Chaotic Inertia Weight in Particle Swarm Optimization

Yong Feng[1], Gui-Fa Teng[1], Ai-Xin Wang[1], and Yong-Mei Yao[2]
[1]*College of Information Science and Technology*, *Agricultural University of Hebei*, *Baoding*, *China*, 071001
[2]*Normal College of Baoding*, *Baoding*, *China*, 071000
E-mail: *fengyong*689@*tom.com, tguifa@heinfo.net, waxwax*346@163.*com*

Abstract

The inertia weight is one of the parameter in Particle Swarm Optimization algorithm. It gets important effect on balancing the global search and the local search in PSO. Basing on the linear descending inertia weight and the random inertia weight, this paper presents the strategy of chaotic descending inertia weight and the strategy of chaotic random inertia weight by introduced chaotic optimization mechanism into PSO. They make PSO algorithm has the characteristics of preferable convergence precision, quickly convergence velocity and better global search ability. The PSO using the chaotic random inertia weight performs especial outstanding comparing with the PSO using random inertia weight, owing to it has rough search stage and minute search stage alternately in all its evolutionary process.

Keywords

Particle Swarm Optimization; inertia weight; chaos

1. Introduction

The inertia weight w is one of the parameter in Particle Swarm Optimization (PSO). It gets important effect on balancing the global search and the local search in PSO. When w is big, particle swarm trend to global search. When w is small, particle swarm trend to local search[1]. Now, the strategies to set inertia weight are the linear descending inertia weight[1], the random inertia weight[2], and the fuzzy adaptive inertia weight[3].

The strategy of linear descending inertia weight sets inertia weight using the following formula[1]:

$$w=(w_1-w_2)\times(MAXiter-iter)/MAXiter+w_2$$

w_1 and w_2 are the original value and the final value of inertia weight. *MAXiter* and *iter* are the maximum iterative time and the current iterative time.

This strategy enhances the performance of PSO. Experiment indicates the iterative effect of former 1500 time is better when inertia weight descend from 0.9 to 0.4. The strategy enhances the convergence velocity of PSO, but it pays out cost: PSO usually gets into the local optimum when it solves the question of more apices function.

The strategy of random inertia weight sets inertia weight using the following formula[2]:

$$w=0.5+0.5\times rand()$$

rand() is a random number in the interval of [0, 1].

Experiment indicates this strategy accelerate the convergence of particle swarm in the early time of the algorithm, and can find quite good result with mostly function.

The strategy of fuzzy adaptive inertia weight sets inertia weight using a fuzzy control system. The fuzzy adaptive inertia weight PSO performs quite excellent on single apex function. But it is hard to find the optimum inertia weight when it solves the question of more apices function. The reason is the control system of the fuzzy adaptive inertia weight PSO can't differentiate the algorithm state is getting into the local optimum or nearing the optimum of a single apex function.

Chaos is a universal phenomena of non-linear dynamics. Chaotic movement has the characteristics of randomicity, ergodicity, regularity. Owing to chaotic movement has these characteristics, chaos may become a available mechanism to avoid getting into the local optimum in searching process. So chaos becomes a new and compellent optimization means. Now it is discovered that the chaotic optimization means have better characteristics of mountain-climbing and getting away from the local optimum. So the chaotic optimization means is better than the stochastic search means[4][5].

Basing on the linear descending inertia weight and the random inertia weight, this paper presents the strategy of chaotic descending inertia weight and the strategy of chaotic random inertia weight by introduced chaotic optimization mechanism into PSO algorithm. Experiment indicates the new strategies is effective and can get better result.

2. Inertia weight PSO

Suppose it has a swarm make up of m particles in a D dimensions search space. The position of the i-th particle is a vector of D dimensions: $X_i=(X_{i1}, X_{i2}, \ldots, X_{iD})^T$, $i=1, 2, \ldots, m$. The fly velocity of the i-th particle also is a vector of D dimensions: $V_i=(V_{i1}, V_{i2}, \ldots, V_{iD})^T$, $i=1, 2, \ldots, m$.

Particles renovate their velocity and position by the following formulae[6]:

$$V_{id}=w\times V_{id}+c_1\times rand()\times(P_{id}-X_{id})+c_2\times rand()\times(P_{gd}-X_{id})$$

$$X_{id}=X_{id}+V_{id}$$

Thereinto, w is a inertia weight coefficient, it relates to its last time velocity; c_1 and c_2 are accelerating factor, they relate to the optimum position oneself undergoing or the particle swarm undergoing; V_{id} is the velocity of particles, $i=1, 2, \ldots, m$, $d=1, 2, \ldots, D$, $V_{id}\in[-v_{max}, v_{max}]$, the maximum velocity v_{max} is a constant to prevent overflow; $p_i=(p_{i1}, p_{i2}, \ldots, p_{iD})^T$ is the current optimum position of the i-th particle, it is called individual extremum; $p_g=(p_{g1}, p_{g2}, \ldots, p_{gD})^T$ is the current optimum position of particle swarm, it is called global extremum; $rand()$ is a random number in the interval of (0, 1).

Every particle renovates their velocity and position used the above formulae in each iterated time, till discovering the optimum result or iterated termination.

3. The chaotic inertia weight

The basic idea of chaotic inertia weight is using a chaotic mapping to set inertia weight coefficient. This paper chooses Logistic mapping to do it. The formula of Logistic mapping is:

$$z=\mu\times z\times(1-z)$$

When $3.57<\mu\leq 4$, Logistic mapping occurs chaotic phenomena. When $\mu=4$, its chaotic result besprinkles the interval of [0, 1]. Figure 1 is a distributing figure in the interval of [0, 1] that Logistic mapping iterate 30000 times. It show the times happening to the interval of [0, 0.1] and [0.9, 1] is very high. The maximum times is more than 1500. The mean times happening to the interval of [0.1, 0.9] is about 200.

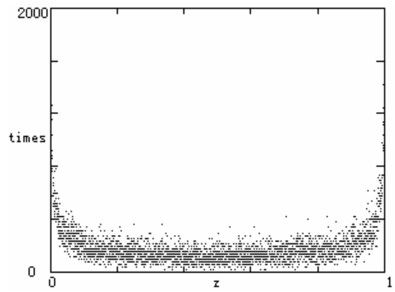


Figure 1. **The distributing figure of Logistic mapping**

The strategies of chaotic inertia weight are given as follows.

(1) The strategy of chaotic descending inertia weight

The strategy of chaotic descending inertia weight sets inertia weight w according to the following steps:

Step 1. Select a random number z in the interval of (0, 1);

Step 2. Make Logistic mapping: $z=4\times z\times(1-z)$;

Step 3. $w=(w_1-w_2)\times(MAXiter-iter)/MAXiter+w_2\times z$

Thereinto, w_1 and w_2 are the original value and the final value of inertia weight, $MAXiter$ and $iter$ are the maximum iterative time and the current iterative time.

(2) The strategy of chaotic random inertia weight

The strategy of chaotic random inertia weight sets inertia weight w according to the following steps:

Step 1. Select a random number z in the interval of (0, 1); Select a random number $rand()$ in the interval of [0, 1];

Step 2. Make Logistic mapping: $z=4\times z\times(1-z)$;

Step 3. $w=0.5\times rand()+0.5\times z$

The two strategies alter the constant item of the linear descending inertia weight and the random inertia weight into "chaotic item". They make inertia weight has chaotic characteristic as well as keep originally varying trend.

We call the PSO used the strategies of linear descending inertia weight, random inertia weight, chaotic descending inertia weight and chaotic random inertia weight as LDIW PSO, RIW PSO, CDIW PSO and CRIW PSO.

4. Experiment

We test the new strategies to verify their performance. Using Benchmark functions to experiment: the swarm's size is 20 in all experiment, every function experiments 500 times independently, every experiment iterates 1500 times, w_1=0.9, w_2=0.4, c_1=c_2=2.0.

Benchmark functions are given as follow:

Sphere:

$$f_1(x)=\sum_{i=1}^{n}x_i^2$$

Rosenbrock:

$$f_2(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$$

Rastrigin:

$$f_3(x)=\sum_{i=1}^{n}[x_i^2-100\cos(2\pi x)+10]$$

Girewank:

$$f_4(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$$

Schaffer f6:

$$f_5(x)=0.5+\frac{(\sin\sqrt{x_1^2+x_2^2})^2-0.5}{[1+0.001(x_1^2+x_2^2)]^2}$$

Benchmark functions' dimensions, the initial range, the criterion and the global optimum result is shown at Table 1.

Table 1. **Benchmark functions**

Function	Di-Men-sion	Initial range	Criterion	The global optimum result
Sphere	30	[−100, 100]	0.01	0
Rosenbrock	30	[−30, 30]	100	0
Rastrigin	30	[−5.12, 5.12]	50	0
Girewank	30	[−600, 600]	0.05	0
Schaffer f6	2	[−100, 100]	0.00001	0

Using appraising indexes as follow to analyze experiment result: Mean Optimum Fitness(MOF), Standard Deviation(SD) and Success Rate(SR). MOF and SD are the mean value and the standard deviation of the optimum fitness results experimenting 500 times independently. SR is the rate of the optimum fitness result in the criterion range experimenting 500 times independently. MOF indicates the precision that the algorithm can get within given iterated times, and it reflects the algorithm's convergence velocity, SD reflects the algorithm's stability and robustness; SR reflects the algorithm's global search capability. The experiment result is shown at Table 2 and Table 3.

Table 2. **The experiment of LDIW PSO and CDIW PSO**

function	PSO	MOF	SD	SR
Sphere	LDIW	0.000137	0.000050	100
	CDIW	0.000092	0.000016	100
Rosenbrock	LDIW	49.379089	9.875818	79.8
	CDIW	44.305058	8.861012	99.6
Rastrigin	LDIW	40.739613	8.147923	78.2
	CDIW	40.044561	8.028912	83.6
Girewank	LDIW	0.026612	0.003322	87.4
	CDIW	0.014773	0.002955	96.2
Schaffer f6	LDIW	0.009235	0.001847	7.4
	CDIW	0.007732	0.001546	22

Table 3. **The experiment of RIW PSO and CRIW PSO**

function	PSO	MOF	SD	SR
Sphere	RIW	0.000227	0.000069	100
	CRIW	0.000087	0.000017	100
Rosenbrock	RIW	178.706706	35.741341	14.6
	CRIW	37.090110	11.618022	99.4
Rastrigin	RIW	45.554978	9.110996	67.2
	CRIW	40.267957	8.053591	91.8
Girewank	RIW	0.076999	0.094359	64
	CRIW	0.016616	0.003323	98.2
Schaffer f6	RIW	0.017707	0.003541	10.4
	CRIW	0.009211	0.001842	24.4

MOF indicates the new strategies can get better optimum fitness result, and they have preferable convergence precision and quickly convergence velocity. SD indicates the deviation of the optimum fitness result that the new strategies get is comparatively less, it indicates the new strategies have better stability and robustness. SR indicates the new strategies have better global search ability, they can get away from the local optimum and search out the global optimum result.

5. The chaotic optimization characteristic in the CRIW PSO

The CRIW PSO performs especial outstanding comparing with the RIW PSO. We print the values of each particle's every dimension in a figure and get the figure of the evolutionary trend of particle swarm. The evolutionary trend of the CRIW PSO in a experiment is shown at Figure 2. Figure 2 indicates the particle swarm has distinct chaotic optimization characteristic: the algorithm search in a biggish range at its initial stage, this stage is called "rough search"; the algorithm search in a lesser range at its later stage, this stage is called "minute search".

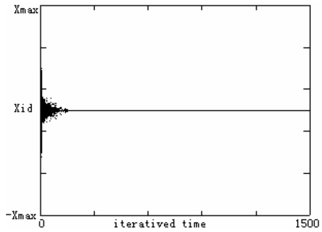
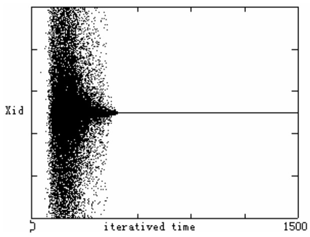
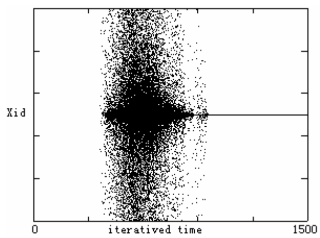


Figure 2. **The evolutionary trend of the CRIW PSO (The swarm's size is 20, the dimension is 30)**

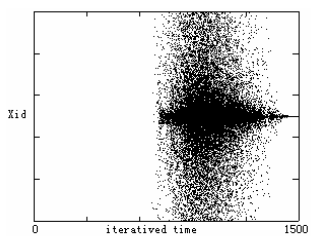
We zoom in the figure of the evolutionary trend of the CRIW PSO to 10^{10}, 10^{30} and 10^{50} multiples (see in Figure 3). All of images are similar in Figure 3. It indicates it has rough search stage and minute search stage alternately in all evolutionary process of the CRIW PSO algorithm. So the algorithm can quickly get to the position near the global optimum to prosecute more aborative search till discovering the optimum result or iterated termination. We think it is the reason why the CRIW PSO performs outstanding.



a. **Zoom in 10^{10} multiples**



b. **Zoom in 10^{30} multiples**



c. **Zoom in 10^{50} multiples**

Figure 3. **The Zoomed evolutionary trend of the CRIW PSO**

6. Conclusion

This paper introduces chaotic optimization mechanism into particle swarm optimization algorithm, and presents the strategies of chaotic descending inertia weight and chaotic random inertia weight. These new strategies utilize the merits of chaotic optimization. They make inertia weight has chaotic characteristic as well as keep originally varying trend. They make PSO has preferable convergence precision, quickly convergence velocity and better global search ability. The CRIW PSO performs especial outstanding comparing with RIW PSO, owing to it has rough search stage and minute search stage alternately in all its evolutionary process.

References

[1] R. C. Eberhart, Y. Shi. "Comparing inertia weight and constriction factors in particle swarm optimization", Proceeding Congress on Evolutionary Computation 2000, San Diego, pp84-88

[2] Y. H. Shi, R. C. Eberhart, "Empirical Study of Particle Swarm Optimization", Proceeding Congress on Evolutionary Computation 1999, Piscataway, pp:1945-1949

[3] Y. Shi, R. C. Eberhart, "Fuzzy Adaptiveive Particle Swarm Optimization", Proceeding Congress on Evolutionary Computation 2001, pp: 101-106, Seoul Korea

[4] T. Ikeguchi, K. Sato, M. Hasegawa, "Chaotic Optimization for Quadratic Assignment Problems", IEEE International Symposium on Circuits and Systems 2002, pp: 469-472

[5] Y. Hayakawa, A. Marumoto, Y. Sawada, "Effects of the Chaotic Noise on the Performance of a Neural Netwok Model for Optimization Problems", Phys ical Rev. E, 1995, pp: 2693-2696

[6] Y. Shi, R. C. Eberhart, "A Modified Particle Swarm Optimization", International Conference on Evolutionary Computation 1998, Anchorage Alaska, pp: 69-73