Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **4**, July **2018**

Opposition-based Learning Differential Ion Motion Algorithm

Yan-Jiao Wang, Chun-Lei Ma

College of Information Engineering
Northeast Dianli University
Jilin 132012, China
hchuang@nuk.edu.tw

Received December, 2017; revised April, 2018

Abstract. *The Ion Motion Optimization algorithm (IMO) is a new physical heuristic global optimization algorithm with simple structure and strong robustness. To further improve the convergence speed and convergence accuracy of IMO, an Opposition-based learning differential ion motion algorithm (OL-IMO) is proposed here. On the one hand, an individual updating method based on the opposition-based learning strategy is proposed, which makes full use of the existing search resources to improve the convergence speed; on the other hand, adding the individual evolutionary information modified random perturbations in the solid phase of renovation, to better balance the algorithm of population diversity and convergence speed. By testing 8 standard test functions, it is shown that OL-IMO algorithm is superior to the IMO algorithm and the other two outstanding optimization algorithms including ADN-RSN-PSO algorithm and MDE algorithm in optimization accuracy, convergence speed and robustness.*
Keywords: PIon Motion Optimization algorithm; Opposition-based learning strategy; The individual evolutionary information.

1. **Introduction.** In recent years, with the popularization of swarm intelligence optimization algorithm, many experts and scholars have produced various effective optimization algorithms. For example, in 2012, the Enhanced parallel cat swarm optimization based on the Taguchi method(EPCSO) proposed by Pei-Wei Tsai and others has achieved good results in solving the problem of numerical optimization [1]. And at 2016, the QUasi-Affine TRansformation Evolutionary (QUATRE) algorithm: A cooperative swarm based algorithm proposed by Zhenyu Meng and others has achieved good results in solving the problem of large-scale optimization [2]. In this paper, the ion motion algorithm is studied. Ion motion algorithm(IMO) [3] is a kind of physical heuristic group intelligence optimization algorithm which simulates the movement rules of anions and cations. It was just proposed by Behzad Javidy et al., Iran Islamic University, in 2015. The test results of 8 standard test functions has shown that compared to the standard Genetic Algorithm(GA) [4], Particle Swarm Optimization algorithm(PSO) [5], Ant Colony Optimization(ACO) [6], Differential Evolution algorithm (DE) [7], and Artificial Bee Colony algorithm(ABC) [8], the IMO algorithm has better optimization effect, which has many advantages, such as fast convergence speed, less parameter setting, simple operation and so on. In summary, the IMO algorithm has become a new star in the field of evolutionary algorithms.

However, like other swarm intelligence optimization algorithms, the IMO algorithm also has low convergence speed and easy to fall into local optimum in the search process. Since it has just been put forward, it has not been paid much attention by scholars in various fields. At present, the performance improvement related articles have not been published, and the theoretical system of the algorithm is not perfect. In view of this, an Opposition-based learning differential ion motion algorithm (OL-IMO) is proposed here. First, an individual update method based on opposition-based learning is proposed, which makes full use of the existing search resources to improve the convergence speed of the algorithm; Secondly, in the solid stage, we introduce different components with evolutionary information to introduce more effective directional information for the generation of the next generation, and enhance the individual's social learning attributes, to better balance the population diversity and speed of convergence of the algorithm. The experiments confirmed that compared to the basic IMO algorithm, all-dimension neighborhood based particle swarm optimization with randomly selected neighbors (ADN-RSN-PSO) [9] and Modified differential evolution with self-adaptive parameters method (MDE) [10], the algorithm proposed in this paper has obvious improvement in convergence speed and convergence accuracy.

The rest of this paper is structured as follows. In the Section 2, we introduce the original IMO algorithm. In the Section 3, we describe the Opposition-based learning differential Ion Motion Optimization algorithm proposed here. In the Section 4, we describe relevant experimental settings and experimental results. In the Section 5, we make a summary of the work.

2. **Standard Ion Motion Optimization algorithm.** The Ion Motion Optimization algorithm divides the ion candidate solutions into two groups, namely, the anion group and the cation group. They perform different evolutionary strategies in the liquid phase and the solid phase, and circulate between the liquid and the solid phase to achieve the purpose of optimizing the ions. The flow diagram of the IMO algorithm is shown in Fig.1.

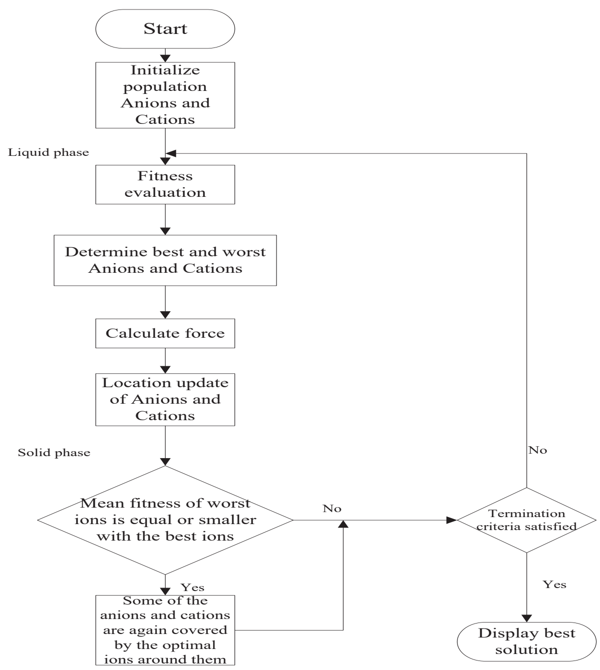

FIGURE 1. General steps of the IMO algorithm.

The key operations of the IMO algorithm in Fig.1 are as follows:

(1) Initialization population

An initial random population consists of NP vectors X_i, $\forall i = 1, 2, ..., NP$, and is randomly generated according to a uniform distribution within the lower and upper boundaries (x_j^L, x_j^U). Each individual is initialized according to the following formula (1).

$$x_{ij}^0 = x_j^L + rand_j \bullet (x_j^U - x_j^L) \tag{1}$$

Where, $rand_j$ is a random number between [0,1], which ensures that each initial solution is different.

(2) Liquid phase

In the liquid phase, the anion group (A) and the cation group (B) are updated according to the following patterns, respectively.

$$A_{i,j} = A_{i,j} + AF_{i,j} \times (Cbest_j - A_j) \tag{2}$$

$$C_{i,j} = C_{i,j} + CF_{i,j} \times (Abest_j - C_j) \tag{3}$$

Where, $i \in \{1, 2, 3, ...NP/2\}$, which is the size of the two groups of ions, $j \in \{1, 2, 3, ...D\}$ representation dimension. Cbest and Abest represent the optimal cation and the optimal anion, respectively (suppose for the minimization problem, the optimal anion, cation is the anion and cation with the lowest fitness value in the entire anion group and the cation group), $AF_{i,j}$ represents the resultant attraction force of anions, and $CF_{i,j}$ indicates the resultant attraction force of cations. The mathematical model is as shown in formula (4) and (5) respectively:

$$AF_{i,j} = \frac{1}{1 + e^{-0.1/AD_{i,j}}} \tag{4}$$

$$CF_{i,j} = \frac{1}{1 + e^{-0.1/CD_{i,j}}} \tag{5}$$

Where $AD_{i,j} = |A_{i,j} - Cbest_j|$, $CD_{i,j} = |C_{i,j} - Abest_j|$, $AD_{i,j}$ is the distance of the i^{th} anion from the best cation in j^{th} dimension, $CD_{i,j}$ calculates the distance of the i^{th} cation from the best anion in the j^{th} dimension.

(3) Solid phase

With the iteration, the ion is gradually gathered near the optimal ion by the gravitational force. To avoid over-concentration of ions to make the algorithm fall into local optimum, therefore set the solid phase, to break the phenomenon of excessive concentration, and provide diversity for the algorithm. The physical process is like this: as the iteration proceeds, the ion motion gradually slows down from the initial intense motion, and gradually the liquid state ions will recrystallize into crystals. IMO simulates this process and proposes a solid phase, with its corresponding pseudo code as shown in Fig.2.

According to the above pseudo code, if both (CbestFit $\geq$ CworstFit/ 2) and (AbestFit $\geq$ AworstFit/2) are met, the solid phases of evolution will take place.

(4) Determination of the terminating conditions

Completion of the solid phase evolution strategy to determine whether to achieve the termination conditions of the algorithm. The termination conditions include the presupposition accuracy, the number of iterations, and so on. If it is reached, the optimal ion is directly output; otherwise, the anions and cations are returned to the liquid phase from the solid phase and continue to be iterated. In such a process, anions and cations are circulated in liquid phase and solid stage, and the optimal solution is gradually obtained with iteration.

```
Algorithm1:Solid phase frame of standard IMO
Input: population NP
Output: New population NP
CbestFit is fitness value of the optimum cation; CworstFit is fitness value of the worst
cation; AbestFit is fitness value of the optimum anion; AworstFit is fitness value of the
worst anion; Φ1andΦ2are random numbers in [−1, 1]; rand() is a function that returns a
random number in [0, 1].
        if (CbestFit>=CworstFit/2&&AbestFit>=AworstFit/2);
            if rand()>0.5
                Ai=Ai+Φ1×(Cbest-1);     %%Mode 1
             else
                Ai=Ai+Φ1×(Cbest);     %%Mode 2
             end
            if rand()>0.5
                Ci=Ci+Φ2×(Abest-1);      %%Mode 1
             else
                Ci=Ci+Φ2×(Abest);     %%Mode 2
            end
            if rand()<0.05
                Re-initializedAiandCi   %%Mode 3
           end
       end
```

FIGURE 2. IMO algorithm solid phase pseudo code

3. Opposition-based Learning differential ion motion algorithm.

The Opposition-based learning differential ion motion algorithm is improved from two aspects of individual regeneration and the generation mode of ions in solid stage, so as to enhance the global optimization ability of IMO algorithm. The details are as follows.

3.1. Improvement of individual renewal mode.

In the IMO algorithm, each individual successively executes the liquid phase and the solid phase to generate new individuals. The new individual is directly substituted for the original individual, regardless of its advantages and disadvantages. This individual renewal model has the following defects: even if the new individual is inferior to the original individual, the original individual can only be abandoned to retain the new individual. Since the original individuals carry some excellent evolutionary information, new individuals are likely to deviate from the original evolutionary direction and let them directly participate in iterative search, which is blindness and will inevitably reduce the convergence speed of the algorithm.

In summary, the new individuals produced through the liquid phase and the solid phase haven't been improved are a key factor affecting the convergence speed of the algorithm. In view of the literature [11] from probability theory has proved that, the probability that the current individual is more far away from the optimal position than its reverse individual is 50%, so the probability that the reverse individual and the current individual are retained to the next generation are equal in one to one competition.While the literature [12] in the experiment prove that opposition-based individuals can effectively increase the diversity of population. For those individuals who have not been improved in this iteration, since their opposition-based individuals also have the ability to get the best solution and they can provide other search locations differed from the most individuals of population. So the participation of opposition-based individuals in the next iteration can provide new evolutionary information for other individuals and supply the diversity of the population to a certain extent. However the opposition-based individuals participating the iteration too much will destroy the original direction of evolution, resulting in reducing the convergence speed of the algorithm. Therefore, the number of opposition-based individuals and the original individuals involved in the iterative search need to be balanced.

To make full use of the existing search resources and improve the convergence speed of the algorithm. This paper puts forward the following improvement measures: comparing the new Xinew generated by the liquid phase and the solid stage with the original Xi, if Xinew is better, Xinew directly replace Xi and participate in iterative search; Otherwise, Xi or the opposition-based solution of Xinew is selected with a certain probability Pm. The corresponding pseudo code is as follows

Algorithm2

Input: population NP
Output: New population NP
f(Xi) is fitness value of the Xi; f(Xinew) is fitness value of the Xinew; rand() is a function that returns a random number in [0, 1]; Pm is 0.5; Xinew is a new individual; $\overline{Xi}$ is the reverse solution of Xinew; Xi is the original individual.

```
for i=1:NP;
  if  f(Xi)<f(Xinew);
      if rand()>Pm;
          Xinew= X̄i;
      else
          Xinew=Xi;
      end
  end
end
```

FIGURE 3. Opposition-based learning pseudo code

Where, the opposition-based solution $\overline{X_i}$ of Xinew is produced as follows: hypothesis $X_{inew} = \{x_1, x_2, ...x_D\}$, it is a point in the D dimensional space, $x_j \in [a_j, b_j]$, $j \in \{1, 2, 3, ...D\}$. The opposition-based point $\overline{X_i} = \{\overline{x_1}, \overline{x_2}, ...\overline{x_D}\}$ corresponding to the X_{inew}, as shown in formula (6), is shown as follows:

$$\overline{x_j} = a_j + b_j - x_j \tag{6}$$

Where, a_j and a_j are the minimum and maximum values for each dimension of the contemporary candidate solutions, respectively.

3.2. **Improvement of ion generation in solid phase.** Through the analysis of the IMO algorithm, it is found that the role of the solid phase is mainly to maintain the diversity of the population. This stage provides three modes for the evolution of ions. Mode 3 is the reinitialization of the individuals, however, the execution probability is very low, so the effect on the convergence rate and the population diversity is minimal. Pattern 2 increases diversity by learning the optimal ion. And mode 1 is to add a (-1,1) random disturbance on the basis of mode 2, and its actual effect is to stretch in the direction of each dimension in the individual dimension. The way of generating decision has its own random unpredictability, specific analysis of the impact of the algorithm are as follows: in the early stage of evolution, population diversity random disturbance has a certain blindness, although it may have some other new individual local information, thus increasing the population diversity. However, since they do not have the information of evolutionary direction, the trend of evolution that originally has certain search direction is damaged to a certain extent, thus affecting the convergence speed of algorithm; In the later stage of evolution, individuals are generally closer to the global optimal solution, and are disturbed by random step size. It is easy to get a relatively large offset, and can not search fine around the optimal solution, so it is difficult to get the global optimal solution. In summary, the search mode 1 in solid stage does not consider the degree of evolution directly, and directly depends on random disturbance to supplement the way of population diversity, which is very unfavorable for algorithm convergence.

Considering that individuals carry information of evolutionary process, we try to modify the search mode 1 with random perturbation in solid stage, and propose the disturbing part as shown in formula (7).

$$\begin{aligned} A_i = A_i + \phi_1 \times (Cbest - 1) + rand() \times (A_i - A_n) \\ C_i = C_i + \phi_2 \times (Abest - 1) + rand() \times (C_i - C_n) \end{aligned} \tag{7}$$

Where, Φ1 and Φ2 are all random numbers between [-1,1], while rand () is the random number between [0,1]. A_n is an anionic individual different from A_i, C_n is a cation individual different from C_i.

By the formula (7) shows that the original random perturbation correction for differential vector, its function includes the following two points: first, the best individual of the population in evolution, compared to exchange information with the original model alone individual and outstanding individual way, due to the non optimal individual by random selection, many the combination will be the introduction of local search more information, so as to increase the diversity of population, reduce the risk of falling into local optimal algorithm; second, compared with the original random disturbance, improve the way of carrying into the information, are more likely to get the new individual than the original outstanding individual, so as to improve the convergence speed of the whole algorithm however, in the late stage of evolution, the population generally tends to the optimal individual, centralized distribution, the vector of the difference between the individual will is very small, thus making disturbance The dynamic range is very small, and it is more beneficial for the individual to carry out fine search at the same time to maintain the diversity of the population, so as to obtain the global optimal solution.

3.3. **Opposition-based learning differential ion motion algorithm.** In this paper, the Opposition-based learning differential ion motion algorithm is proposed. The following steps are as follows:

Step 1 Parameter initialization, including the number of NP, the number of iterations Tmax, the boundary x_j^L and x_j^U of the optimization problem, the dimension D of the optimization problem.

Step 2 The population is initialized randomly, and the population is randomly divided into anion group A1 and cation group C1, according to the formula (1).

Step 3 Calculation of all ion fitness values;

Step 4 The optimal and worst anions and cations were selected.

Step 5 Anion population and cation population perform the liquid phase according to the formula (2) and formula (3) in sections 2 respectively, and the new anionic population A2 and cation population C2 were obtained.

Step 6 The anion population A2 and the cation population C2 perform the solid phase according to the 3.2 part, and the new anionic population A3 and the cation population C3 are obtained.

Step 7 Calculate the fitness of A3 and C3 of anionic population. According to the 3.1 way, we determine the new cation population A1 and anion population C1 from A1, C1, A3 and C3.

Step 8 Determines whether the termination condition is achieved, such as the preset accuracy and the maximum iteration number. It is the algorithm that ends and outputs the best result. Otherwise, it will go back to step 4 and enter the liquid stage again.

4. **Experiment simulation and result analysis.**

4.1. **Standard test functions and performance comparison index.** To fully test the performance of the proposed algorithm, 8 test functions are selected from the CEC2005 benchmark function list, as shown in Table 1. In this, the D represents the number of dimensions, the dimension D=30 of F1∼F6, the dimension D=2 of the F7, and the dimension D=3 of the F8. F1 ∼ F3 is a single peak function, F4∼ F6 is a multi-modal function, and F7 ∼ F8 is a low dimensional function with only a small number of local minimum values.

The performance comparison of test algorithms mainly focuses on 2 aspects:

(1) The accuracy of optimal solution: when the number of iterations reaches the maximum value of Tmax, the accuracy of the optimal solution obtained by the algorithm is the best, the worst, the mean and the variance of the optimal solution obtained in many independent operations.

(2) Convergence speed: before the function evaluation number reached the maximum value Max_FEs, the number of function evaluation times NFEs consumed when the optimal solution obtained by the algorithm reaches the predetermined termination error value Ter_Err (termination error value);

In this paper, the experimental parameters were set as follows: NP = 50 the predetermined error value of the test function Ter_Err is set to 1.0e-10, the maximum number of function evaluations Max_Fes = 100000, each algorithm runs independently for 30 times.

4.2. **Experimental simulation.** To fully test the performance of the proposed method, the proposed algorithm is compared to the basic IMO algorithm and the two global evolutionary algorithms (these two global evolutionary algorithms include: in 2017, Wei Sun et al. proposed the All-dimension neighborhood based particle swarm optimization with randomly selected neighbors (ADN-RSN-PSO) [9]. In 2016, an improved differential evolution algorithm (MDE)based on adaptive parameter method [10]). To ensure the fairness of the comparative experiment, for the same optimization problem, the algorithm uses the same random initial population in a single run independently, to measure the difference in performance optimization quality; and for the same optimization problem, using the same algorithm in multiple independent operation is different random initial population. To measure the algorithm to overcome the initial (external) random interference, ensure to search for a satisfactory solution. In addition, the algorithm uses the same population size NP and termination criteria, which are as follows: NP=50, the maximum number of iterations Tmax=1000.

4.2.1. *Comparison of convergence accuracy.* To investigate the effect of the algorithm on the accuracy of the results, Table 2 gives the results of 30 independent experiments in the 8 different functions of the proposed algorithm and IMO algorithm, ADN-RSN-PSO algorithm and MDE algorithm, including the average accuracy of the optimal solution, the standard deviation, best value, the worst values.

From Table 2, it can be concluded that for ADN-RSN-PSO algorithm, except function F2, F5, and F6, the others solution accuracy is lower than that of MDE algorithm, and F7 and F8 are similar to MDE algorithm. The MDE algorithm can only obtain the theoretical optimal value of F1, F3, and F4. The IMO algorithm can get the theoretical optimal values of functions F4, and the accuracy of all functions is higher than that of ADN-RSN-PSO algorithm. Compared with MDE algorithm, except for function F1 and F3, the accuracy of other functions is higher than MDE algorithm. The OL-IMO algorithm can obtain a theoretical optimal value of 0 of f1-f6, and the remaining two functions are also more accurate than the other three algorithms.

Table 1. Eight benchmark functions

Function	D	Initialization range	Optimum value
F1: Sphere function $f_1(\vec{X}) = \sum_{i=1}^{D} x_i^2$	30	$-100 \leq x_i \leq 100$	0
F2: Schwefel's problem 1.2 $f_2(\vec{X}) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	30	$-100 \leq x_i \leq 100$	0
F3: High conditioned Elliptic function $f_3(\vec{X}) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i$	0 30	$-100 \leq x_i \leq 100$	
F4: Griewank's function $f_4(\vec{X}) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$-600 \leq x_i \leq 600$	0
F5: Rastrigin's function $f_5(\vec{X}) = \sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10)$	30	$-5.12 \leq x_i \leq 5.12$	0
F6: Expanded Scaffer's function $f(x_1, x_2) = 0.5 + \frac{\sin^2 \sqrt{x_1^2+x_2^2} - 0.5}{1+0.001(x_1^2+x_2^2)^2}$ $f_6(\vec{X}) = f(x_1, x_2) + f(x_3, x_4) + ... + f(x_D, x_1)$	30	$-100 \leq x_i \leq 100$	0
F7: Branin function $f_7(\vec{X}) = (x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10$	2	$[-5, 10] \times [0, 15]$	0.398
F8: Hartman's function $f_8(\vec{X}) = -\sum_{i=1}^{D} c_i \exp(\sum_{j=1}^{3} a_{ij}(x_j - P_{ij})^2)$	3	$0 \leq x_i \leq 1$	-3.86278

4.3. **Comparison of convergence speed.** This part mainly investigates the convergence speed of the algorithm, and the parameters of each algorithm remain unchanged. Table 3 calculates the number of function calls (FEs) and the success rate of achieving the preset accuracy when the algorithms converged to the required precision level in the 30 independent experiments.

From Table 3, we can see that for the ADN-RSN-PSO algorithm, only the function f1∼f4 can be successfully converged to the preset precision, however, the successful convergence rate is less than 30%. The MDE algorithm has only function F1, F3, and F4 converges to the preset precision and the successful convergence rate is 100%. The IMO algorithm can be successfully converged for all functions, and the successful convergence rate for other functions is 100% except F5. The successful convergence rate of the OL-IMO algorithm is 100%. And the convergence rate is the fastest.

5. **Conclusions.** To enhance the convergence speed of the IMO algorithm for solving function optimization and convergence precision. This paper proposes an opposition-based update method based on individual learning strategies, make full use of the existing search resources, to enhance the convergence speed; on the other hand, adding the individual evolutionary information correction the random disturbance in the solid phase of

Table 2. The mean, variance, best value and worst values for 30 independent runs

fun		ADN-RSN-PSO	MDE	IMO	OL-IMO
f_1	Mean best value	0.7713	0	3.3187e-39	0
	Standard deviation	2.5951	0	1.7831e-38	0
	Optimal value	0	0	0	0
	Worst value	13.1486	0	9.7713e-38	0
f_2	Mean best value	39.5832	1.2573e+04	3.2087e-35	0
	Standard deviation	209.3445	4.2709e+03	1.7572e-34	0
	Optimal value	0	5.7625e+03	0	0
	Worst value	1.1477e+03	2.4555e+04	9.6244e-34	0
f_3	Mean best value	1.2996e+05	0	6.4440e-35	0
	Standard deviation	6.3692e+05	0	2.7727e-34	0
	Optimal value	0	0	0	0
	Worst value	3.4911e+06	0	1.4592e-33	0
f_4	Mean best value	0.0659	0	0	0
	Standard deviation	0.2046	0	0	0
	Optimal value	0	0	0	0
	Worst value	1.0239	0	0	0
f_5	Mean best value	9.3929	28.7933	0.0534	0
	Standard deviation	39.7363	6.2707	0.2891	0
	Optimal value	1.4207e-07	16.0045	0	0
	Worst value	218.9896	38.7343	1.5839	0
f_6	Mean best value	3.0355	6.0520	1.9849e-12	0
	Standard deviation	2.8440	0.2821	1.0872e-11	0
	Optimal value	2.6623e-04	5.3135	0	0
	Worst value	7.4726	6.5542	5.9547e-11	0
f_7	Mean best value	0.3993	0.3979	0.6397	0.4479
	Standard deviation	0.0024	0	0.3029	0.1470
	Optimal value	0.3979	0.3979	0.3979	0.3979
	Worst value	0.4083	0.3979	1.3586	0.9372
f_8	Mean best value	0	0	-3.7052	-3.7789
	Standard deviation	0	0	0.1185	0.1613
	Optimal value	0	0	-3.8552	-3.8609
	Worst value	0	0	-3.3732	-2.9493

Table 3. Average number of FEs and success rate for 30 independent runs tested on $f_1 - f_6$

Fun	Number of Function Evaluations (Success Rate)			
	ADN-RSN-PSO	MDE	IMO	OL-IMO
f_1	17,300(23%)	70,700(100%)	21,100(100%)	11,100(100%)
f_2	15,300(27%)	100,000(0%)	31,000 (100%)	15,200(100%)
f_3	19,540(17%)	87,400(100%)	30,600(100%)	16,900(100%)
f_4	18,600(20%)	73,600(100%)	19,900(100%)	12,000(100%)
f_5	100,000(0%)	100,000(0%)	26,400(94%)	17,600(100%)
f_6	100,000(0%)	100,000(0%)	50,600(100%)	23,400(100%)

renovation, to better balance population diversity and algorithm convergence speed. By testing 8 standard test functions, it is shown that the Opposition-based learning differential ion motion algorithm (OL-IMO) has obvious advantages over the IMO algorithm and the other two outstanding optimization algorithms so far in solving accuracy, convergence speed and robustness.

Acknowledgments. The authors disclosed receipt of the following financial support for the research, authorship of this article: This work was supported in part by the National Natural Science Foundation of China under grants NO.61501107, and the Education Department of Jilin province science and technology research project of "13th Five-Year" under grants [2016] No.95. And the Project of Scientific and Technological Innovation Development of Jilin No.201750219, No.201750227.

REFERENCES

[1] P. W. Tsai, J. S. Pan, S. M. Chen, et al. Enhanced parallel cat swarm optimization based on the Taguchi method, *Expert Systems with Applications An International Journal*, vol.39, no.7, pp. 6309-6319, 2012.
[2] Z. Meng, J. S. Pan, H. Xu, QUasi-Affine Transformation Evolutionary (QUATRE) algorithm: A cooperative swarm based algorithm for global optimization, *Knowledge-Based Systems*, vol.109, pp.104-121, 2016.
[3] B. Javidy, A. Hatamlou, and S. Mirjalili, Ions motion algorithm for solving optimization problems, *Applied Soft Computing*, vol. 32, no. 3, pp. 72-79, 2015.
[4] D. Beasley, An overview of genetic algorithms: Part 1, *University Computing*, vol. 15, no. 6, pp. 170-181 1993.
[5] G. C. Chen and Y. U. Jin-Shou, Particle Swarm Optimization Algorithm, *Information & Control*, vol. 306, pp. 1369-1372, 2005.
[6] F. Han, Z. X. Zhou, Y. Sun, Reactive power optimization based on the improved dual population ant colony algorithm, *Journal of Northeast Dianli University*, vol. 30, no. 4, pp.48-52, 2010.
[7] A. K. Qin and P. N. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, *Evolutionary Computation, 2005. The 2005 IEEE Congress on. IEEE*, vol. 2, pp. 1785-1791, 2005.
[8] B. Akay and D. Karaboga, A modified Artificial Bee Colony algorithm for real-parameter optimization, *Information Sciences*, vol. 192, no. 1, pp. 120-142, 2012.
[9] W. Sun, A. Lin and H. Yu et al. All-dimension neighborhood based particle swarm optimization with randomly selected neighbors, *Information Sciences*, vol. 405, no. C, pp. 141-156, 2017.
[10] X. Li, Modified differential evolution with self- adaptive parameters method *Journal of Combinatorial Optimization* vol.31, no.2, pp. 546-576, 2016
[11] S. W. Wang, L. X. Ding, C. W. Ding, et al. A Hybrid differential Evolution with Elite Opposition-Base learning, *Journal of Wuhan University (SCIENCE EDITION)*, vol. 59, on. 2, pp. 111-116, 2013.
[12] D. H. Xia, Y. X. Li, J. Zhu, Differential evolution algorithm using orthogonal design opposition-based learning, *Journal of Huazhong University of Science and Technology (National Science Edition)*, vol. 45, no. 5, pp. 23-27, 2017.