Improved Particle Swarm Optimizers with Application on Constrained Portfolio Selection

Li Li[1], Bing Xue[1], Lijing Tan[2], and Ben Niu[1,3,*]

[1] College of Management, Shenzhen University, Shenzhen 518060, China
[2] Measurement Specialties Inc Shenzhen 518107, China
[3] Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China
drniuben@gmail.com

Abstract. Inertia weight is one of the most important adjustable parameters 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, a novel PSOs with non-linear inertia weight based on the arc tangent function is provided. The performance of the proposed PSO models are compared with standard PSO with linearly-decrease inertia weight using four benchmark functions. The experimental results demonstrate that our proposed PSO models are better than standard PSO in terms of convergence rate and solution precision. The proposed novel PSOs are also used to solve an improved portfolio optimization model with complex constraints and the primary results demonstrate their effectiveness.

Keywords: Particle swarm optimization, inertia weight, arc tangent function, portfolio optimization.

1 Introduction

Particle swarm optimization (PSO) [1, 2] is a population-based global optimization method proposed by Kennedy and Eberhart, which is 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 shows 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 optimum. There are few parameters to adjust in the PSO, and 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

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2010, LNCS 6215, pp. 579–586, 2010.

inertia weight (NIW) [12], the random inertia weight (RIW) [13], etc. In this paper, a new non-linear strategy on the inertia weight is proposed. To illustrate the effectiveness and performance of the strategy for optimization problems, a set of four benchmark functions and an improved portfolio optimization model are used.

2 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 $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ 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 $v_i = (v_{i1}, v_{i2}, \ldots\ldots, v_{in})$, which controls the distance and the direction 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 algorithm searches for the optimization by updating generations according to the following formulas:

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

$$x_i(t+1) = x_i(t) + V_i(t+1), \tag{2}$$

where t means that 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.

3 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 previous past studies, we tried to introduce monotone increasing or decreasing strategy to update w.

In the arc tangent function $y = \arctan(x)$ is an increasing function. however the speed of increase is slower and slower. When the independent variable $x = \pi/2$, the result $y = 1$. 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 is:

$$w(t) = (w_{start} - w_{end}) * \arctan(\pi/2 * (1 - (\frac{t}{t_{\max}})^k)) + w_{end}, \tag{3}$$

where w_{start} is the initial value of the w and w_{end} is the final value, which also is the smallest one. $t_{\max}$ is the maximum number of iterations. According to the Equation (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. w is also not too small in later period, so it guarantee the convergence rate in prior period and the exploration in later period.

There is a control variable k, which can control the smoothness of the curve that reflects the relationship between the w and t. The experiments show that: when $k = 0.2$, the function between w and t is convex function. When $k = 1$, it is almost a linear one leaning to convex. When $k = 2$, it is a concave function.

Table 1. Results of the Griewank with different k

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

The experiments about the multimode function Griewank were done to choose the best k 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 1.

In Table 1, when k is during [0.4~0.7], the mean and the standard deviations of the function values are both stable. So k 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 is set as 0.4.

4 Experimental Study

4.1 Test Functions and Parameters Setting

To illustrate performance of our proposed method, four nonlinear benchmark functions that are commonly used in evolutionary computation literature [14-16] were performed, and also compared with that of improved PSO based on a linearly-decrease inertia weight (LIW). The four test functions are listed in Table 2.

In every experiment, the w in the two methods (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.

Table 2. 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

4.2 The Result and the Analysis

The results of the four functions are listed in Table 3, the mean relative performance generated by three algorithms are shown in Figs 1-4.

The data in Table 3 show that proposed method (ATW) can obtain more precise results (smaller mean of the function) and the stronger robustness (smaller standard deviations) for all of the four functions than LIW. As seen from the figures, ATW is with the faster convergence rate and able to get the best solution.

The results also indicated that the non-liner inertia weight (ATW) performs better than linear one (LIW).

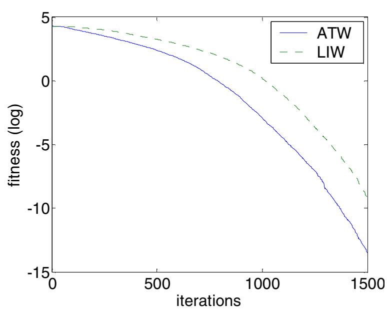

Fig. 1. Sphere function

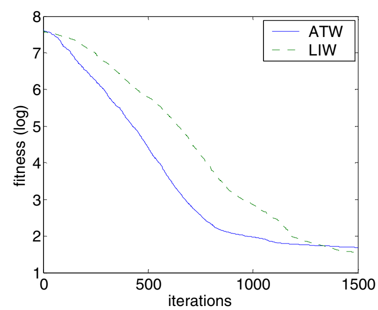

Fig. 2. Rosenbrock function

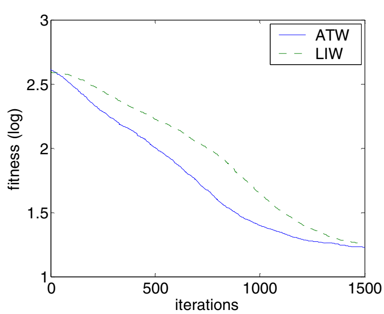

Fig. 3. Rastrigin fuction

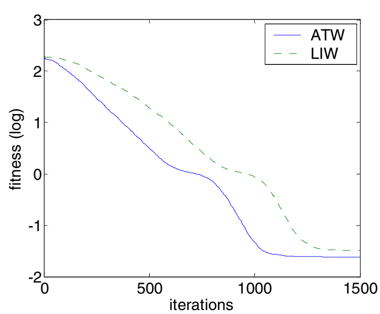

Fig. 4. Griewank function

Table 3. The result for four functions

Function	Algorithm	Max	Min	Std	Mean
Sphere	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
Rosenbrock	ATW	248.3628	1.8195	49.6680	48.9274
	LIW	567.3387	4.4772	107.4373	70.1539
Rastrigin	ATW	28.8538	6.9640	5.3089	16.9652
	LIW	33.8585	6.9649	5.8284	18.0666
Griewank	ATW	0.0811	5.7732e-015	0.0205	0.0239
	LIW	0.1052	9.9886e-011	0.0256	0.0328

5 Application on the Portfolio Optimization

5.1 Portfolio Optimization

Modern portfolio analysis started from pioneering research work of Markowitz (1952) [17] who proposed the original mean–variance model. In this paper, based on the original mean-variance model, we present an improved mean–variance model considering the transaction fee produced by selling and buying, and other constraint conditions such as no short sales, the original portfolio owned by the investor. It assumes that an investor allocates his/her wealth among n ($i = 1, \dots\dots, n$) assets. Some notations are introduced as follows:

x_i is the proportion of the money used in the ith asset, and $\sum_{i=1}^{n} x_i = 1$. $x_i \geq 0$ means there is no short sales. $x_i^0 \geq 0$ is the original the original portfolio owned by the investor. So when $x_i^0 = 0$, there is no original investment in the ith asset, or it is the new one introdeced in this period. r_i is the yield of the ith asset; $R_i = E(r_i)$ means the expected rate of revenue of the ith asset. $\sigma_{ij} = \mathrm{cov}(r_i, r_j)$ is the covariance of r_i and r_j . k_i^b and k_i^s are the transaction fee when buying and selling the ith asset respectively, which are caculated by proportion. And k_i^s is usuall more than k_i^b . λ is the risk-averse factor, wihch distribute in [0, 1]. Smaller λ represents the investor could bear larger risk.

Based on these defined variables, Our improved portfolio optimization model can be formulated as:

$$\min F(x) = \min\{\lambda \sum_{i=1}^{n}\sum_{i=1}^{n} \sigma_{ij} x_i x_j - (1-\lambda)\sum_{i=1}^{n} R_i x_i - \sum_{i=1}^{n} [\mu \cdot k_i^b \cdot (x_i - x_i^0) + (1-\mu) \cdot k_i^s \cdot (x_i^0 - x_i)]\},$$

$$\begin{cases} \sum_{i=1}^{n} x_i = 1; \\ x_i \geq 0. \end{cases} \tag{4}$$

where $\mu = \begin{cases} 1, \dots\dots x_i \geq x_i^0 \\ 0, \dots\dots x_i \leq x_i^0 \end{cases}$

5.2 Illustrative Examples

5.2.1 Parameters Representation and Parameters Setting

The position and the velocity of the particle are constructed in the real-number encoding method (see table 4).

Table 4. The encoding of the particle

$x_1, x_2, \dots \dots, x_n$	$v_1, v_2, \dots \dots, v_n$	F_x
The position of the particle in every dimension	The velocity of the particle in every dimension	The fitness

Five assets are chosed as the sample, which are from different industry and different place. so $n = 5$ .it is assumes that the investor has the same original proportion in every assets, that is $x_i^0 = 0.2$. We set $k_i^b = 0.00065$, $k_i^s = 0.00075$ in the experiment. λ is set as 0.15, 0.65, 0.9 to denote the different kinds investors. In our experimental studies, $w_{start} = 0.9$, $w_{end} = 0.4$. The other parameter $c_1 = c_2 = 2.0$, $k = 0.4$, the swarm size are all 200, and $t_{\max} = 100$. A total of 50 runs for each experimental setting are conducted.

5.2.2 Experimental Results

Numerical results with three different λ obtained by the ATW and LIW are showed in the Table 5. Figs 5-7 present the mean relative performance with different λ generated by the two methods.

The revenue rate and the risk rate, the max value, the min value, the standard deviation and the mean value are summarized in Tables 6. It is obviously that for almost of all the different risk preferences, ATW gets smaller standard deviation and mean value, which demonstrated it outperforms PSO in terms of result robustness and solution quality.

From Figs 5-7, it is clearly found that ATW has quicker convergence rate with LIW in the different situations compared.

Table 5. Numerical results with different λ

	$\lambda = 0.15$		$\lambda = 0.65$		$\lambda = 0.9$	
	ATW	LIW	ATW	LIW	ATW	LIW
Revenue Rate	9.283e-002	8.137e-002	3.058e-002	3.239e-002	1.769e-002	1.200e-002
Risk Rate	1.561e-001	1.062e-001	6.731e-003	7.842e-003	2.409e-003	3.535e-003
Max	-4.840e-002	-4.372e-002	-6.258e-003	-5.578e-003	1.175e-003	3.283e-003
Min	-5.550e-002	-5.324e-002	-6.328e-003	-6.240e-003	3.996e-004	1.182e-003
Mean	-5.307e-002	-4.745e-002	-6.316e-003	-5.9469e-003	6.351e-004	1.967e-003
Std	1.552e-003	2.174e-003	1.869e-005	1.397e-004	2.215e-004	4.735e-004

All the results presented in the tables and figures can proved that the SMPSO could be a more effective way for the investors to solve the portfolio optimizations problems.

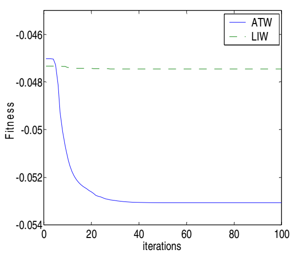

Fig. 5. $\lambda = 0.15$

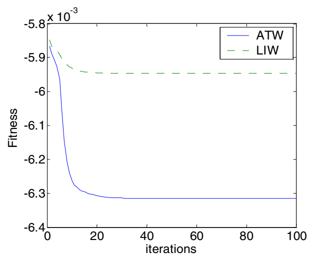

Fig. 6. $\lambda = 0.65$

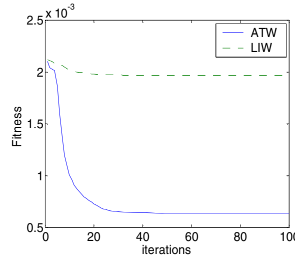

Fig. 7. $\lambda = 0.9$

6 Conclusions

This paper presents a novel PSO algorithm with non-linear inertia weight (ATW) based on the arc tangent function. The performance of ATW is evaluated by the experiments on four representative instances. It provided better quality solutions and more efficacious compared with LIW. They are also used to solve the portfolio optimization, and the result of the study showed that ATW is the more effective approach.

Future work is focused on optimizing the performance of ATW. 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 ATW.

Acknowledgment

This work is supported by Guangdong Natural Science Foundation (Grant no. 9451806001002294), Shenzhen-Hong Kong Innovative Circle project (Grant no. SG200810220137A), 863 Project(No. 008AA04A105), and Project 801-000021 supported by SZU R/D Fund.

References

1. Kennedy, J., Eberhart, R.C.: A New Optimizer Using Particle Swarm Theory. In: 6th International Symposium on Micromachine and Human Science, pp. 39–43 (1995)
2. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: IEEE International Conference on Neural Networks, vol. IV, pp. 1942–1948 (1995)
3. Clerc, M., Kennedy, J.: The Particle Swarm: Explosion Stability and Convergence in a Multi-Dimensional Complex Space. IEEE Transactions on Evolutionary Computation 6, 58–73 (2002)
4. Eberhart, R.C., Shi, Y.: Comparison between Genetic Algorithms and Particle Swarm Optimization. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 611–616. Springer, Heidelberg (1998)

5. Eberhart, R.C., Shi, Y.: Particle Swarm Optimization: Developments, Applications and Resources. In: Congress on Evolutionary Computation, vol. 1, pp. 68–81 (2001)
6. Shi, Y., Eberhart, R.C.: Empirical Study of Particle Swarm Optimization. In: Congress on Evolutionary Computation, vol. 3, pp. 1945–1949 (1999)
7. Niu, B., Li, L.: A Novel PSO-DE-Based Hybrid Algorithm for Global Optimization. In: Huang, D.-S., Wunsch II, D.C., Levine, D.S., Jo, K.-H. (eds.) ICIC 2008. LNCS (LNAI), vol. 5227, pp. 156–163. Springer, Heidelberg (2008)
8. Niu, B., Xue, B., Li, L.: Symbiotic Multi-swarm PSO for Portfolio Optimization. In: Huang, D.-S., Jo, K.-H., Lee, H.-H., Kang, H.-J., Bevilacqua, V. (eds.) ICIC 2009. LNCS, vol. 5755, pp. 776–784. Springer, Heidelberg (2009)
9. Zhang, L.P., Yu, H.J., Hu, S.X.: Optimal Choice of Parameters for Particle Swarm Optimization. Journal of Zhejiang University Science 6, 528–534 (2004)
10. Shi, Y., Eberhart, R.C.: Parameter Selection in Particle Swarm Optimizations. In: Porto, V.W., et al. (eds.) EP 1998. LNCS, vol. 1447, pp. 591–600. Springer, Heidelberg (1998)
11. Shi, Y., Eberhart, R.C.: A Modified Particle Swarm Optimizer. In: IEEE International Conference on Evolutional Computation, pp. 69–73 (1998)
12. Wang, L., Wang, X.K.: Modified Particle Swarm Optimizer Using Non-Linear Inertia Weight. Computer Engineering and Applications 43, 47–48 (2007)
13. Van den Bergh, F.: An Analysis of Particle Swarm Optimizer, University of Pretoria, South Africa (2002)
14. Niu, B., Zhu, Y.L., He, X.X.: MCPSO: A Multi-Swarm Cooperative Particle Swarm Optimizer. Applied Mathematics and Computation 185, 1050–1062 (2007)
15. Liang, J.J., Suganthan, P.N., Qin, A.K.: Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions. IEEE Transactions on Evolutionary Computation 10, 281–295 (2006)
16. Parsopoulos, K.E., Vrahatis, M.N.: UPSO-A Unified Particle Swarm Optimization Scheme. Lecture Series on Computational Sciences, pp. 868–873 (2004)
17. Markowitz, H.W.: Foundations of Portfolio Theory. Journal of Finance 46, 469–477 (1991)