Simultaneously Optimizing Inertia Weight and Acceleration Coefficients via Introducing New Functions into PSO Algorithm

Wanli Yang1,2, Xueting Zhou2*, Yulan Luo1

1School of Electrical and Information Engineering, Hunan Institute of Technology, Hengyang, Hunan, 421002, China
2School of Electrical and Information Engineering, Hunan University, Changsha, Hunan, 410082, China

*Corresponding author’s e-mail: 2015001984@hnit.edu.cn

Abstract. Because of the drawbacks of easy premature in initial iteration stages, the low convergence accuracy and slowed-down converging speed in final stages of the particle swarm optimization (PSO) algorithm therefore the simple particle swarm optimization (SPSO) algorithm with dynamic changes of inertia weight and acceleration coefficients (IASPSO) has been put forward. IASPSO algorithm provides a parameter optimization strategy by using exponential decreasing inertia weight and sine function acceleration coefficient to improve global exploration capacity. Simulation tests are carried out with classic Benchmark test functions. The simulation results show that compared with other PSO algorithms, IASPSO algorithm can converge to the better global optimization with a fast converging velocity and high convergence precision, promoting the optimization performance of the algorithm.

1. Introduction

Particle Swarm Optimization (PSO) algorithm, first proposed by Kennedy and Eberhart in 1995, is a swarm intelligence optimization algorithm [1]. It has successfully solved complicated optimization problems such as nonlinear optimization, non-differentiable minimization, and has been widely applied to scientific and engineering fields like route optimization, and system control [2,3]. Nevertheless, PSO has the drawbacks of easy premature in initial iteration stages, low convergence accuracy and slowed-down converging speed in final stages, which restricted the applications of PSO. Therefore, it is of great significance to carry out parameter optimization for PSO.

To overcome the drawbacks of PSO, quantities of researches have been studied by scholars. Shi.Y successively proposed inertia weight, linear inertia weight, fuzzy inertia weight. The several nonlinear inertia weight strategies have been reported respectively, which have obtained better optimization, but have ignored the features in each stage of algorithm optimization. A mean particle swarm optimization algorithm was proposed to adaptively change the inertia weight with the iterative process, which effectively improves the global search ability of the algorithm [4]. A PSO algorithm for dynamically adjusting the inertia weight was proposed, which effectively avoids the optimization process falling into local optimum [5]. An adaptive PSO algorithm based on guiding strategy was proposed [6]. A simplified PSO algorithm without particle velocity item come forward, which overcame the velocity item’s negative influence in optimization [7]. A fruit fly optimization algorithm that introduces inertia weights and learning factors was proposed, which took the influence of inertia weight and acceleration
coefficients on the algorithm’s search capability and obtained better optimal performance \[8\]. However, as parameters in each search stage were of linear variation, it could not relevantly report the real search process.

In order to improve the optimal performance of PSO algorithm, a parameter optimization strategy, simple particle swarm optimization (SPSO) algorithm with dynamic changes of inertia weight and learning factor, is proposed in this article. In addition, the convergence conditions of PSO algorithm are derived based on difference equation analysis. It is clear that SPSO has effectively overcome the drawbacks of easy premature in initial iteration stages, low convergence accuracy and slowed-down converging speed in final stages.

2. Particle Swarm Optimization algorithm

Please follow these instructions as carefully as possible so all articles within a conference have the same style to the title page. This paragraph follows a section title so it should not be indented \[2\]. PSO is an optimization algorithm based on community and fitness. Suppose a group of particles has \(N\) particles, the number of dimensions is \(d\), then the position of any particle \(i\) in the particle space can be expressed as \(x_i = (x_{i1}, x_{i2}, \ldots, x_{id})\), the velocity of particle \(i\) can be expressed as \(v_i = (v_{i1}, v_{i2}, \ldots, v_{id})\). The formulas of particles’ refresh rate and position are as follows:

\[
\begin{align*}
\dot{x}_{id}^{k+1} &= \omega \dot{x}_{id}^k + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \\
x_{id}^{k+1} &= x_{id}^k + \dot{x}_{id}^{k+1}
\end{align*}
\]

where \(x_{id}^k\) is the dimension \(d\)’s velocity and \(x_{id}^k\) is the dimension \(d\)’s position when the iteration of particle \(i\) is \(k\), \(\omega\) is inertia weight, nonnegative constants \(c_1\) and \(c_2\) are acceleration coefficients, \(r_1\) and \(r_2\) are uniform random numbers on \([0, 1]\), \(p_{id}\) is the present particle’s optimal position in history, and \(p_{gd}\) is the whole particle swarm’s optimal position in history. Based on PSO algorithm, a simplified PSO algorithm was put forward in \[5\], which is described as follows:

\[
x_{id}^{k+1} = \omega x_{id}^k + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id})
\]

Formula (3) omits velocity three items, including inertia item, particle’s self-cognitive item and particle’s social-cognitive item.

3. Parameter Optimization on Particle Swarm Optimization algorithm

3.1. Inertia weight optimization

Inertia weight is an important regulation parameter in PSO algorithm which is directly related to the optimal optimization of the algorithm. In this section, the optimization strategy of weight inertia has been discussed.

3.1.1 Linearly decreasing inertia weight.

\[
\omega = \omega_1 + (\omega_2 - \omega_1) \left( \frac{T_{\text{max}} - t}{T_{\text{max}}} \right)
\]

where \(\omega_1\) is initial weight, \(\omega_2\) is final weight, \(T_{\text{max}}\) is maximum iterations, \(t\) is the present iterations.

Although linear weight has greatly improved the performance of PSO algorithm, yet there are some drawbacks: the exploitation capacity is weak in the early iterations, which can easily miss the global optimum; the exploration capacity is also frail in the later iterations, which is easy to be caught in local optimum and difficult to predict the maximum iterations in the PSO algorithm.

3.1.2 Nonlinearly decreasing inertia weight.

1) Downward parabola decreasing inertia weight
\[
\omega = \omega_1 - (\omega_1 - \omega_2)(\frac{t}{T_{\text{max}}})^2
\]

2) Upward parabola decreasing inertia weight
\[
\omega = \omega_1 + (\omega_1 - \omega_2)(\frac{t}{T_{\text{max}}})^2 + (\omega_2 - \omega_1)(\frac{2t}{T_{\text{max}}})
\]

3) Exponential decreasing inertia weight
\[
\omega = \omega_2 \left(\frac{\omega_1}{\omega_2}\right)^{\frac{t}{T_{\text{max}}}}
\]

3.1.3 Options of inertia weight.
Comprehensive analysis and comparison of the above-mentioned inertial weights, we found that the
descent velocities of both upward parabolas decreasing inertia weight and exponential decreasing
inertia weight are greater than that of linearly decreasing inertia weight, and the velocities decrease
with the increase of iterations. In addition, it was testified that the exponential decreasing inertia
weight could lead to faster convergence and higher convergence precision in the algorithm. Therefore,
the optimization strategy of exponential decreasing inertia weight is adopted.

3.2. Inertia weight optimization
Acceleration coefficient works as an important parameter in PSO algorithm. Whether it has a proper
value is directly related to the optimal optimization of the algorithm. Several optimization strategies of
acceleration coefficient are introduced in this report.

3.2.1 Linear acceleration coefficient.
1) Synchronous acceleration coefficient
\[
c_1 = c_2 = c_{\text{max}} - \frac{t \cdot (c_{\text{max}} - c_{\text{min}})}{T_{\text{max}}}
\]
where \(c_{\text{max}}\) and \(c_{\text{min}}\) are respectively maximum value and minimum value of acceleration coefficient.

2) Asynchronous acceleration coefficient
\[
\begin{cases}
    c_1 = c_{1m} + \frac{t \cdot (c_{1n} - c_{1m})}{T_{\text{max}}} \\
    c_2 = c_{2m} + \frac{t \cdot (c_{2n} - c_{2m})}{T_{\text{max}}}
\end{cases}
\]
where \(c_{1m}\) and \(c_{1n}\) are respectively initial value to iteration and final value to iteration of acceleration
coefficient \(c_1\), and \(c_{2m}\) and \(c_{2n}\) are respectively initial value to iteration and final value to iteration of
acceleration coefficient \(c_2\).

3.2.2 Nonlinear acceleration coefficient.
1) Sine function acceleration coefficient
3.2.3 Optimization strategies of acceleration coefficient.

Studies show that, with larger $c_1$ and smaller $c_2$, the algorithm has better exploration capacity in the initial stage, while with smaller $c_1$ and larger $c_2$, the algorithm restrains better global optimum in the late stage. Compared with synchronous acceleration coefficients, it is more suitable to the practice of the algorithm’s evolution carrying out adjustment in phases, and it is better for enhancing the algorithm’s optimizing capacity when keeping both relatively independent. Simulated analysis shows it has better optimization to adopt strategy (10) than to adopt strategy (11). Based on comprehensive analysis, the optimization strategy of sine function acceleration coefficient is adopted in this article.

3.3. Optimization strategies of inertia weight and acceleration coefficient

Based on the comprehensive analysis on the optimization strategies of inertia weight and acceleration coefficient, our work focus on exponential decreasing inertia weight and sine function acceleration coefficient. For convenience, the algorithm is recorded as IASPSO.

4. Simulation and analysis

4.1. Algorithm comparison

To verify the validity of parameter optimizing strategy in IASPSO algorithm, IASPSO is compared and analysed with 5 other algorithms. For convenience, the 5 other algorithms are recorded as PSO, A1-SPSO, A2-SPSO, A3-SPSO and A4-SPSO, whose detailed descriptions are as follows:

1) PSO: $\omega=0.8$, and $c_1=c_2=2.0$.
2) A1-SPSO: calculate the PSO when $\omega=0.8$, and $c_1$ and $c_2$ are sine functions.
3) A2-SPSO: calculate the PSO when $\omega$ is linear function, and $c_1$ and $c_2$ are sine functions.
4) A3-SPSO: calculate the PSO when $\omega$ is upward parabola, and $c_1$ and $c_2$ are sine functions.
5) A4-SPSO: calculate the PSO when $\omega$ is downward parabola, and $c_1$ and $c_2$ are sine functions.

4.2. Algorithm test functions

4 classic test functions (Sphere, Rosenbrock, Griewank and Rastrigin) were chosen to be tested, among which Sphere and Rosenbrock were unimodal functions and Griewank and Rastrigin were multimodal functions. Each global minimum of them was 0. Their function expressions were as follows:

1) Sphere function

$$\begin{align*}
c_1 &= 2\sin^2\left[\frac{\pi \cdot (T \text{ max} - t)}{2T \text{ max}}\right] \\
c_2 &= 2\sin^2\left[\frac{\pi \cdot t}{2T \text{ max}}\right]
\end{align*}$$

$$\begin{align*}
c_1 &= 1.3 + 1.2\cos\left(\frac{\pi \cdot t}{T \text{ max}}\right) \\
c_2 &= 2 - 1.2\cos\left(\frac{\pi \cdot t}{T \text{ max}}\right)
\end{align*}$$
\[ f_1(x) = \sum_{i=1}^{d} x_i^2, x_i \in (-100,100)^d \]  

(12)

2) Rosenbrock function

\[ f_2(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2) + (x_i - 1)^2] \]
\[ x_i \in (-2.048,2.048)^d \]  

(13)

3) Rastrigrin function

\[ f_3(x) = \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i) + 10] \]
\[ x_i \in (-10,10)^d \]  

(14)

4) Griewank function

\[ f_4(x) = \frac{1}{4000} \sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \]
\[ x_i \in (-600,600)^d \]  

(15)

4.3. Analysis of simulation results

To verify the validity and feasibility of IASPSO, 6 algorithms were tested applied to 4 test functions. Their parameters were set as follows: particle swarm scale \(N=40\), dimensionality \(d=30\), initial weight \(\omega_1=0.9\), final weight \(\omega_2=0.4\), adjustable parameter \(m=10\); the convergence precision of \(f_1(x), f_2(x)\) and \(f_3(x)\) was \(10^{-10}\), and the convergence precision of \(f_4(x)\) was \(10^{-5}\); the maximum iteration of \(f_1(x)\) and \(f_4(x)\) \(T_{\text{max}}=50\), and the maximum iteration of \(f_2(x)\) and \(f_3(x)\) \(T_{\text{max}}=100\). All of the 6 algorithms were operated 50 times applied to all functions.

Figure 1-4 shows 4 test functions’ fitness curves of 6 algorithms, it should be noted that the curves 1-6 in the figure1 to figure 4 represent the six algorithms IASPSO, PSO, A1-SPSO, A2-SPSO, A3-SPSO and A4-SPSO respectively. It can be observed from figure 1-4 that, in all the algorithms, the average fitness value of IASPSO has the fastest descent velocity with the increase of iterations, and remains a faster convergence rate in the late stage of iteration; and for all the test functions, IASPSO can keep higher convergence precision and restrain itself to global optimization.

Observed from the above-mentioned analysis results, and compared with the other algorithms, IASPSO algorithm has remarkable improvement in convergence velocity, convergence precision and optimizing capacity, and it avoids precocity effectively. For single-peak function and multimodal function, IASPSO algorithm achieves better optimal performance. IASPSO algorithm has improved the algorithm’s optimal performance and optimizing capacity.

Figure 1. Sphere function’s fitness curve. Figure 2. Rosenbrock function’s fitness curve
5. Conclusion
By analysing the parameter and optimization mechanism for PSO algorithm, IASPSO algorithm with exponential decreasing inertia weight and sine function acceleration coefficient has been put forward. The proof and comparison of simulation tests of Benchmark function illustrates that IASPSO algorithm has the remarkable improvement in convergence velocity and convergence precision, as well as avoids precocity effectively. Furthermore, the results testify the rationality and validity of the parameter optimization strategy of IASPSO algorithm. The novel strategy offers particular theoretical value and reference for the follow-up studies on PSO algorithm.

Acknowledgment
Thanks to the Hunan Provincial Department of Education project (17C0431) for funding this research.

References
[1] Eberhart R C, Kennedy J. (1995) A new optimizer using particle swarm theory. In: Proceeding of the Sixth International Symposium on Micro Machine and Human Science. Piscataway. pp.39-43.
[2] Eberhart R C, Shi Y H. (2001) Particle swarm optimization: developments, applications and resources. In: Proceedings of the IEEE Congress on Evolutionary Computation. Piscataway. pp.81-86.
[3] Yu L., Sun Y., Xu R. (2017) Improved Particle Swarm Optimization Algorithm and Its Application in Reactive Power Partitioning of Power Grid. Automation of Electric Power Systems, 41(3): 89-95,128.
[4] Zhao Z.G., Lin Y.J., Yin Z.Y. (2016) A mean particle swarm optimization algorithm based on adaptive inertia weight. Computer Engineering and Science, 38(3): 501-506.
[5] Dong H.B., Li D.J., Zhang X.P. (2018) Particle Swarm Optimization Algorithm with Dynamically Adjusting Inertia Weight. Computer Science, 45(2): 98-102,139.
[6] Zhang F.L., Zhang Y. (2017) Adaptive particle swarm optimization algorithm based on guiding strategy. Applications Research of Computers, 34(12): 3599-3602.
[7] Cheng R., Jin Y. (2015) A social learning particle swarm optimization algorithm for scalable optimization. Information Sciences, 291(6): 43-60.
[8] Wang Y.B., Wang Y.X. (2018) A multiple subgroups fruit fly optimization algorithm based on sequential quadratic programming local search. Computer Engineering and Science, 40(5): 906-915.