The Novel Non-linear Strategy of Inertia Weight in Particle Swarm Optimization

Li Li [1], Bing Xue [1], Ben Niu[1], Yujuan Chai [2], Jianhuang Wu [3]

[1] *College of Management, Shenzhen University, Shenzhen, Guangdong 518060, China*

[2] *Faculty of Science, McMaster University, Hamilton, Ontario L8S4L8, Canada*

[3]*Shenzhen Institute of Advanced Integration Technology, Chinese Academy of Sciences/ The Chinese University of Hong Kong, Shenzhen 518055, China*

drniuben@gmail.com

***Abstract*: Inertia weight is one of the most important adjustable parameter of particle swarm optimization (PSO). The proper selection of inertia weight can prove a right balance between global search and local search. In this paper, two novel PSOs with non-linear inertia weight based on the tangent function and the arc tangent function are provided, respectively. The performance of the proposed PSO model is compared with standard PSO with linearly-decrease inertia weight. The experimental results demonstrated that our proposed PSO model is better than standard PSO in terms of convergence rate and solution precision.**

Keywords-Particle swarm optimization, inertia weight, tangent function, arc tangent function.

I. INTRODUCTION

The particle swarm optimization (PSO)[1], [2] is a population-based global optimization method proposed by Kennedy and Eberhart, which motivated by the group organism behavior such as bee swarm and bird flock. Compared with other evolutionary computation techniques such as genetic algorithms (GA), PSO is easy in implementation and there are few parameters to adjust, and it has faster convergence rate[3]–[6]. PSO has been successfully applied in science and engineering [7], [8].

As a new algorithm, PSO still has many disadvantages. For instance, it show significant performance in initial iterations, however, the particles are more and more familiar and the swarm loses its diversity along with the developing of the computation. So there may be premature convergence and it is hard to escape the local optimal. Among the adjustable parameters of PSO, the inertia weight is the most important one[9], [10], and lots of investigations have been undertaken to provide the improved ways of the inertia weight to enhance the performance of PSO, including the linearly-decrease inertia weight (LIW)[11], the nonlinearly-decrease inertia weight (NIW)[12]–[14], the random inertia weight (RIW)[15], and so on. In this paper, we analyze the features of PSO and the importance of the inertia weight. Based on the traits of the tangent function and the arc tangent function, two new non-linear strategies about the inertia weight are produced. In order to illustrate the effectiveness and performance of the two strategies for optimization problems, a set of four benchmark functions are used.

978-1-4244-3867-9/09/$25.00 ©2009 IEEE

II. STANDARD PARTICLE SWARM OPTIMIZATION

In PSO, each potential solution is called a bird or particle with no weight and no volume. The *i*th particle flies in the n dimension search space to find the optimization. There is a vector $x_i=(x_{i1},x_{i2},\dots,x_{in})$ presenting the position of the *i*th particle, where $x_{id}\in[l_d,u_d]$, $d\in[1,n]$, l_d,u_d are the lower and upper bounds of the *d*th dimension. The velocity for the *i*th particle is represented as $v_i=(v_{i1},v_{i2},\dots\dots,v_{in})$, which controls the distance and the direction when *i* is flying and it is clamped to a maximum velocity $v_{\max}$ specified by the problem to be solved. Moreover, the best previous position of the *i*th particle is individual best called *Pbest*. The best one of all the *Pbest* is colonial best called *Gbest* denoting the best previous position of the swarm. The system is initialized with a population of random solutions, and based on the *Pbest* and *Gbest*, the algorithm searches for the optimization by updating generations according to the following formulas:

$$V_{id}(t+1)=wV_{id}(t)+c_1\cdot rand()\cdot(p_{id}-x_{id}(t))+c_2\cdot rand()\cdot(p_{gd}-x_{id}(t)), \quad (1)$$

$$x_i(t+1)=x_i(t)+V_i(t+1), \quad (2)$$

where t means algorithm is going on the *t*th generation. c_1 and c_2 are set to constant value, which are normally taken as 2. *rand*() is random value, uniformly distributed in [0, 1]. p_{id} presents the *Pbest* while p_{gd} presents the *Gbest*. w is inertia weight, which controls the influence of previous velocity on the new velocity, and it can make a balance between the global search and the local search: Global search performance is good with large while a small facilitates the local search.

III. THE NOVEL NON-LINEAR INERTIA WEIGHT PSO

Based on the researches on w, it has been proved there will be a faster convergence rate with a larger w, but the precision of the result can not be guaranteed. While a smaller one can get more precise result, but the convergence rate is too slow and the algorithm may get into the local optimal. So a proper variation of w can improve the performance of PSO. During the past studies, we tried to introduce monotone increasing or decreasing strategy to update w .

In the tangent function $y = \tan(x)$, the result y increases along with the independent variable x , and the speed of increase also increases. When $x = 0.875$, $y = 1$. According to these features, we can use the tangent function to build a new strategy of the w . After a large scale of experiments, the final equation for the w is:

$$w(t) = (w_{start} - w_{end}) * \tan(0.875 * (1 - (\frac{t}{t_{max}})^{k_1})) + w_{end} , \quad (3)$$

where w_{start} is the initial value of the w , which is also the largest value and normally $w_{start} = 0.9$, w_{end} is the final value of the w , which also is the smallest one and normally $w_{end} = 0.4$. t_{max} is the maximum number of iterations. According to the Eq.(3), w is nonlinearly-decrease along with the increase of the iteration. The algorithm facilitates global search in the initial iteration, so the particles can fly around the total search space quickly, then the local search become stronger. The new strategy enhances the capability of the algorithm. avoids premature convergence and escapes the local optimal. There is a coefficient 0.875 in the Eq.(3) to guarantee the w distributed in [0.4, 0.9]: When $t = 1$, $w(t) = w_{start} = 0.9$ and when $t = t_{max}$, $w(t) = w_{end} = 0.4$.

There is a control variable k_1 , which can control the smoothness of the curve that reflects the relationship between the w and t . Figs 1-3 show the three different curves respectively when $k_1 = 0.2$, $k_1 = 1$, $k_1 = 3$. It can be found that: when $k_1 = 0.2$, the functions between the w and t is convex function. When $k_1 = 1$, it is almost a linear one leaning to concave. and when $k_1 = 3$, it is a concave function.

A multimode Griewank function is employed to choose the best k_1. k_1 is confined in [0.1~2.0] in the experiments. The experimental results are listed in Table I.

From Table I, it shows that when k_1 is during [0.4~0.6] and [1.4~1.7], the mean and the standard deviations of the function values are both stable. Through other experiments, it has been proved that the algorithm produces better result and convergence rate when $k_1 = 0.6$ for many other functions. So k_1 should be chose during [0.4~0.6]. In the following experiments in this paper TANW is used to represent PSO algorithm proposed above and $k_1 = 0.6$.

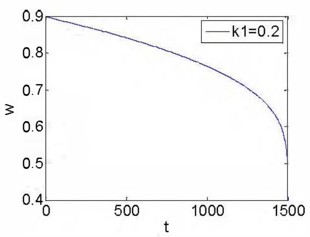

Fig. 1. $k_1 = 0.2$

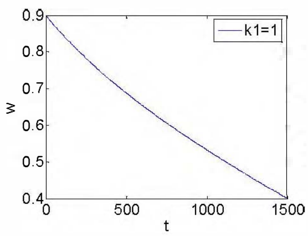

Fig. 2. $k_1 = 1$

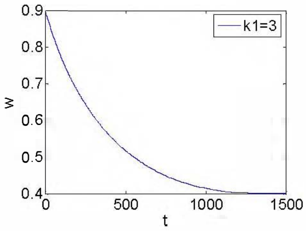

Fig. 3. $k_1 = 3$

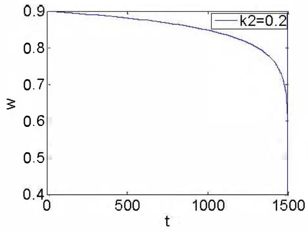

Fig. 4. $k_2 = 0.2$

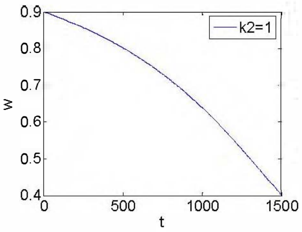

Fig. 5. $k_2 = 1$

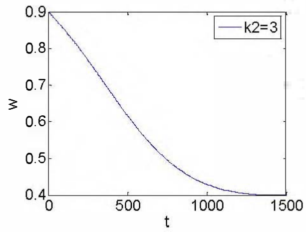

Fig. 6. $k_2 = 3$

TABLE I RESULTS OF THE GRIEWANK WITH DIFFERENT k_1

k_1	Mean	Std	k_1	Mean	Std
0.1	0.0294	0.0276	1.1	0.0302	0.0230
0.2	0.0300	0.0223	1.2	0.0355	0.0194
0.3	0.0329	0.0248	1.3	0.0413	0.0291
0.4	0.0266	0.0191	1.4	0.0230	0.0212
0.5	0.0258	0.0278	1.5	0.0289	0.0273
0.6	0.0254	0.0207	1.6	0.0263	0.0200
0.7	0.0315	0.0307	1.7	0.0261	0.0198
0.8	0.0300	0.0343	1.8	0.0334	0.0219
0.9	0.0301	0.0235	1.9	0.0257	0.0253
1.0	0.0264	0.0267	2.0	0.0280	0.0256

For the same reason, the arc tangent function $y = \arctan(x)$ is also an increasing one, however the speed of increase is slower and slower. When the independent variable $x = 1.56$, the result $y = 1$. As the tangent function and the arc tangent function are reciprocal functions, they show the familiar law and also must be a little different. So the arc tangent function also can be used to build a new improvement of w , as the following equation shows:

$$w(t) = (w_{start} - w_{end}) * \arctan(1.56 * (1 - (\frac{t}{t_{max}})^{k_2})) + w_{end} , \quad (4)$$

where w_{start}, w_{end}, t, t_{max} denotes the same meanings as in the Eq.(3). w is decreasing along with t . The difference is that the speed of decrease is slower in prior period and faster in later period. The w is also not too small in later period, so it guarantee the convergence rate in prior period and the exploration in later period. The algorithm can escape the local optimal effectively. There is also a coefficient 1.56 in the Eq. (4) to guarantee the w distributed in [0.4, 0.9]: When $t=1$, $w(t) = w_{start} = 0.9$, and when $t = t_{max}$, $w(t) = w_{end} = 0.4$.

Like the Eq.(3), there is a control variable k_2, which can control the smoothness of the curve that reflects the relationship between the w and t . Figs 4-6 show the three different curves respectively when $k_2 = 0.2$, $k_2 = 1$, $k_2 = 3$. It can be found that: when $k_1 = 0.2$, the function between the w and t is convex function. When $k_1 = 1$, it is almost a linear one leaning to convex. when $k_1 = 3$, it is a concave function. Compared Figs1-6, the two functions about the w and t both are from the convex function to concave one along with the increase of the control variable, and the former one is faster. The shape of two kinds of curves is different, too.

The experiments about the multimode function Griewank were done to choose the best k_2 confined in [0.1~2.0]. The experimental results (i.e., the mean and the standard deviations of the function values found in 20 runs) are listed in Table II.

In Table II, when k_2 is during [0.4~0.7], the mean and the standard deviations of the function values are both stable. So k_2 should be chose during [0.4~0.7]. In the following experiments in this paper ATW is used to represent the improved PSO based on this strategy and $k_2 = 0.4$.

TABLE II RESULTS OF THE GRIEWANK WITH DIFFERENT k_2

k_2	Mean	Std	k_2	Mean	Std
0.1	0.0280	0.0280	1.1	0.0453	0.0629
0.2	0.0273	0.0256	1.2	0.0352	0.0284
0.3	0.0331	0.0251	1.3	0.0292	0.0379
0.4	0.0245	0.0202	1.4	0.0354	0.0388
0.5	0.0270	0.0232	1.5	0.0384	0.0373
0.6	0.0247	0.0243	1.6	0.0538	0.0707
0.7	0.0245	0.0251	1.7	0.0745	0.0850
0.8	0.0287	0.0232	1.8	0.0617	0.0769
0.9	0.0304	0.0343	1.9	0.0748	0.1130
1.0	0.0262	0.0206	2.0	0.1779	0.2043

IV. EXPERIMENTAL STUDY

A. Test Functions and Parameters Setting

To illustrate performance of the novel ways, four nonlinear benchmark functions that are commonly used in evolutionary computation literature [16]-[19] were performed, and also compared with the performance of improved PSO based on a linearly-decrease inertia weight (LIW), which is the most widely used nowadays, and the expression for LIW is showing in Eq.(5). The four test functions are listed in Table III.

$$w(t) = w_{start} - \frac{w_{start} - w_{end}}{t_{max}} \times t, \quad (5)$$

In every experiment, the w in the three methods (TANW、ATW and LIW) are all during [0.9, 0.4], that is $w_{start} = 0.9$, $w_{end} = 0.4$. $c_1 = c_2 = 2.0$, the population size is 40, the allowable error $\sigma = 1e-80$, and $t_{max} = 1500$. A total of 50 runs for each experimental setting are conducted.

B. The Result and the Analysis

The results of the four functions are listed in Tables IV- VII, the mean relative performance generated by three algorithms are shown in Figs 7-10.

TABLE III BENCHMARK FUNCTIONS AND PARAMETERS SETTING

Function	Dim	Search space	v_{max}
Sphere	20	(-100,100)	100
Rosenbrock	20	(-30,30)	30
Rastrigrin	20	(-10,10)	10
Griewank	20	(-600,600)	600

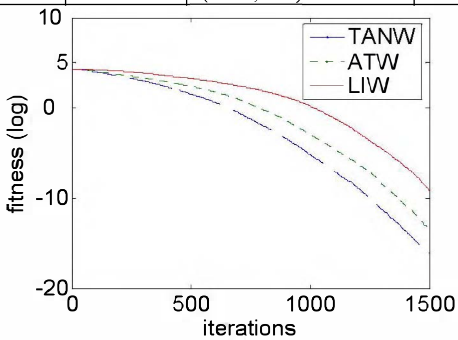

Fig. 7. Sphere function

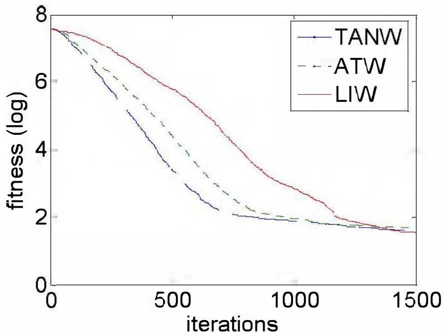

Fig. 8. Rosenbrock function

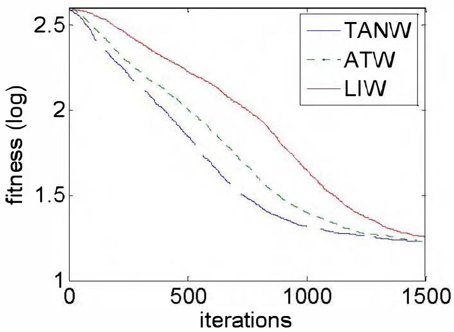

Fig. 9. Rastrigin fuction

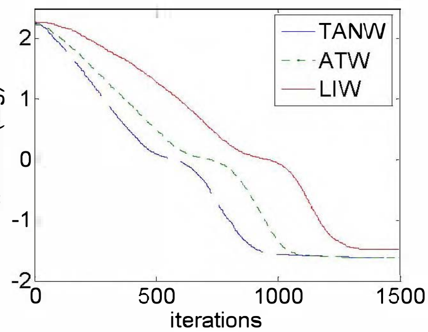

Fig. 10. Griewank function

TABLE IV THE RESULT FOR SPHERE OSENBROCK FUNCTION

Algorithm	Max	Min	Std	Mean
TANW	2.6552e-015	1.2535e-020	3.7727e-016	9.0940e-017
ATW	5.0653e-013	3.4536e-017	8.0530e-014	3.2708e-014
LIW	9.7600e-009	4.8377e-012	1.6531e-009	6.8240e-010

TABLE V THE RESULT FOR ROSENBROCK FUNCTION

Algorithm	Max	Min	Std	Mean
TANW	263.8481	0.2125	55.0677	41.0477
ATW	248.3628	1.8195	49.6680	48.9274
LIW	567.3387	4.4772	107.4373	70.1539

TABLE VI THE RESULT FOR RASTRIGIN FUNCTION

Algorithm	Max	Min	Std	Mean
TANW	35.8185	6.9647	5.8282	16.9156
ATW	28.8538	6.9640	5.3089	16.9652
LIW	33.8585	6.9649	5.8284	18.0666

TABLE VII THE RESULT FOR GRIEWANK FUNCTION

Algorithm	Max	Min	Std	Mean
TANW	0.0787	0	0.0208	0.0240
ATW	0.0811	5.7732e-015	0.0205	0.0239
LIW	0.1052	9.9886e-011	0.0256	0.0328

From the tables and the figures above, we can discovery that:

1) For the simplest unimodal Sphere function, the results generated by TANW is the most robustness (the smallest standard deviations) and the most precision (the smallest mean fitness value). At the same time, it can be concluded that the result got by ATW is worse than by TANW, and better than by LIW. The two improved algorithms own much faster convergence rate than LIW. In the Fig. 7, the three curves show the similar shape, and it can reflect the features of the unimodal function's optimization.
2) For the non-convex and morbid unimodal Rosenbrock function, the values in Table V indicated that ATW found the most robustness (the smallest standard deviations) results, and TANW found the most precision (the smallest mean fitness value) ones. On the whole, they both outperformed LIW. Moreover, the ANTW is most effective in the convergence rate.
3) Rastrigin function is non-linear and multimodal, and it has lots of local optimal values. From the Table VI, it can be observed that the results generated by ATW is the most robustness (the smallest standard deviations). LIW and TANW got the approximate effectiveness in the robust. TANW produced the most precision (the smallest mean fitness value) results, and ATW produced the second precision ones. The two improved algorithms presented much faster convergence rate than LIW.
4) Griewank function is a typical multimodal one. To optimize it can test the global search ability of the algorithm. For this function, the Table VII and Fig. 10 can show that the TANW and ATW improved the robustness, the convergence rate and the precision of the result. TANW generated the fast convergence rate solution, and ATW did better in the robustness and the precision of the result.
5) From all above tables and figures, we can conclude that the convergence rate of TANW is faster than ATW, but in the later period, the exploration ability of TANW is worse. The convergence ability of ATW is weak in prior period, but can escape the local optimal more effectively. From the unimodal function to the complicated multimodal function, ATW performances better and better. TANW works well in unimodal function like Sphere, as it owns fast convergence and need less generation to get the best solution. On the other hand, ATW is fit for optimizing the multimodal function like Griewank because of the strong ability of escaping the local optimal in later period. From the figures, we can found the two new improved algorithms show the similar shape curves in most cases. The phenomenon may relate to that the tangent function and the arc tangent function are reciprocal functions.

V. CONCLUSION AND FUTUER WORK

This paper presents two novel PSO algorithms with non-linear inertia weight based on the tangent function and the arc tangent function. The performance of them is evaluated by the experiments on four representative instances. They provide better quality solutions, and it is more efficacious compared with PSO algorithm with a learning decreasing inertia weight.

Future work is focused on optimizing the performance of TANW and ATW. TANW should have a stronger ability to escape the local optimal in later period, and ATW should have a faster convergence rate. In addition, extensive study of the applications in more complex practical optimization problems is necessary to fully investigate the properties and evaluate the performance of TANW and ATW.

ACKNOWLEDGMENT

This work is supported by Guangdong Natural Science Foundation(Grant no. 9451806001002294), Shenzhen-Hong Kong Innovative Circle project (Grant no. SG200810220137A), Project 801-000021 supported by SZU R/D Fund, and National Natural Science Foundation (Grant no. 60803108).

REFERENCES

[1] J. Kennedy and R. C. Eberhart, "A new optimizer using particle swarm theory", in *Proceedings of the 6th International Symposium on Micromachine and Human Science*, Nagoya, Japan, pp.39--43, 1995.

[2] J. Kennedy and R. C. Eberhart, "Particle swarm optimization", in *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia, Vol. IV, pp.1942–1948, 1995.

[3] R. C. Eberhart, and Y. Shi "Particle swarm optimization: developments, applications and resources", in *proc. IEEE Congress on Evol Comput, Piscataway*, NJ, pp.81-86, 2001.

[4] M. Clerc and J. Kennedy, "The particle swarm: explosion stability and convergence in a multi-dimensional complex space", *IEEE Transactions on Evolutionary Computation*, vol. 6, pp.58–73, 2002.

[5] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization", *Evolutionary programming VII: proceedings of the seventh annual conference on evolutionary programming*, San Diego CA, Springer-Verlag, Berlin, Vol. 1447, pp.611–616, 1998.

[6] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization", in *Proceedings of Congress on Evolutionary Computation*, Washington, DC, Vol. 3, pp.1945–1949, 1999.

[7] T. Sousa, A. Silva and A. Neves, "Particle swarm based data mining algorithms for classification tasks", *Parallel Computing*, Vol. 30, pp.767–783, 2004.

[8] K. P. Wang, L. Huang, C. G. Zhou and W. Pang, "Particle swarm optimization for traveling salesman problem", *Proceedings of International Conference on Machine Learning and Cybernetics*, pp.1583–1585, 2003.

[9] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimizations", *Lecture Notes in Computer Science*, pp.591-600, 1998.

[10] L. P. Zhang, H. J. Yu, S.X. Hu, "Optimal choice of parameters for Particle Swarm Optimization", *Journal of Zhejiang Unverisity*, pp.28-534, 2005.

[11] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer", *Proc IEEE Int Confon Evolutional Computation. Anchorage*, pp.69-73, 1998.

[12] L. Wang and X. K. Wang, "Modified particle swarm optimizer using non-linear inertia weight", *Computer Engineering and Applications*, China, pp. 47-48, 2007.

[13] A. Chatterjee and P. Siarry, "Nonlinear inertia weight variation for dynamic adaptation in pattiele swarm optimization", *Computers & Operations Research*, pp.859-871, 2006.

[14] L. H. Rong and Y. L. Gao, "A Particle Swarm Optimization Algorithm with the Strategy of Nonlinear Decreasing Inertia Weight", in *the papers of Shangluo College*, China, December 2007.

[15] F. van den Bergh, "An Analysis of Particle Swarm Optimizer", *South Africa: Department of Computer Science, University of Pretoria*, 2002.

[16] B. Niu, Y. L. Zhu, X. X He, H. Wu, "MCPSO: A Multi-Swarm Cooperative Particle Swarm Optimizer", *Applied Mathematics and Computation 185*, pp.1050-1062 , 2007.

[17] K. E Parsopoulos and M. N. Vrahatis, "UPSO-A Unified Particle Swarm Optimization Scheme", *Lecture Series on Computational Sciences*, pp. 868-873, 2004.

[18] J. J. Liang, P. N. Suganthan, A. K. Qin and S. Baskar, " Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions", *IEEE Transactions on Evolutionary Computation 10*, pp. 281-295, 2006.

[19] T. Peram , K. Veeramachaneni, and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization", In *proceeding of Swarm Intelligence Symposium, IEEE Press*, Indiana, USA, pp.174-181, 2003.