probability as well as the integration with other algorithm. Zheng H. Q. et al. [5] designed a self-adaptive step cuckoo search algorithm (ASCS) by introducing the mechanism of the self-adaptive step. The simulation results manifest that the convergence rate and the quality of the optimal solution are improved in the early iterations of the algorithm. By means of investigating the correlation between the parameters and convergence, the step and detection probability were improved by Jin Q. et al [6], which can efficiently improve the global convergence ability of the algorithm. Cuckoo search algorithm with dimension by dimension improvement (DDICS) was proposed by Wang L. J. et al [7], which made up the drawback of interference phenomena among dimension by dimension improvement in multi-dimension function optimization problems. Dynamic adaptation cuckoo search algorithm (DACS), was put forward by Zhang Y. W. et al [8], introduced the feedback control scheme of algorithm parameters into algorithm framework in CS. Meanwhile, by utilizing Rechenberg’s 1/5 criteria as the assessment index in evolution process, it introduced the learning factor to balance the diversification and intensification of population. For the sake of adaptation to various optimization problems, an improved CS algorithm, named swarm feature feedback cuckoo search (SFFCS) algorithm, was proposed by Jia Y. L. et al [9] on the basis of the feedback control principle. Swarm features such as age structure and success rate of mutation were introduced as feedback information to adjust the parameters dynamically. Xiao H. H. and Duan Y. M. [10] designed an improved CS algorithm with differential evolution strategy (DECS). The individual variation was completed in the algorithm before population with two weighted differences increasing on its individuals before entering the next iteration, then crossover operation and select operation were performed to obtain optimal individual.

It is worth noting that DACS [8] and SFFCS [9] belong to the same kind of improved modes, which adjusted step and detection probability dynamically according to the population characteristics and achieved superior performance. On the basis of the difference of fitness value and search capability of the the individuals, the population is divided into some sub-population, whose the step and the location after being discovered of the individuals are updated. As a consequence, an improved CS algorithm, called Dynamic Adaptive Multi-Cuckoo Search Algorithm (DAMCS) was proposed. The population of the proposed algorithm consisted of one elite sub-population, one ordinary sub-population and one developing sub-population for the difference of the fitness value of the individuals. Each sub-population was evolved with different step to adjust Levy flights. It changed the step adaptively in line with the degree of fitness calculation and the fitness value. The population of the algorithm consisted of one elite sub-population and one sub-population to be developed by the fitness value of the population after they were spotted, Elite sub-population learns from the best individual to strengthen local search ability, developing sub-population evolved with mutation operator of differential evolution algorithm to overstep the local optimum. Sub-population will be transformed basing on the fitness value in the next iteration, and each sub-population fully communicated information. The experimental results of 8 standard test functions indicate that, DAMCS algorithm have higher accuracy and stronger adaptability when compared with other improved cuckoo search algorithms and other swarm intelligence algorithms.

2. Cuckoo Search

In nature, cuckoos find an appropriate nest to lay eggs in a random or random-like way. To apply the cuckoo brood behavior for optimization problems, YANG X S and DEB S idealized the brood parasitic behavior of some cuckoos and the following three rules have been put forward [11]:

a. In cuckoo population, each cuckoo puts one egg to a randomly selected nest;

b. The high-quality nests will not change, and this can guarantee the cuckoo population that contains the better solutions, not worse than before at least;
c. The nest number is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability \( p_a \in [0, 1] \).

The process of hatching cuckoo is very complicated, and CS algorithm was proposed by simplifying it. A feasible solution corresponding to an individual, for instance, d dimensional vector, represents an individual, namely, a candidate in the decision space. Each individual corresponds to a fitness value. Besides, the quality of an individual is measured through the fitness value, and the offspring population is generated by Levy flight.

\[
X_i^{t+1} = X_i^t + \alpha \odot L(\lambda) \quad (i = 1, 2, ..., n)
\]

where \( X_i^t \) is the \( i \)th individual of \( t \)th generation, \( \alpha \) is the step length of Levy flight, and \( \odot \) is a positive number. The product stands for entry-wise multiplications. \( L(\lambda) \) is a random walk, the step is drawn from Levy distribution.

\[
L(\lambda) \sim \mu = r^{-\lambda} \quad (1 < \lambda \leq 3)
\]

Levy flight is a random linear motion, which is randomly directed and scale-free, random orientation and no characteristic scale. The step of each sequence obeys the heavy tailed distribution, sometimes long, sometimes short, which can guarantee the global optimal solution found successfully [12]. A random number \( R \) is randomly generated when the individual is updated. Detection probability \( P_a \) is compared with the random number \( R \), if \( R > P_a \), the individual changes as follows:

\[
X_i^{t+1} = X_i^t + rand_d^* (X_k^t - X_i^t) \quad (i = 1, 2, ..., n)
\]

where \( rand \in [0, 1]^d \), \( X_i^t \) and \( X_k^t \) are random individuals of \( t \)th generation; otherwise, it will keep the current location unchanged.

3. Dynamic Adaptive Multi-cuckoo Search Algorithm

3.1. DAMCS

CS algorithm randomly generates the offspring generation by Levy flight, and it has disadvantages of slow convergence and low precision for the irregular step. Therefore, inspired by literature [13], the step adjustment of dynamic adaptation was proposed. The population consists of one elite sub-population, one ordinary sub-population and one developing sub-population by different fitness. \( n \) is the population size, \( f_i \) is the fitness value of the \( i \)th individual, and \( f_{avg} = \frac{\sum f_i}{n} \) is the average fitness value of the population, \( f_{avg1} \) is the average fitness value of the individuals whose fitness value is less than \( f_{avg} \), \( f_{avg2} \) is the average fitness value of the individuals whose fitness value is bigger than \( f_{avg} \).

If \( f_i > f_{avg} \) and the \( i \)th individual is part of elite sub-population, the \( i \)th individual is part of ordinary sub-population, if \( f_i \leq f_{avg1} \), the \( i \)th individual is part of developing sub-population. The step changes adaptively on account of different sub-population. If the \( i \)th individual is spotted and \( f_i > f_{avg} \), it is part of elite sub-population, the individuals of the elite population generate offspring by learning from the best individual, then the convergence rate is accelerated and the capacity of local search is enhanced; if the \( i \)th individual is spotted and \( f_i \leq f_{avg} \), it is part of developing sub-population, the individuals of the elite
population generate offspring by mutation operator of DE algorithm, which can help the improved algorithm to skip the local optimum.

3.2. Step Adjustment of Dynamic Adaptation

CS algorithm, randomly generating the offspring generation by Levy flight is quite aimless, which therefore leads to slow convergence rate for the undirected searching methods. In order to accelerate convergence rate and enhance diversity of the population, the step of the individual was altered adaptively in the light of fitness value and calculation times of fitness, and eventually figured out that different sub-population has different step adjustment.

The individuals of elite sub-population have greater fitness value and near optimal individual. In order to improve the algorithm local search ability, the step should be a small variable. Due to the algorithm gradually converging with the increase of fitness calculation times, the progressively smaller step is designed to prevent the algorithm missing the optimum. The step of the individual of elite sub-population is rectified as follows:

\[
\alpha_i = \frac{1}{\sqrt[4]{T}} \left[ (\alpha_{ub} - \alpha_{lb})(f_i - f_{avg})/(f_{max} - f_{avg}) \right]
\]  

(4)

where \(T\) is the calculation times of fitness, \(\alpha_{ub}\) is the upper limit of the step, \(\alpha_{lb}\) is the lower limit of the step, and \(f_{max}\) is the maximum fitness value of the current population.

The individuals of ordinary sub-population have strong global and local search ability, and the step is a small variable. The step should be large in initial period; nevertheless, it reduces nonlinearly with the increase of fitness calculation times, which ensures strong global and local search ability and the algorithm has a fine search accuracy in later period. The step of the individual of ordinary sub-population is adjusted as follows:

\[
\alpha_i = \frac{1}{\sqrt[4]{T}} \left[ \alpha_{lb} + (\alpha_{ub} - \alpha_{lb})(T/T_{max})^2 \right]
\]  

(5)

where \(T_{max}\) is the maximum times of fitness calculations.

The individuals of developing sub-population are far away from the optimum, but it might help to jump out of the local optimum. Hence, the step follows a normal distribution which possesses good randomness, and it reduces with the increase of fitness calculation times. The step of the individual of ordinary sub-population is modulated as follows:

\[
\alpha_i = \frac{1}{\sqrt[4]{T}} \cdot N(0,1)
\]  

(6)

where \(N(0,1)\) is the normal distribution.

3.3. Learning from Optimum

If the individual of the elite sub-population is spotted, it generates offspring by equation (3) and the global convergence of the algorithm is guaranteed by this way of preference random walking, while the convergence rate slows down obviously. Aimed at this deficiency, the strategy of learning from the optimum was proposed.

\[
x_{i+1} = \text{Best}^t + R^* (x^*_i - x^*_j)
\]  

(7)

where \(\text{Best}^t\) is the optimum of \(i_{th}\) generation, \(R \in [-1,1]\) is a random number, \(X^*_i\) and \(X^*_j\) are randomly individuals of \(t_{th}\) generation.
3.4. Mutation Operator

Differential Evolution (DE) algorithm is an evolutionary algorithm based on population differences, and an intermediate population can be got through recombination according to the difference of population. Then, the offspring is generated by competition between the current population and the intermediate population. The most marked characteristic of DE algorithm is mutation operator, and it generates the offspring by the difference of two random weighted individuals after one random individual is selected. The mutation operator is added to the algorithm which reinforces its search ability. In early iterations, the individuals in the population are quite different, and the global search of the algorithm is enhanced by mutation operator; however, the diversity of the population decreases and the algorithm converges gradually with more and more iterations. The local search of the algorithm is enhanced by mutation operator. The individuals of the developing sub-population are used to help jumping out of local optimum by reference to mutation operation, algorithm global convergence is guaranteed. The formula of mutation operation is as follows:

$$x_{i}^{t+1} = x_i^t + F (x_k^t - x_j^t)$$

where $x_j^t, x_k^t, x_i^t$ are random individuals of $t$th generation, $F \in [-1,1]$ is a random variant.

3.5. Implementations of DAMCS

It is necessary to emphasize the process of the improved algorithm, and the implementation of DAMCS has been introduced as follows:

Step 1: Randomly generate an initial population $X'$ of size $n$ from the decision space $[lb, ub]$.

Step 2: Calculate the fitness $f_i$ of each individual and record the best bird’s nest.

Step 3: if $f_i > f_{avg2}$, the step $\alpha$ is calculated by equation (4); if $f_{avg1} \leq f_i < f_{avg2}$, the step $\alpha$ is calculated by equation (5); if $f_i \leq f_{avg1}$, the step $\alpha$ is calculated by equation (6).

Step 4: Generate each individual by $X_i^{t+1} = X_i^t + \alpha_i \in L(\lambda)$.

Step 5: Calculate the fitness of each offspring individual and save the individual with a better fitness.

Step 6: Random number $r \in (0,1)$ with uniform distributed is generated, when $r > P$, if $f_i > f_{avg}$, the individual will change by equation (7), otherwise the individual will change by equation (8).

Step 7: Calculate the fitness of each offspring individual and save the individual with a better fitness.

Step 8: If the stopping criterion is satisfied, stop and put out the final solution, otherwise go to Step 3.

4. Simulation

4.1. Comparison with Other Variants of CS Algorithm

4.1.1. Experimental design

<table>
<thead>
<tr>
<th>Test functions</th>
<th>The design space</th>
<th>Optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1 = \sum_{i=1}^{d} x_i^2$</td>
<td>$[-100,100]$</td>
<td>0</td>
</tr>
<tr>
<td>$f_2 = \sum_{i=1}^{d} (\frac{1}{j \leq i} x_i)^2$</td>
<td>$[-100,100]$</td>
<td>0</td>
</tr>
</tbody>
</table>
In this paper, eight test examples (denoted as $f_1, \sim f_8$) with different characteristics are used to investigate the effectiveness of MADCS and compared with the classical CS, ASCS [5], DECS [6], DACS [7], DDICS [8] and SFFCS [11]. The selected test environment is as follows: Windows7, CPU for Intel dual core 2.6GHZ, memory 6G, Matlab7.0. The parameters of the algorithm are set as follows: Population size n=25, discovery probability $P_a = 0.25$, the maximum number of fitness calculations $T_{\text{max}} = 2.5 \times 10^5$, lower limit of the step $\alpha_l = (ub - lb)/4000$, upper limit of the step $\alpha_u = (ub - lb)/200$. The parameter settings of other algorithms are listed in the literature [5]-[8], [11]. Different algorithms generate different numbers of new nests, therefore, 30 independent runs are executed on each test instance and fitness calculation times are used as evaluation criterion to guarantee the fairness. The test function is shown in Table 1.

### 4.1.2. Analysis of convergence precision

Table 2 presents the mean and standard deviation of the optimization of functions deriving from CS, ASCS, DECS, DDICS, SFFCS and MADCS, and the optimal result has been expressed in bold. Table 2 indicates that all algorithms can find the real optimum for function $f_5$ except DACS. The difference between CS and ASCS is not large in that ASCS only makes adjustment of step, but ASCS is easier to fall into local optima than CS. As to $f_5, f_6, f_7$, DECS, DDICS and SFFCS can find real optimum successfully, and the convergence accuracy of DECS is higher than CS and ASCS in all functions except $f_2, f_3, f_4$ for mutation operation. The convergence accuracy of DACS is higher than CS in all functions except $f_4, f_5, f_6$, and it’s higher than DECS in all functions except $f_2, f_3$. Due to the strategy of dimension by dimension, the DDICS successfully finds the global optimum in $f_1, f_5, f_6$ and shows good effect in all function expect $f_4$. The convergence accuracy of SFFCS is lower than CS and ASCS in $f_4$, but the quality of the solution of SFFCS is better than CS, ASCS, DECS, DACS, SFFCS and MADCS in all functions except $f_4$, the correctness and validity of the selection of population characteristics are verified. The statistical test reveals that MADCS is significantly better than several other algorithms in all the test functions. As to $f_1, f_3, f_5, f_6, f_8$, DAMCS is able to find all the optimal solution over a majority of runs; particularly, the quality of solution of DAMCS is 66, 66, 68, 60, 64, 45 orders of magnitude higher than that of CS, ASCS, DECS, DACS, DDICS and SFFCS respectively in $f_2$, and it shows strong ability of global search and adaptability to different problems.
standard deviation of DAMCS is smaller than that of other algorithms, which shows the stability of DAMCS.

Table 2. The Average Optimal Value of 30 Runs in the Same Calculation Times of Fitness

<table>
<thead>
<tr>
<th>Functions</th>
<th>Indicators</th>
<th>CS</th>
<th>ASCS</th>
<th>DECS</th>
<th>DACS</th>
<th>DDICS</th>
<th>SFFCS</th>
<th>DAMCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>f_1</td>
<td>mean</td>
<td>1.905e-39</td>
<td>8.082e-32</td>
<td>3.747e-92</td>
<td>3.218e-49</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>5.825e-37</td>
<td>1.293e-31</td>
<td>5.070e-92</td>
<td>2.198e-49</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>f_2</td>
<td>mean</td>
<td>2.941e-04</td>
<td>5.138e-04</td>
<td>2.254e-02</td>
<td>1.171e-10</td>
<td>7.054e-06</td>
<td>1.647e-25</td>
<td>2.190e-70</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>3.412e-04</td>
<td>6.184e-04</td>
<td>1.504e-02</td>
<td>1.589e-10</td>
<td>5.355e-06</td>
<td>2.300e-25</td>
<td>6.925e-70</td>
</tr>
<tr>
<td>f_3</td>
<td>mean</td>
<td>2.417e+00</td>
<td>1.229e+00</td>
<td>5.030e+00</td>
<td>1.916e-14</td>
<td>2.806e-02</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>1.753e+00</td>
<td>1.513e+00</td>
<td>4.322e+00</td>
<td>2.046e-14</td>
<td>3.871e-03</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>f_4</td>
<td>mean</td>
<td>8.105e-03</td>
<td>8.573e-03</td>
<td>3.261e-02</td>
<td>4.135e-02</td>
<td>1.923e-01</td>
<td>7.172e-02</td>
<td>1.358e-05</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>4.809e-03</td>
<td>2.892e-03</td>
<td>1.443e-02</td>
<td>1.553e-02</td>
<td>2.929e-02</td>
<td>1.150e-02</td>
<td>1.085e-05</td>
</tr>
<tr>
<td>f_5</td>
<td>mean</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.975e-16</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.629e-15</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>f_6</td>
<td>mean</td>
<td>1.479e+01</td>
<td>1.289e+01</td>
<td>0</td>
<td>2.094e+02</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>3.726e+00</td>
<td>3.816e+00</td>
<td>0</td>
<td>9.556e+01</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>2.557e-01</td>
<td>3.117e-15</td>
<td>0</td>
<td>2.512e-15</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>f_8</td>
<td>mean</td>
<td>1.304e-18</td>
<td>8.480e-15</td>
<td>1.969e-06</td>
<td>2.079e-35</td>
<td>7.452e-87</td>
<td>9.65e-247</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>7.043e-18</td>
<td>7.237e-15</td>
<td>1.943e-06</td>
<td>1.208e-35</td>
<td>8.613e-85</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3. The Average Calculation Times of Fitness of 30 Runs in the Same Convergence Precision

<table>
<thead>
<tr>
<th>Functions</th>
<th>CS</th>
<th>ASCS</th>
<th>DECS</th>
<th>DACS</th>
<th>DDICS</th>
<th>SFFCS</th>
<th>DAMCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>f_1</td>
<td>5.686e+04</td>
<td>5.524e+04</td>
<td>2.485e+04</td>
<td>4.389e+04</td>
<td>8.641e+03</td>
<td>5.507e+03</td>
<td>2.223e+03</td>
</tr>
<tr>
<td>f_2</td>
<td>2.500e+05</td>
<td>2.500e+05</td>
<td>2.500e+05</td>
<td>1.653e+05</td>
<td>2.324e+05</td>
<td>8.061e+04</td>
<td>1.194e+04</td>
</tr>
<tr>
<td>f_3</td>
<td>2.500e+05</td>
<td>2.500e+05</td>
<td>2.500e+05</td>
<td>1.945e+05</td>
<td>2.500e+05</td>
<td>3.454e+04</td>
<td>2.774e+03</td>
</tr>
<tr>
<td>f_4</td>
<td>2.500e+05</td>
<td>2.500e+05</td>
<td>2.500e+05</td>
<td>2.500e+05</td>
<td>2.500e+05</td>
<td>2.500e+05</td>
<td>2.124e+05</td>
</tr>
<tr>
<td>f_5</td>
<td>5.577e+04</td>
<td>6.092e+04</td>
<td>2.298e+04</td>
<td>1.483e+05</td>
<td>6.178e+03</td>
<td>4.661e+03</td>
<td>3.233e+03</td>
</tr>
<tr>
<td>f_6</td>
<td>2.500e+05</td>
<td>2.500e+05</td>
<td>6.218e+04</td>
<td>2.500e+05</td>
<td>1.708e+04</td>
<td>1.077e+04</td>
<td>2.612e+03</td>
</tr>
<tr>
<td>f_7</td>
<td>2.500e+05</td>
<td>1.023e+05</td>
<td>3.218e+04</td>
<td>4.256e+04</td>
<td>1.159e+04</td>
<td>8.611e+03</td>
<td>2.715e+03</td>
</tr>
<tr>
<td>f_8</td>
<td>9.184e+04</td>
<td>1.042e+05</td>
<td>3.272e+04</td>
<td>5.281e+04</td>
<td>1.140e+04</td>
<td>6.951e+03</td>
<td>3.753e+03</td>
</tr>
</tbody>
</table>

4.1.3. Analysis of convergence rate
Fig. 1. The mean optimal values in different algorithms over 30 independent runs in the same calculation times of fitness. (a)~(h): $f_1 \sim f_8$.

Table 3 presents the average of the optimal value deriving from CS, ASCS, DECS, DACS, DDICS, SFFCS and MADCS. And the optimal result has been expressed in bold. In this paper, in order to prevent the algorithm from being continuously calculated as a result of local optimization, if the algorithm fails to reach the convergence error within $2.5 \times 10^5$ calculation times of fitness, it fails to find optimal values and its' calculation times of fitness is $2.5 \times 10^5$. From Table 3, as for $f_2$, $f_3$ and $f_4$, CS, ASCS and DECS all can't
reach the convergence optimum quickly.

In order to illustrate the capability of convergence of different algorithm visually, Fig. 1 plots convergence curve of different algorithm of $f_i \sim f_8$, and it shows that DAMCS has higher convergence speed than other algorithms (including CS, ASCS, DECS, DACS, DDICS and SFFCS).

4.1.4. Analysis of population diversity

Diversity, as an important indicator to evaluate the quality of algorithm, Diversity determines the global and local search capability and whether being in stagnation or not. The diversity indicator [14] is computed in this paper according to

$$\text{Diversity}(t) = \frac{1}{|L|} \times n \times \sqrt{\sum_{j=1}^{d} \left(\frac{X_{i,j,t} - \bar{X}_{j,t}}{Lt} \right)^2}$$

(12)

where $|L|$ is maximum diagonal length, $n$ is population size, $d$ is dimension of the problem, $X_{i,j,t}$ is the $j$th dimension of the $i$th individual in the $t$th generation, and $\bar{X}_{j,t}$ is the average value of the $j$th dimensions of the all individuals in the $t$th generation. Taking functions $f_5$ and $f_7$ as examples, the number of iterations is set to 4000, and the variation curves of the population diversity of DAMCS and CS are plotted in Fig. 2. As for CS, the population diversity of declines rapidly and disappears almost around 900~1900 generation, but the population diversity was restored to a high level. As for MADCS, the population diversity declines rapidly in initial period but increases and becomes stable at a high level in later period for the different updating step methods of different sub-population and mutation operation, and it illustrates that the population diversity of MADCS is well maintained.

4.1.5. Adaptability analysis of step $\alpha$

Fig. 3 plots the varying curve of step $\alpha$ of a random individual as iteration goes on with example in function $f_2$. Due to the fact that the step decreases along with the increase of the calculation times of fitness, its' trend reduced gradually of the overall curve except some fluctuation. Because the individual can belong to different sub-population in different periods, the variable curve of step has no monotonicity, at times relatively stable and at times greatly variable. MADCS can easily skip the local optimum even if it is stuck sometimes, and it illustrates the global search capability.

4.2. Comparison with Other Improved Evolutionary Algorithms

![Fig. 2. Mean of CS and MADCS over 30 independent runs in the same iteration. (a) $f_5$ (b) $f_7$.](image-url)
Fig. 3. The varying curve in MADCS of step $\alpha$.

In this subsection, this paper compared MADCS with GSDE, JADE, BSA, ABC and GSA involved in literature [15]. $f_3 \sim f_6$ from 8 standard functions are selected randomly as the test functions, and the parameter of DAMCS remains unchangeable and the other algorithms are detailed in literature[15]. Table 4 presents the average value and standard deviation of the optimization of functions deriving from the six algorithm. JADE, BSA, ABC and GSA can’t find the global optimal solution within the specified calculation number of fitness. As to $f_3$, $f_4$, GSDE can’t find the global optimal solution, but its search accuracy is further higher than that of JADE, BSA, ABC and GSA, and GSDE discovers the global optimal solution of $f_5$, $f_6$. DAMCS can find the global optimal solution of $f_1$, $f_3$ and $f_4$, and it is superior to the other five algorithms in the search accuracy.

Table 4. The Average Optimal Value of 30 Runs in the Same Calculation Times of Fitness

<table>
<thead>
<tr>
<th>Functions</th>
<th>Indicators</th>
<th>JADE</th>
<th>BSA</th>
<th>ABC</th>
<th>GSA</th>
<th>GSDE</th>
<th>DAMCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_3$</td>
<td>mean</td>
<td>2.798e-03</td>
<td>2.940e+01</td>
<td>5.567e-02</td>
<td>2.264e+01</td>
<td>6.901e-11</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>2.122e-03</td>
<td>2.285e+01</td>
<td>5.153e-02</td>
<td>1.193e-01</td>
<td>1.508e-10</td>
<td>0</td>
</tr>
<tr>
<td>$f_4$</td>
<td>mean</td>
<td>5.778e-04</td>
<td>5.778e-04</td>
<td>7.777e-02</td>
<td>4.135e-02</td>
<td>7.746e-04</td>
<td>4.055e-05</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>4.809e-03</td>
<td>9.817e-03</td>
<td>1.558e-02</td>
<td>1.553e-02</td>
<td>1.410e-04</td>
<td>1.255e-05</td>
</tr>
<tr>
<td>$f_5$</td>
<td>mean</td>
<td>4.491e+00</td>
<td>1.612e+02</td>
<td>2.293e+02</td>
<td>1.975e-16</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>5.749e-01</td>
<td>6.639e+01</td>
<td>3.307e+01</td>
<td>2.629e-15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$f_6$</td>
<td>mean</td>
<td>2.036e+02</td>
<td>2.877e+02</td>
<td>2.408e+02</td>
<td>2.094e+02</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>1.185e+01</td>
<td>2.781e+01</td>
<td>1.616e+01</td>
<td>9.556e+01</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

5. Conclusion

Dynamic Adaptive Multi-Cuckoo Search Algorithm was proposed in respect of defects of slow convergence rate and low search precision of classic CS. On the basis of the difference of fitness value on the individual, the population of DAMCS consists of the elite sub-population, the ordinary sub-population and the developing sub-population. Each sub-population is evolved with different step of Levy flights. The step changes adaptively in accordance with different sub-population and calculation times of fitness. The population of DAMCS consists of the elite sub-population and the developing sub-population by the fitness value of the population after they are spotted. The elite sub-population learns from the best individual to strengthen local search ability, and the developing sub-population evolves with mutation operator of differential evolution algorithm to overstep the local optimum. Each sub-population will change on account of the fitness value in the next iteration, and each sub-population communicates information well. The
simulation indicates the validity and efficiency of DAMCS when compared with other improved cuckoo search algorithms and other swarm intelligence algorithms. In addition, this paper is considering the probability of DAMCS to solve the practical optimization problem.

Acknowledgement

Project supported by Natural Science Foundation of China (11371015), Natural Science Foundation of Sichuan Education (18ZA0469), Talent Foundation of China West Normal University (17YC385) and University Innovation Team (CXTD2015-4).

References


Yi Wen was born in Sichuan Province of China, in 1994. She received the B.S. degree with science from China West Normal University of Sichuan, Nanchong, in 2015. She is currently pursuing the M.S. degree with science from China West Normal University. Her research interests include intelligence computation and algorithm design.
Dazhi Pan was born in Sichuan Province of China, in 1974. He received the B.S. degree in applied mathematics from China West Normal University, Nanchong, China in June 1996 and the M.S. degree in computer applications from Kunming University of Science and Technology, Kunming, China in June 2003 and the Ph.D degree in petroleum engineering computation from Southwest petroleum University, Chengdu, China in January 2012. He is a professor at the school of mathematics and information China West Normal University, China. His current research interests include intelligent computing and algorithm analysis and design.