International Core Journal of Engineering Volume 5 Issue 12, 2019
ISSN: 2414-1895 DOI: 10.6919/ICJE.201911_5(12).0012

A Research of Simplified Particle Swarm Optimization Algorithm with Weight and Learning Factor

Xiaoyue Cao [a,*] , Xuxiu Zhang, Yumeng Sun, Zhenfu Kuai

School of Dalian Jiaotong University, Dalian 116021, China

[a]1971103922@qq.com

Abstract

In the view of the traditional particle swarm optimization algorithm is easy to get into the slow speed convergence and long search time when it solved the high-dimensional mathematical problems. In order to solve the problem that the independent adjustment of inertia weight and learning factor weakens the unity and intelligence of pso, the particle state factors is added to inertia weights and learning factors, A pso combining inertial weights and learning factors is presented.Based on the method of linear change of inertia weight, this algorithm adds the variable operator of inertia weight, which makes the inertia weight change dynamically with the iteration stage and aggregation of particles.And on the basis of simplifying particle swarm optimization, asynchronous transformation learning factor is added to make the position update formula dynamically updated according to the search situation of particles .Finally, the improved algorithm is compared with the existing algorithm on four test functions, and it is proved that the improved algorithm has obvious improvement on the optimization accuracy, iteration speed and convergence success rate.

Keywords

particle swarm optimization ,adaptive, Inertia weight, simple particle swarm optimization.

1. Introduction

Particle swarm optimization (PSO) is a kind of stochastic evolutionary optimization algorithm based on swarm intelligence proposed by Kennedy and Eberhart in 1995[1].PSO algorithm is quickly accepted and applied in electromagnetic optimization [2-3], parameter optimization [4-5] and other fields due to its advantages of fewer parameters to be adjusted, fast convergence speed and high accuracy.Due to the shortcoming of particle swarm optimization (pso), many improvements have been made[6-8].There are also many improvements to particle swarm optimization and the improvement of inertia weight and learning factor is the most simple and quick.Shi et al. introduced inertia weights that decrease linearly with the number of evolution[9], which greatly improved the running speed and search ability of PSO algorithm.There are linear and nonlinear methods to improve the inertia weight[9-10].There are many similar selection strategies for learning factors[11-12].All these methods improved theglobal and local search capability of the algorithm.

However, the independent adjustment of parameters such as inertia weight and learning factor weakens the intelligence of the algorithm. Based on the analysis of the improvement of particle swarm optimization algorithm in the above literatures, this paper links inertia weight and learning factor through particle state factor, and gives a simplified particle swarm optimization algorithm integrating weight and learning factor. The improved algorithm is compared with the linear particle swarm

ISSN: 2414-1895 DOI: 10.6919/ICJE.201911_5(12).0012

optimization (LPSO) and the simplified particle swarm optimization (SPSO) to prove the effectiveness of the improved algorithm.

2. Particle state factor and particle aggregation factor

First initialize a swarm of particles $x_i = (x_{i1}, x_{i2}, \cdots x_{in})$ in each iteration, particles update its own speed and position through the individual extreme value *pbest* and the global extreme value *gbest*, the updating formula is shown in equations (1) and (2):

$$v_{id}^{k+1} = w * v_{id}^{k} + c_1 * rand_1 * \left(pbest_{ij}^{k} - x_{ij}^{k}\right) + c_2 * rand_2 * \left(gbest_i^{k} - x_{ij}^{k}\right) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \quad (2)$$

c_1 and c_2 are velocity factors, and usually they are both 2. $rand_1$ and $rand_2$ are random numbers distributed between [0,1]. k is the current number of iterations. v_{id} is the speed of particles. w is the inertia weight. In each iteration, the particle compares the current value with the optimal value to update the global optimal value. Therefore, the value of global optimal value e is determined by the change of individual extreme value, and the relationship between global and individual also reflects the current state of particles. In the minima problem, the individual extremum is less than the global optimal value, which is also less than the average value. The average reflects the average mass of all the particles. Define $F = f_i / f_{avg}$ as the particle state factor.

f_i is the current fitness value of the particle. f_{avg} is the average fitness value of the particle, and its mathematical formula is defined as $f_{avg} = n^{-1}\sum_{i=1}^{n} f_i \cdot F$ reflects the current search state of the particle, it means that the larger F is, the further the current particle is from the optimal solution. The smaller F is, the closer the current particle is to the optimal solution.

Definition $\sigma^2 = \sum_{i=1}^{n}\left(F - f_{best} \bullet f_{avg}^{-1}\right)$

f_{best} is the optimal fitness value of the particle, that is the global optimal value. Therefore, the formula of particle aggregation factor is defined as follows:

$$S = \sigma^2 / \sigma_{\max}^2 \text{, among it } \sigma_{\max}^2 = \max_{1 \le i \le n}\{\sigma^2\}$$

According to the formula, $S \in [0,1]$. We call S the aggregation factor. To a certain extent, S reflects the aggregation of particles, which reflects the diversity of particles. The larger S is, the farther the particle is from the optimal value and the more dispersed it is. On the contrary, the smaller S is, the more concentrated the particles are.

3. An improved strategy of integrating weights and learning factors

3.1 The change of inertia weight

Shi[7] et al proposed a method of linear decline, the experiment shows that although LDW improves the speed and accuracy of function optimization, the method of inertia weight changing with the number of iterations is only effective near the optimal solution. On the basis of LDW inertia weight change formula, the following formula is proposed:

$$w = \partial \times [w_{\max} - (w_{\max} - w_{\min}) \times R / R_{\max}] \quad (3)$$

∂ is an inertia weight variation operator whose size is determined by the aggregation of particles. Combined with s-shaped curve, namely Sigmoid function, the change formula proposed is as follows: $1/(1+e^{-1/s})$.

ISSN: 2414-1895 DOI: 10.6919/ICJE.201911_5(12).0012

The reciprocal function sigmoid decrements in the domain and early fast decline which is conducive to fast search and determine values closer to $pbest$.Late slow decline which beneficial to search carefully after particle concentration and improve the accuracy of the algorithm. The larger s is, the farther the particle is from the optimal value and the more dispersed it is. At this time, should be taken as a smaller value to enhance the local search ability. On the contrary, the smaller s is, the more concentrated the particles are, ∂ should be given a larger value to enhance the global search capability.

Specific adjustment strategies of ∂ are as follows:

$$\partial = \begin{cases} \partial_{\max} & S \le H_1 \\ \partial_{\min} + (\partial_{\max} - \partial_{\min}) \times (1 / (1 + e^{-\frac{1}{s}})) & H_1 < S < H_2 \\ \partial_{\min} & S \ge H_2 \end{cases}$$

H_1 and H_2 are the threshold values of particle aggregation degree, and meet $H_1 < H_2$,when the threshold value is exceeded, set $\partial_{\min}$ and $\partial_{\max}$.

3.2 Improvement of position update formula

The simplified particle swarm optimization eliminates the velocity term and reduces the second order of the differential equation to the first order, which greatly speeds up the search speed of the algorithm and makes the algorithm more simple and efficient.

On the basis of this, some scholars put forward the $pbest$ and $gbest$ in the linear combination substitution formula using the global optimal value and the local optimal value. Its improvement is publicized as follows:

$$X_i^{t+1} = wX_i^t + c_1 r_1 \left((p_{best} + g_{best})/2 - X_i^t \right) + c_2 r_2 \left((p_{best} - g_{best})/2 - X_i^t \right) \tag{4}$$

c_1 and c_2 are fixed values. In order to enhance the unity and intelligence of the evolutionary process when adjusting inertia weights and learning factors, in this paper, a factor of inertia weight is taken as the main parameter of the change of learning factor, particle state factor, which strengthened the relationship between inertia weight and learning factor and to better adapt to complex nonlinear changes. Specific adjustment strategies for learning factors c_1 and c_2 are as follows:

$$\begin{aligned} c_1 &= c_{1ini} + (c_{1ini} - c_{1fin}) \bullet F \\ c_2 &= c_{2ini} - (c_{2fin} - c_{2ini}) \bullet F \end{aligned} \tag{5}$$

c_{1ini} and i c_{2ini} are the initial values of c_1 and c_2 . c_{ifin} and c_{2fin} represent the termination values. c_1 is a learning factor that determines the ability of particles to "social cognition". c_2 is the learning factor that determines the particle's ability to "recognize itself". F measures the search state of the particle during the search process. When the problem is the minimum problem, if F is larger, it means that the particle is farther away from the optimal solution. In this case, c_1 should take a larger value and c_2 should take a smaller value. This approach makes the particle learn from the best of the self to the best of the society, and it strengthen the global search ability; If F is smaller, it means that the particle is closer to the optimal solution. At this time, c_1 should be a smaller value while c_2 should be a larger value, so that the particle can be closer from the optimal learning of society to the optimal learning of self and strengthen the local search ability.

3.3 The flow of algorithm

The improved algorithm flow is as follows:

ISSN: 2414-1895 DOI: 10.6919/ICJE.201911_5(12).0012

Step1: initialize the particles, make their values constrained within the specified range, and make each particle have position vector X_i ;

Step2: calculate the fitness value of the particle, set f_i as the fitness value of the current position of the particle, and calculate that f_{avg} is the average fitness value.

Step3: compare the fitness value of the particle with all the positions of the particles it passes through, and take the better one as the current best position, denoted as $pbest$;

Step4: compare the fitness value of the particles with the positions all the particles have passed through, and take the better one as the global best position, denoted as $gbest$;

Step5: if the algorithm reaches the maximum number of iterations, execute step 8; otherwise, execute step 6;

Step6: update the position formula of all particles successively according to formula (4),calculate the inertia weight according to formula (3),Calculate the learning factor according to formula (5),calculate the fitness value;

Step7: increase the number of iterations by one and execute step 3;

Step8: reach the maximum number of iterations and output $gbest$.

4. Simulationg test and analysis

4.1 Experimental design

In order to verify the effectiveness of the improved algorithm, SPSO and LPSO are compared with the DSPSO proposed in this paper. Four test functions are compared in terms of convergence accuracy, convergence speed and dimension.

In the experiment, the population size was 30. The number of iterations is 1000.Theinertia weight in LPSO decreases linearly from 0.9 to 0.4.Parameter setting of DSPSO: $w_{\max}=0.9$, $w_{\min}=0.4$.

This experiment will test and analyze from the following three aspects:

(1) the experiment calculated the average value of the four functions running 50 times in 3, 10 and 30 dimensions under the three algorithms, and analyzed the convergence accuracy of each dimension.

(2) compare the convergence rates of the four algorithms through the average fitness evolution curve.

(3) under the condition of fixed convergence accuracy and specified number of iterations, the success rates of the four functions under the three algorithms were compared.

Test functions are as follows:

Tab.1 The information of test function

函数名	函数表达式	解空间	最优值位置	最优值
Schwefel2.22	$f_1(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	$[-10,10]^D$	$(0)^D$	0
Ackley	$f_2=-20\exp\left(-20\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	$[-32,32]^D$	$(0)^D$	0
Griewank	$f_3(x)=\frac{1}{400}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$[-600,600]^D$	$(0)^D$	0
Penalized1.1	$f_4(x)=\frac{\pi}{n}\left\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right]+(y_n-1)^2\right\}+\sum_{i=1}^{n}u(x_i,10,100,4)$	$[-50,50]^D$	$(0)^D$	0

ISSN: 2414-1895 DOI: 10.6919/ICJE.201911_5(12).0012

4.2 Analysis of test results

4.2.1 Comparison of algorithm convergence accuracy

In this paper, the four functions were run for 50 times in 3 dimensions, 10 dimensions and 30 dimensions respectively to obtain the average value. The optimal value and average value are shown in table 2 below.

From table 2, on a macro level, DSPSO is superior to SAPSO and LPSO in terms of functions, f_1, f_3, f_4 no matter in 3, 10 or 30 dimensions. For the function f_2 ,the optimal value of all three algorithms is the same, but the average value is different: The accuracy of SPSO and LPSO in 3 and 30 dimensions is about 10-6,however, the optimal value and average value of DSPSO are all 8.8818E-16 in 3, 10 or 30 dimensions. This means that DSPSO has good stability for processing f2, a function used for many local optimal values and independent of independent variables.

As the dimensions of the four functions increase, the functions become more complex. The search accuracy of SPSO and LPSO gradually decreased, indicating that the search ability of SPSO and LPSO for high-dimensional functions was gradually weakened. In the case of DSPSO, in the function, the value of DSPSO is always 8.8818e-16, reflecting the stability of DSPSO. In the function f_4 , the accuracy of DSPSO varies from 10^{-46} to 10^{-6}.but it is far better than other two kinds of algorithms in the same dimension. In conclusion, the DSPSO algorithm has a good effect on search accuracy and solving function problems of high latitude.

a is the average, b is the optimal value

Tab 2. The results of function test

函数	维数	理论值		LPSO	SPSO	DSPSO
*f*1	3	0	a	1.1891e-69	3.0384e-40	1.867e-217
			b	3.58627e-67	4.4292e-32	1.424e-209
	10	0	a	0.033706	3.0988e-17	1.3376e-26
			b	0.0956932	2.4708e-15	2.9167e-26
	30	0	a	4.5745	2.4585e-05	4.3675e-07
			b	2.435554	0.6028495	4.6572e-06
*f*2	3	0	a	8.8818e-16	8.8818e-16	8.8816e-16
			b	5.8565e-06	2.9282e-06	8.8818e-16
	10	0	a	8.8818e-16	8.8818e-16	8.8818e-16
			b	0.0002941	0.0001475	8.8818e-16
	30	0	a	8.8818e-16	8.8818e-16	8.8818e-16
			b	5.7355e-06	2.8677e-06	8.8818e-16
*f*3	3	0	a	0	0	0
			b	0	0	0
	10	0	a	0.5436	0.35857	0.024737
			b	1.617722	0.63957	0.085173
	30	0	a	1.384	0.64322	1.9069e-07
			b	5.31914	0.82533	0.019103

ISSN: 2414-1895 DOI: 10.6919/ICJE.201911_5(12).0012

f_4	3	0	a	2.3358e-31	2.3358e-31	3.4126e-52
			b	2.3358e-31	3.0728e-30	2.3358e-46
	10	0	a	5.1957e-32	6.0994e-28	4.7116e-32
			b	6.7122e-09	0.12442	4.7551e-32
	30	0	a	7.3992e-09	0.73519	1.1869e-08
			b	0.02073	2.680078	4.7930e-06

4.2.2 Comparison of algorithm evolution speed

The fitness evolution curve of the four functions is as follows:

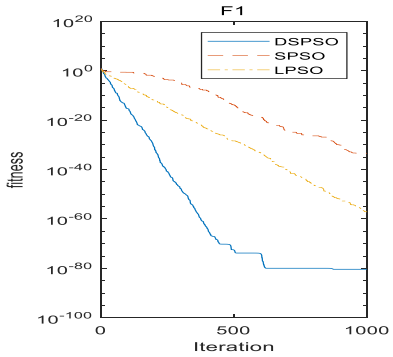

Fig. 1. fitness evolution curve of function f_1

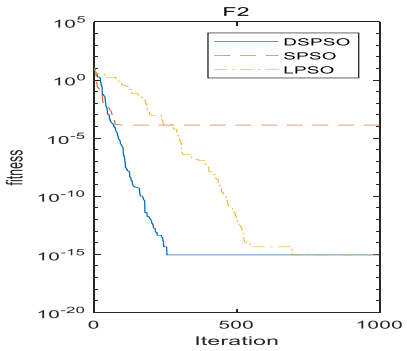

Fig. 2. fitness evolution curve of function f_2

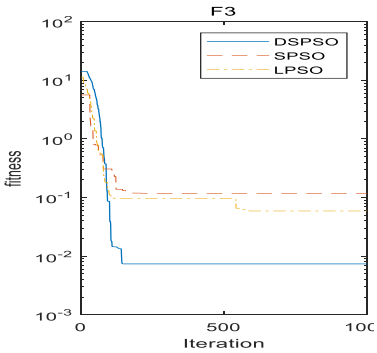

Fig. 3. fitness evolution curve of function f_3

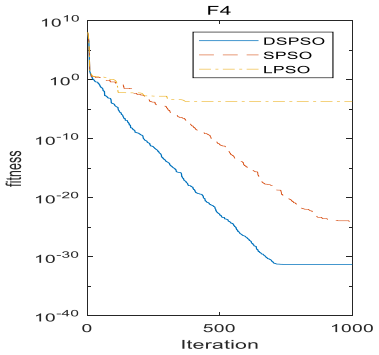

Fig. 4. fitness evolution curve of function f_4

In figure 1, since the DSPSO algorithm completed the iteration around 600 times, SPSO and LPSO did not complete the iteration. The optimal iteration value after the program is run is shown in figure 5.

```
:iton found by DSPSO is :  4.4033e-81
:iton found by SPSO is :  4.0006e-34
:iton found by LPSO is :  6.4185e-58
```

Fig.5. The optimal value of function f1

It can be seen from figure 1 and figure 5 that DSPSO completed iteration in 600 with an accuracy of10-81, much higher than that of SPSO and LPSO.SPSO and LPSO did not complete the iteration within 1000 times, indicating that the search speed of DSPSO is much higher than the other two algorithms. In figure 2, SPSO fell into local optimization early, and the final convergence accuracy of DSPSO and LPSO was the same. However, DSPSO completed the iteration at 200, while LPSO successfully completed the iteration at 650, and the speed was lower than that of DSPSO. In figure 3, both speed and accuracy are better than the other two algorithms. In figure 4, SPSO fell into local

ISSN: 2414-1895 DOI: 10.6919/ICJE.201911_5(12).0012

optimization early. In the two algorithms DSPSO and LPSO, it can be seen from the slope that the search speed of DSPSO is better than LPSO, and the precision is slightly better than LPSO.

4.2.3 Comparison of algorithm success rate

Each function is simulated 50 times, the number of iterations is 1000.When each function does not converge when it reaches the set number of iterations, the convergence is considered to fail.The success rate of convergence is analyzed in the following table.

Tab.3 .The success rate of function

函数	算法	成功收敛次数	成功率(%)	函数	算法	成功收敛次数	成功率(%)
f_1	SPSO	25	50	f_3	SPSO	22	44
	LPSO	33	66		LPSO	33	66
	DSPSO	30	60		DSPSO	47	94
f_2	SPSO	7	14	f_4	SPSO	9	18
	LPSO	19	38		LPSO	48	96
	DSPSO	50	100		DSPSO	50	100

f_1 is unimodal function, f_2、 f_3、 f_4 is bimodal functions. As can be seen from the above table, DSPSO algorithm has no great advantages over SPSO and LPSO when dealing with single-peak function problems like f_1 .However, on the problem of f_2 , f_3 and f_4 multi-peak, the success rate of DSPSO algorithm is significantly higher than that of the other two algorithms. In conclusion, the improvement of APSO algorithm has obvious advantages in dealing with multi-peak complex function problems.

5. Conclusion

After experimental analysis, simplified particle swarm optimization algorithm with weight and learning factor is proposed. The inertia weight and learning factor are connected by the particle state factor, which enhances the unity and intelligence of the algorithm. The local and global searching ability of the algorithm is improved and the precision searching speed of the algorithm is accelerated. In the future algorithm research, group algorithm should be applied to practical engineering, such as motor control parameter adjustment, to reduce overshot.

Acknowledgments

Fund project:

Funded by the national science and technology support program(2015BAF20B02);

National natural science foundation of China(61471080, 61201419);

Liaoning natural fund guiding plan(1553737612631);

Supported by the China scholarship council(201608210308).

References

[1]Eberhart, R., Kennedy, J.. A new optimizer using particle swarm theory[P]. Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on,1995.

[2]Liu Z H , Wei H L , Zhong Q C , et al. GPU Implementation of DPSO-RE Algorithm for Parameters Identification of Surface PMSM Considering VSI Nonlinearity[J]. IEEE Journal of Emerging and Selected Topics in Power Electronics, 2017:1-1.

ISSN: 2414-1895 DOI: 10.6919/ICJE.201911_5(12).0012

[3]Liu Z H , Zhang J , Zhou S W , et al. Coevolutionary Particle Swarm Optimization Using AIS and its Application in Multiparameter Estimation of PMSM[J]. IEEE Transactions on Cybernetics, 2013, 43(6):1921-1935.

[4]Liu Z H , Wei H L , Zhong Q C , et al. Parameter Estimation for VSI-Fed PMSM based on a Dynamic PSO with Learning Strategies[J]. IEEE Transactions on Power Electronics, 2016:1-1.

[5]Liu Z H , Li X H , Zhang H Q , et al. An Enhanced Approach for Parameter Estimation: Using Immune Dynamic Learning Swarm Optimization Based on Multicore Architecture[J]. IEEE Systems, Man, and Cybernetics Magazine, 2016, 2(1):26-33.

[6] KENNEDY J .Stereotyping: improving particle swarm performance with cluster analysis ［C］/ / Proceedings of the 2000 Congress on Evolutionary Computation.Piscataway: IEEE,2000,2: 1507 - 1512.

[7]LIU B,WANG L,JIN Y-H,et al.Improved particle swarm optimization combined with chao［J］.Chaos, Solitons and Fractals, 2005, 25(5) : 1261 - 1267.

[8]SHI Y,EBERHART R C.Fuzzy adaptive particle swarm optimization［C］/ / Proceedings of the 2001 IEEE Congress on Evolutionary Computation. Piscataway: IEEE, 2001: 101 - 106.

[9]Shi, Y., Eberhart, R.. A modified particle swarm optimizer[P]. Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, 1998.

[10] CHATTERJEE A ,SIARRY P .Nonlinear inertia variation for dynamic adaption in particle swarm optimization ［J］.Computers ＆ Operation Research,2006,33(3) : 859 - 971.

[11] SUGANTHAN P N. Particle swarm optimiser with neighbourhood operator ［C］/ / CEC 99: Proceedings of the 1999 Congress on Evolutionary Computation.Piscataway: IEEE,1999,3: 1958 - 1962.

[12] RATNAWEERA A,HALGAMUGE S.Self-organization hierarchical particle swarm optimizer with varying acceleration coefficients ［J］. Evolutionary Computation, 2004, 8(3) : 240 - 255.