

Chaotic System Design of Swarm Intelligent Optimization Algorithm

D. Pan^{1, a}

¹School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, 210094, China

^aComput_Sci@163.com

Keywords: swarm intelligence, hybrid particle swarm; algorithm; optimization design

Abstract. As swarm intelligence algorithm has such problems as poor convergence performance, long search time, and low search efficiency and it is also easy to stagnate at locally optimal solution in the process to solve complex optimization problems, self-adaptation hybrid particle swarm optimization algorithm based on improvement of two-dimensional mapping henon is put forward, which provides the combination mechanism of chaotic mapping and particle swarm with benchmark standard test problem as test function to prove the effectiveness of algorithm. The aim is to elevate the performance of swarm intelligence and to increase its capacity to solve issues in complex optimization problems.

Introduction

Optimization problems are common in the systems of production and living, therefore, it has great theoretical and practical significance in the study of highly effective optimization techniques^[1,2,3]. Swarm intelligent optimization algorithm, which is to solve complex optimization problems, has been widely noticed and applied, as it is simple and easy to be parallel and for intelligent search. To promote the convergence performance of algorithm, it is a shortcut to combine the best characters of different algorithms to form stronger hybrid algorithm. In the process of solving complex problems, single swarm intelligent optimization algorithm presents following shortcomings: low accuracy, weak generalization ability, easy to stagnate at locally optimal solution, and so on. As the swarm intelligent hybrid optimization algorithm comprehensively makes use of the differences and complementarity of various swarm intelligent optimization algorithm, it can realize complementary advantages and information appreciation, and then enhance the overall performance of complex optimization problem^[4,5].

As swarm intelligence algorithm has such problems as poor convergence performance, long search time, and low search efficiency and it is also easy to stagnate at locally optimal solution in the process to solve complicated optimization problems, this paper adopts the chaotic mapping technique in chaotic system, emphasizes on the improvement of chaotic mapping and its organic combination with swarm intelligence algorithm, and provides binarization algorithm of chaotic mapping to analyze the performance of pseudorandom sequence produced in new system. Simulation experiment has solved many objective optimization problems and VRP (vehicle routing problem) as it puts forward new chaotic ant swarm algorithms and provides the mapping subprogram from chaotic image to swarm intelligence algorithm.

Improved chaotic particle swarm optimization algorithm

Chaotic phenomenon exists in nonlinear dynamic system, which has such advantages as being sensible to initial value, general, regular and ergodic^[6]. The most typical chaotic model is Logistic mapping, with the formula is:

$$y_i(t) = \mu y_i(t-1)(1 - y_i(t-1)) \quad (1)$$

In the formula, $y_i \in [0,1]$ is random; t standards for present chaotic iterations; μ is control parameter, with the general value of 4. At this time, the system is totally chaotic.

Particle Swarm Optimization (PSO) is a kind of swarm intelligence optimization algorithm adopting speed position search model, ^[7] in which the optimal solution can be found through the

coordination among individuals, and it can solve some complex optimization problems. Every particle can adjust the flight route in a dynamic way according to its flying experience and swarm flying experience, with the main features of rapid convergence, easy coding and easy to realize.

Suppose in D target solution space, $S = \{P_1, P_2, \dots, P_m\}$ indicates a swarm composed of m potential solutions; the position of particle i in present D target solution space is written as $P_i^d(t) (i=1,2,\dots,m; d=1,2,\dots,D)$; the present flight speed of particle i is written as $V_i^d(t) (i=1,2,\dots,S)$; individual extremum $P_{best}^d(t)$ stands for the best position of particle/self-experience; the global extremum $G_{best}^d(t)$ stands for the best position that all the particles have experienced. In order to seek for the optimal solution, in every iteration, particles renew their positions and speeds through global extremum $G_{best}^d(t)$ and individual extremum $P_{best}^d(t)$, with the expression as following:

$$\begin{cases} V_i^d(t+1) = W \bullet V_i^d(t) + C_1 \bullet R_1(P_{best}^d(t) - P_i^d(t)) + C_2 \bullet R_2(G_{best}^d(t) - P_i^d(t)) \\ P_i^d(t+1) = P_i^d(t) + V_i^d(t+1) \end{cases} \quad (2)$$

In this formula, $V_i^d(t)$, $V_i^d(t+1)$, $P_i^d(t)$ and $P_i^d(t+1)$ stand for the speed and positions of particle t at present moment and next moment; C1 and C2 are learning factors, whose values are between 1.4 and 2.0; R1 and R2 are random numbers in [0,1]; W is inertia weight, which is to balance global search and local search abilities.

In PSO algorithm, parameters of C1, C2, R1 and R2 are key factors influencing PSO algorithm. Standard PSO algorithm is fast in convergence rate, but in the later period, it is slow, with particle swarm having convergence and easy to stagnate into local minimum, which is prematurity phenomenon. However, chaotic optimization algorithm have features of being sensitive to initial value, easy to jump out of local minimum, fast in search, highly accurate in computation, and with high asymptotic convergence. According to this, this paper introduces chaotic optimization design and prematurity judgment and treatment mechanism, and puts forward the improvement of chaotic particle swarm optimization algorithm, so as to enhance the convergence speed of algorithm effectively and to avoid later vibration of algorithm partially.

Chaotic optimization design

The chaotic search idea in ICPSO algorithm is mainly showed in the following five aspects:

(1) Inertia factor W has very important influence on the convergence of chaotic particle swarm optimization algorithm; in order to effectively balance local and global search abilities, elevate algorithm searching performance, and quicken convergence speed, this paper introduces chaos into inertia factor to optimize W and maps to $[W_{min}, W]$, with the adjustment formula as following:

$$W(t) = 4.0W(t-1)(1 - W(t-1)) \quad (3)$$

$$W(t) = W_{min} + (W_{max} - W_{min})W(t) \quad (4)$$

In which, $[W_{max}, W_{min}]$ is the value range of inertia factor, whose range is [0.4, 0.9].

(2) In order to overcome the low efficiency caused from random values, chaos is introduced into random numbers, with the update formula as following:

$$R_i(t) = 4.0R_i(t-1)(1 - R_i(t-1)) \quad (5)$$

In which $R_i(t) \in [0,1], i=1,2$

(3) Learning factors C and C2 stand for the weight of random acceleration terms which pull every particle to the optimal positions; in order to effectively enhance the convergence rate of algorithm, chaos is introduced into learning factors C and C2, with the update formula as following:

$$CW(t) = 4.0C(t-1)(1 - C(t-1)) \quad (6)$$

$$C(t) = C_{\min} + (C_{\max} - C_{\min})W(t) \quad (7)$$

(4) The positions and speeds of chaotic sequence initialization particles are adopted, which not only doesn't change the randomness of PSO algorithm in initialization, but also uses chaos to elevate swarm's diversity and the ergodicity of search.

(5) Introducing chaos into the optimal position that all the particles have searched till now, and taking the optimal position in the produced chaotic sequence as particles' updated position, can effectively avoid particles' position convergence and make insert particles jump out of local optimal values rapidly.

As the speed of particle swarm optimization slows down in the later period, it is easy to fall into the minimum. When the particle swarm meets premature convergence, the particle swarm shows strong convergence, when particles have "assembling" phenomenon. Therefore, the criterion for the premature convergence of particle swarm is that: when the average distance between particles $Dis < \alpha$ (α is preset constant) and the swarm fitness variance $\delta^2 < H$, (H is preset constant), particle swarm will show phenomenon of premature convergence. Average particle distance can be used to show the distribution dispersion degree among individuals, with the formula as following:

$$Dis = \frac{1}{N \cdot L} \sum_{i=1}^N \sqrt{\sum_{d=1}^D (P_{id} - \overline{P_d})^2} \quad (8)$$

In this formula, L is the maximum length of the opposite angles in search space; D is the dimension number of solution space; N is for the size of swarm; P_{id} is the coordinate value in the i dimension of the i particle; $\overline{P_d}$ is for the mean value of all the particles in the i dimension. If the value is small, the swarm is more concentrated; conversely, if the Dis is large, it means the swarm is more dispersed, which means that the swam has good diversity.

Chaotic particle swarm optimization

After chaotic particle swarm optimization (CPSO) is put forward, there have been a lot of attempts in the combination of chaotic strategies and particle swarm algorithm; BingqunRen[8] et al. propose chaotic optimization strategy based on logistic which has been successfully applied in PSO. The chaotic particle swarm algorithm put forward by Amir Hossein et al.[9] is also based on logistic chaotic mapping. Gao L.F.[10] adopts self-adaptive chaotic strategy to revise the amplitude of particle speed elastically, and applies it to typical multipole function optimization successfully. Chaos is the regularity of irregular behaviors in the system. The model that is usually mentioned in chaotic system is insect amount model, which is Logistic model, with the iteration formula as following:

$$x_{n+1} = f(x_n) = \mu \cdot x_n \cdot (1 - x_n), 0 < x_n < 1, n = 1, 2, 3 \dots \quad (9)$$

The experiment shows that when $\mu=4$, the system is in total chaotic state, and Logistic sequence concentrates in the two ends in Chebyshev's inequality (; the other one-dimension chaotic mapping that is usually used is Tent mapping; relatively to logistic mapping, the probability intensity and power spectral density of Tent mapping are more even with smaller auto-correlation, which is more ideal.

Tent mapping has better chaotic features than logistic mapping, but it has several weaknesses and shortcomings: there are unstable periodic points in Tent mapping iteration sequence, and for example, 0.25, 0.50, and 0.75 will all iterate to fixed point 0, with minor cycle, just as 4 periodic phenomenon: (0.2, 0.4, 0.6, 0.8); when Tent mapping is in specific application, in a real number field and under the circumstance of fixed precision realization, compared with continuous system, digitization chaotic system's dynamic characteristics have degraded seriously; because of the deviation brought by limited accuracy, the gained chaotic sequence will have differences from theoretical values, which can be seen in Figure 1.

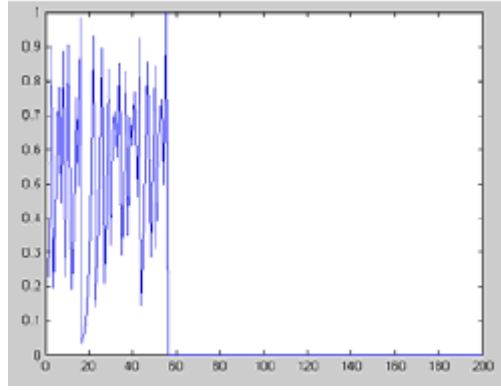


Figure 1 Standard tent mapping produces random sequence

Henon mapping is two-dimensional discrete chaotic mapping, with the iteration formula as:

$$\begin{cases} x_{n+1} = 1 + y_n - ax_n^2 \\ y_{n+1} = bx_n \end{cases} \quad (10)$$

New binaryzation method of chaotic sequence is put forward. Firstly, multiple binary sequences are produced from multilevel quantization chaotic actual values, and exclusive-or operation is taken on sequence to produce ultimate chaotic sequence, with specific algorithm as above; two henon vector x binarization sequences are got; from probability density figure, it can be seen that the probability distribution of old-henon x is not uniform, but the revised new-henon has a uniform probability distribution. Formula (11) is subprogram of chaotic mapping, in which x_i is decision variable, $x_{i\max}$ is the maximum value of decision variable, $x_{i\min}$ is the minimum value of decision variable, and chx_i is chaotic sequence.

$$chx_i = \frac{x_{i\max} - x_i}{x_{i\max} - x_{i\min}} \quad (11)$$

Improved-henon is:

$$x_i = x_{i\max} - chx_i \times (x_{i\max} - x_{i\min}) \quad (12)$$

And then the decision variable x_i is produced; the chaotic variable chx_i can be updated later and new decision variable x_i can be updated.

To judge whether there is premature phenomenon in the later period of algorithm operation, this paper puts forward new judgment standard for prematurity; if

$$abs((P_{id}(t - x_{id}(t)) * (P_{id}(t - 1) - x_{id}(t - 1))) \leq 1e - 6 \quad (12)$$

it shows that the displacement distance of particles is very small, which means the particle's present position is rather close to the historical optimal value, and particle is in a relatively stable state; the algorithm is likely to have prematurity, and needs to adjust chaotic variables to regulate the adaptability of algorithm, that is to add chaotic disturbance quantity in algorithm. Firstly, make use of Improved-henon chaotic mapping subprogram to produce two sets of vector sequences, and use formula 12 to update decision variable r , to replace $r1$ and $r2$ produced by rand function; notice $r1$, $r2$ and x dimension in Improved-henon chaotic mapping, and the formula is used for twice.

The local pheromone updating formula of adding chaotic disturbance quantity is:

$$v_{id}(t + 1) = w \times v_{id}(t) + c_1 h_1 (p_{id}(t) - x_{id}(t) + c_2 h_2 (p_{gd}(t) - x_{id}(t))) \quad (13)$$

If the flight directions of particles for the two times are the same, which means the symbols of recent two times of $pid(t) \times id(t)$ are the same, it indicates that the particle goes close to the individual optimal solution from one side; on the contrary, if the flight directions in the two times are not the same, it indicates that the ion has flied over the individual optimal solution and is swinging around the optimal solution, and at this time, it is necessary to reduce the chaotic regulatory factors properly, $h1 = h1 * 0.6$, to reduce the step size in search and to increase particle search accuracy. In addition, if the value is 0, it means that the particle doesn't move at all in recent two times, which indicates that the speed is close to 0 and prematurity appears; to deal with prematurity, the chaotic parameter is

increased, $h1=h1*1.2$, until to jump out of local optimal solution; the management of $h2$ is the same as $h1$.

Procedure of chaotic particle swarm optimization

(1) The maximum value for the largest initialization iteration steps in initialization is $nc=1000$, with inertia weight as w , and learning factor as $r1=r2=2$.

(2) In $[0,1]$, a D dimension space particle is produced randomly; with the optimization henon chaotic mapping formula 11, chaotic sequence can be gained; with formula 12, N D dimension particles $x_i, i=1,2,\dots,N$ in optimization space (domain of definition) can be gained through chaotic mapping subprogram; similarly, the flight speed of initialized particles can also be got.

For each particle

Calculate its fitness value when it experienced the best position.

If its fitness value is better than the best relevance grade at present (p Best in history),

Update $p(t)$. Set present value as $pbest$.

End for

Set the best $pbest$

Particle is global $gbest$.

For each particle

Adjust particle speed according to formula (1)

If $((P_{id}(t - x_{id}(t)) * (P_{id}(t - 1) - x_{id}(t - 1))) > 1e - 6$

and $abs((P_{id}(t - x_{id}(t)) * (P_{id}(t - 1) - x_{id}(t - 1))) > 1e - 6$ then

Calculate particle speed according to formula.

Else if $((P_{id}(t - x_{id}(t)) * (P_{id}(t - 1) - x_{id}(t - 1))) < 0$

Adjust chaotic parameter $r1=r1*0.6, r2=r2*0.6$, reduce step size in particle search and increase particle search accuracy.

Else

Make premature treatment, and add chaotic parameter control degree to make $l=r1*1.2, r2=r2*1.2$ till it jumps out of the locally optimal solution.

end if

Adjust particle position according to formula.

End for

While maximum iterations or minimum error criteria is not attained.

Simulation experiment and analysis

To verify the effectiveness of algorithm, internationally used benchmark test function put forward by CEC is adopted, and 3 test functions are chosen, which are Rosenbrock, Rastrigin, and Griewank functions, among which Rastrigin and Rosenbrock functions are usually used to test the global convergence of algorithm. Rosenbrock is non-convex function; Rastrigin is multimoding function; Griewank function has numerous local poles, which is usually used in testing the exploratory performance of algorithms, with MatLab7.8 as the test platform and Windows server2003 to program. The experiment parameters: particle number, also known as swarm scale, $M=30$, $wmax=0.95$ $wmin=0.4$; the speed V is limited in $[xmax-xmin/2, -xmax-xmin/2]$, and the mean value is the average after 50 experiments. After a large amount of operation tests and comparison with other algorithms, better results have been gained and the main contrast results are given.

Table 1 is chosen Improved-henonPSO algorithm test function, including formula description, independent variable's value range, and extreme point description information. Table 5.2 shows the results of benchmark standard test functions processed by Improved-HenonPSO particle swarm optimization and previous particle swarm optimization, and the comparison, in which best stands for the optimal solution gained by the algorithm, and mean signifies the average of acquired solution

values. Figure 2 includes function graph and function evolution curve of Improved-HenonPSO, in which red dots are the known optimal solutions of function optimization.

Table 1 Improved-henonPSO test functions

Test functions	Formulas	Search space
Griewank	$f_1(x) = \sum_{i=1}^n (x_i^n / 4000 - \prod_{i=1}^n \cos(x_i / \sqrt{i})) + 1$	$x_i \in [-600, +600]$, if $x_i=0$, minimum value 0 is reached
Rosenbrock	$f_2(x) = \sum_{i=1}^n (100(x_{i+1} - x_i^2) + (x_i - 1))^2$	$x_i \in [-30, +30]$, if $x_i=1$, minimum value 0 is reached
Rastrigin	$f_3(x) = \sum_{i=1}^n (x_i^n - 10 \cos(2\pi x_i) + 10)$	$x_i \in [-5.12, +5.12]$, if $x_i=0$, minimum value 0 is reached

Rosenbrock test issue belongs to continuous and single problem; from the function graphs in Figure 2, 3, and 4, it can be seen that the value interval has a flat trend, which means that the information amount prepared for algorithm is not too much and it is hard for global optimum; it is an excellent function to verify the convergence of algorithm. As for Rosenbrock test function, the bestvalue of Improved-henon chaotic particle swarm is 8.9856e-030, far higher than ordinary PSO and contrast chaotic particle swarm algorithm, with meanvalue as 1.30e-010, which is also better than comparing algorithm. From the function evolutionary figure, it can be seen that its convergence speed keeps at a fast speed. Griewank and Rastrigin belong to complex non-linear multi-peak issue; as both have a large amount of local extreme values, the swarm diversity, global search performance, convergence ability to escape from local extreme values and to avoid prematurity of new algorithm can be tested. The best value of Rastrigi and Improved-henon chaotic particle swarm is 1.77635683940025e-015, which is also far higher than ordinary PSO and comparing algorithm; similarly, its average value is 0.0024881E+00, which significantly outperforms several orders of magnitudes. From the function evolution graph, it can be seen that the convergence speed always keeps at excellent values. The best value for Griewank and Improved-henon chaotic particle swarm is 0.368e-05, far higher than ordinary PSO and comparing algorithm, whose average value outperforms basic PSO algorithm.

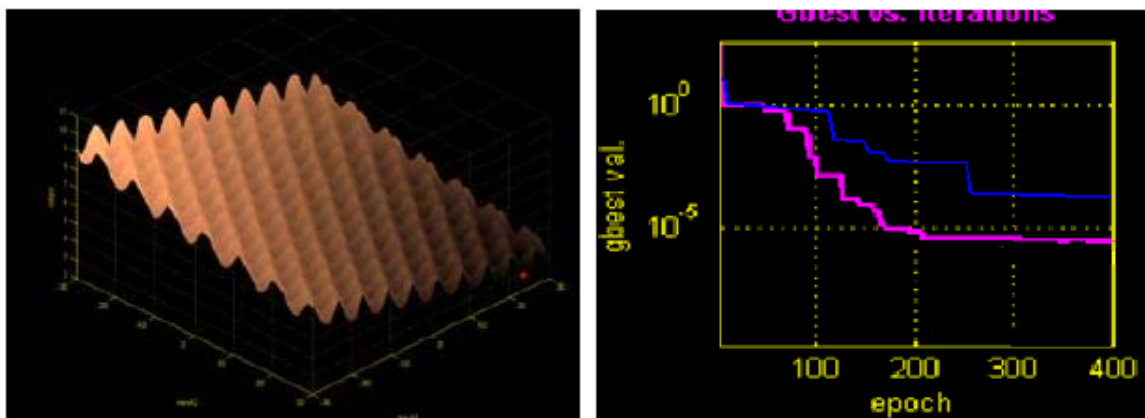


Figure 2 Griewank function graph and function evolution curve

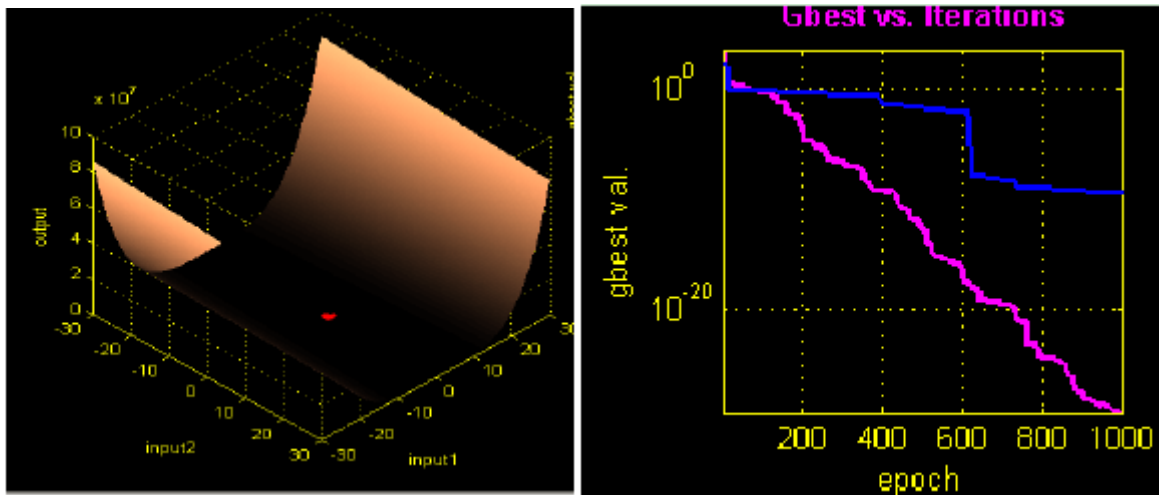


Figure 3 Rosenbrock function graph and function evolution curve

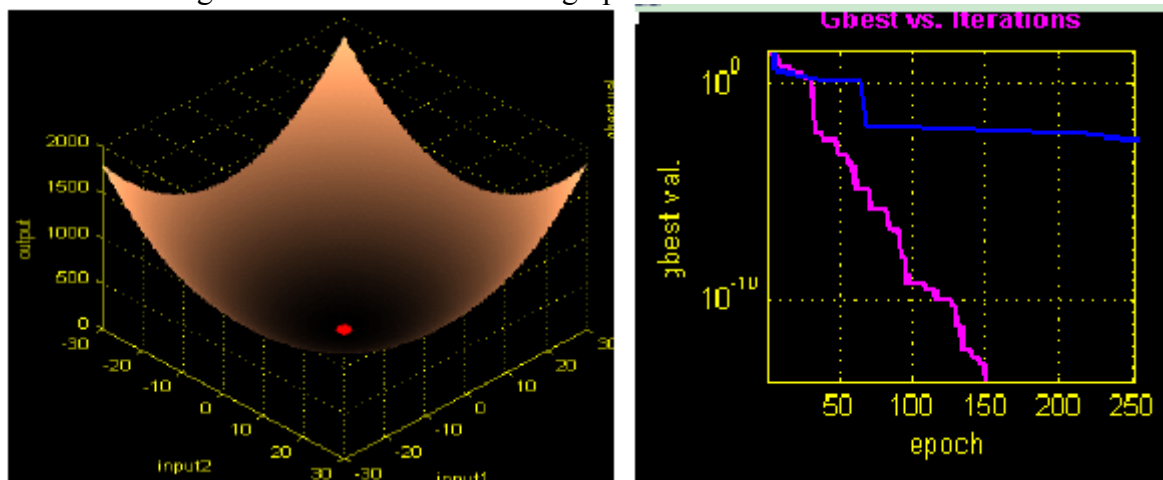


Figure 4: Rastrigin function graph and function evolution curve

In terms of searching precision and successful times, Improved-HenonPSO outperforms basic particle swarms; especially in searching precision and success rate, Improved-HenonPSO is obviously superior to basic particle swarms. Improved chaotic particle swarm has good optimizing ability to ordinary functions and shows high efficiency in the optimizing of Rosenbrock and Rastrigin functions; in terms of the optimal solution, average value, or search time, it is obviously better than standard PSO. As for Griewank, the effect cannot meet the expectation; although good optimal solutions are gained and the average is better than standard PSO, it is not in the ascendant in search time, and it is mainly because that the chaotic pseudorandom sequence consumes part of time, which needs improvements in the future.

Summary

This paper puts forward chaotic particle swarm optimization, improves the complexity and randomness of two-dimensional henon mapping through adding absolute value terms in standard chaotic mapping henon, and proposes new binarization sequence algorithm to improve the random sequence characteristics and probability densities of the x value. Through improving Henon chaotic mapping, initializing particle swarm's position and adjusting particle swarm algorithm's searching speed dynamically, forming new Improved-HenonPSO and adopting standard benchmark test problems to make simulation testing, it shows that for most of the test functions, new algorithm is obviously better than standard PSO in precision and success times, which has superior application prospect.

References

- [1] Li X.J., Shao Z.J., Qian J.X.. "A kind of optimizing model based on autonomous animals: fish algorithm", *System Engineering Theory and Practice*, 2002, 22: 12-21
- [2] Liu Y.M., Zhao Q.S., Sui C.L.. "A particle swarm algorithm based on dynamic neighbors and mutagenic factors", *Control and Decision*, 2010, 25: 29-30.
- [3] Tao Xinmin, Liu Furong, Liu Yu, et al. A kind of particle swarm optimization algorithm with multi-scale and cooperative mutation particles. *Software Journal*, 2012, 2: 18-20.
- [4] Wu Xiaojun, Yang Zhanzhong, Zhao Ming. Uniform search particle swarm algorithm. *Journal of Electronics*, 2011, 39: 1647-1652.
- [5] Zhou Xinyu, Wu Zhijian, Wang Hui, et al. A kind of particle swarm optimization algorithm that elites learn reversely. *Journal of Electronics*, 2013, 41: 1647-1652
- [6] Qin J, Yin Y X, Ban X J. A hybrid of particle swarm optimization and local search for multimodal functions. In *Advances in Swarm Intelligence First International Conference*, Beijing, China, 2010: 589-596
- [7] Bingqun Ren, Weizhou Zhong. Multi-objective Optimization using Chaos Based PSO. *Inform. Technol. J.* 2011, 10: 1908-1916.
- [8] Amir Hossein, Gandomi Gun Jin Yun, Chaos-enhanced accelerated particle swarm optimization. *Commun Nonlinear*. 2013: 327-40
- [9] Gao Leifu. Global optimization algorithm of chaotic elastic particle swarm. *Control and Decision* 2009, 24: 1546-1548.

2. Report

- [1] Karaboga D. An idea based on honey bee swarm for numerical optimization. Engineering Faculty, Computer Engineering Department, Erciyes University, Technical Report-TR06, 2005