Bulletin of Electrical Engineering and Informatics
ISSN: 2302-9285
Vol. 5, No. 4, December 2016, pp. 489~494, DOI: 10.11591/eei.v5i4.561

A Improved Particle Swarm Optimization Algorithm with Dynamic Acceleration Coefficients

Gang Ma*, Renbin Gong, Qun Li, Gang Yao
PetroChina Research Institute of Petroleum Exploration & Development-Northwest,
Lanzhou 730020, Gansu Province, China
*Corresponding author, e-mail: magangzh@126.com

Abstract

Particle swarm optimization (PSO) is one of the famous heuristic methods. However, this method may suffer to trap at local minima especially for multimodal problem. This paper proposes a modified particle swarm optimization with dynamic acceleration coefficients (ACPSO). To efficiently control the local search and convergence to the global optimum solution, dynamic acceleration coefficients are introduced to PSO. To improve the solution quality and robustness of PSO algorithm, a new best mutation method is proposed to enhance the diversity of particle swarm and avoid premature convergence. The effectiveness of ACPSO algorithm is tested on different benchmarks. Simulation results found that the proposed ACPSO algorithm has good solution quality and more robust than other methods reported in previous work.

Keywords*: PSO, dynamic acceleration coefficients, heuristic*

1. Introduction

Parcicle swarm optimization (PSO) is a population-based search optimization technique introduced by Kennedy and Eberhart in 1995 [1] [2]. On the simulation of the behavior of bird flocking, the particle swarm algorithm starts with the random initialization of a population of individuals in the search space [1]. It tries to find the global best solution by adjusting the trajectory of each individual toward its own best location and the best particle of the whole swarm at each iteration [1]. The PSO method is becoming very popular due to its simplicity of implementation and quick convergence to a global optima. Since PSO was introduced, it has become a popular optimizer and has widely been applied in practical problems. Syahputra [3] uses PSO Algorithm to optimize configuration of distribution network with distributed energy resources integration. Lu [4] introduces PSO algorithm into the Artificial Neural Network training and construct a BP neural network model to improve the prediction accuracy for the cost forecasting of transmission line project based on historical project data. Wang [5] introduced PSO into the hardware structure design of the dynamic compensator to facilitate the application of an optimized compensator to real-time online measurement.

In the particle swarm algorithm, the trajectory of each individual in the search space is adjusted by dynamically altering the velocity of each particle, according to its own flying experi ence and the flying experience of the other particles in the search space. In a *D*-dimensional space, a particle represents a potential solution presented by their velocity and position and *D* is the number of the parameters. During a search process, each particle is attracted by its previous best particle (*pbest*) and the global best particle (*gbest*) as follows.

$$v_{id}(t+1) = w \cdot v_{id}(t) + c1 \cdot r1 \cdot \left(pbest_{id}(t) - x_{id}(t)\right) + c2 \cdot r2 \cdot \left(gbest_{d}(t) - x_{id}(t)\right) \quad (1)$$

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

Where $i = 1,2, \dots, N$ is the particle's index, N is the population size, $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ is the position of the ith particle. $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ represents the velocity of the ith particle; the $pbest_i = (pbest_1, pbest_2, \dots, pbest_D)$ is the best previous position in the ith particle own history and the $gbest = (gbest_1, gbest_2, \dots, gbest_D)$ is the global best particle found by all particle so far.

Received June 9, 2016; Revised October 23, 2016; Accepted November 6, 2016

The inertia weight *w* was introduced by Shi and Eberhart [6] [7] for the classical PSO, which is used to balance the exploration and the exploitation abilities, and suggested that the optimal solution can be improved by altering the value of w linearly decreasing (PSO-*w*) from 0.9 to 0.4. In general, the higher values of w help particles to search space more thoroughly in the process and find the global minima, while lower values help to search the local space and find the local minima.

$$w = (w_{max} - w_{min}) \cdot \frac{MAXITER - iter}{MAXITER} + w_{min} \quad (3)$$

Jiao [8] made an improved method on the inertia weight and proposed the dynamic inertia weight decreases according to iterative generation increasing as below.

$$w = w \cdot u^{-k}, (\, w\epsilon[0,1], u\epsilon[1.001, 1.005]\,) \quad (4)$$

The $r1$ and $r2$ are two uniform random numbers generated independently within the range of [0, 1], $c1$ and $c2$ are two learning factors which control the influence of the social and cognitive components, and $t = 1, 2, \ldots, FEs$ indicates the iteration number. The first part of (1) represents the previous velocity, which keeps its own momentum for particles to roam across the search space. The second part, known as the cognitive component, represents the personal judging of each particle that helps the particle to move toward its own best position found so far. In the past decades, many variants of PSO have been proposed [6] [9] [10].

2. Research Method

In this paper, we propose a dynamic parameter adjusting strategy and a neighborhood learning strategy. The major objective of this development is to improve the performance after a predefined number of generations, through empirical simulations with well-known benchmarks.

2.1. Parameter Dynamically Adjusting Strategy

Although the PSO-*w* [6] can find a satisfactory solution with fast speed, its ability to find optimum solution is limited, because of the diversity loss at the latter stage of evolution process. To find the optimum solution [1], it is better to encourage the individuals to search through the entire search space during the early part of the search, avoiding clustering around local optima, and during the later stage, convergence towards the global optima is encouraged.

The parameters affects the performance of PSO in the optimum solutions. Under this new development, we reduce social component and increase the cognitive component, by dynamiclly changing the acceleration coefficients. With a large social component and small cognitive component at the beginning, particles are allowed to move around the search space, instead of moving toward the local optimum. On the other hand, a small social component and a large cognitive component allow the particles to converge to the global optima in the latter part of the optimization. This modification can be mathematically represented as follows:

$$c1 = 2.0 + 0.5 \cdot cos\frac{\pi \cdot iter}{MAXITER} \quad (5)$$

$$c2 = 2.0 - 0.5 \cdot cos\frac{\pi \cdot iter}{MAXITER} \quad (6)$$

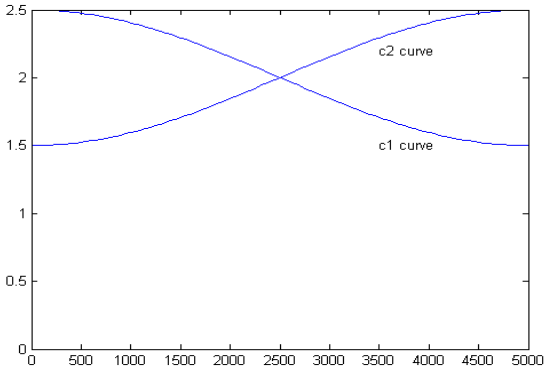

Figure 1. The Curve of $c1$ and $c2$ Trend

2.2. Swarm Diversity Enhanced Method

In the basic PSO algorithm, particles are attracted by their pbests and the gbest and all particles move quickly to the same direction and the loss of swarm diversity is a serious problem for PSO algorithms. To maintain diversity of a swarm, some excellent strategies have been proposed. Riget [9] proposed a diversity-guided PSO, which defines a repulsion phase based on a new velocity updating mechanism and calculating diversity of the swarm is necessary. Sun et al. [10] introduced a mutation operator to enhance the swarm diversity. When the diversity is less than a predefined value, a mutation on $gbest$ is conducted to increase the dissimilarities among particles. The method has good performance but it costs much computational time to calculate the swarm diversity.

In order to maintain diversity and avoid calculating the diversity, this paper proposes a novel mechanism instead of diversity computing for PSO. We add a third part which represents the collaborative effect of the particles, to find the global optima.

$$v_{id}(t+1) = w \cdot v_{id}(t) + c1 \cdot r1 \cdot \left(pbest_{id}(t) - x_{id}(t)\right) + c2 \cdot r2 \cdot \left(gbest_{d}(t) - x_{id}(t)\right) + (c3 + c4) \cdot r3 \cdot \left(pbest_{jd}(t) - x_{id}(t)\right) \quad (7)$$

The $gbest(t)$ does not change for a constant times which means that the particles find the local optima and the diversity loss of the swarm is a problem. When this happens, we can set $c3 = 0.1$ to enhance the mutation for each particle. When $pbest_i$ does not change for a constant times which means that the particle has been trapped around the local optima and we can set $c4 = 0.1$ to enhance the mutation for the particle. The $pbest_{jd}(t)$ is best position for the particle *j* found so far and $r3$ is a random number in [0, 1].

2.3. Proposed PSO

The flowchart of the ACPSO algorithm is shown in Figure 2.

Table 1. Benchmark Problems Tested in the Experiments

Problems	Name	X	$f(x^D)$
F1	Rosenbrock's function	[-2.048, -2.048]	0
F2	Ackley's function	[-32.768,32.768]	0
F3	Griewanks's function	[-600,600]	0
F4	Rastrigin's function	[-5.12,5.12]	0
F5	Nocontinuous Rastrigin's function	[-5.12,5.12]	0
F6	Schewfel function	[-500,-500]	0
F7	Weierstrass function	[-0.5,-0.5]	0
F8	E_ScafferF6 function	[-100,100]	0

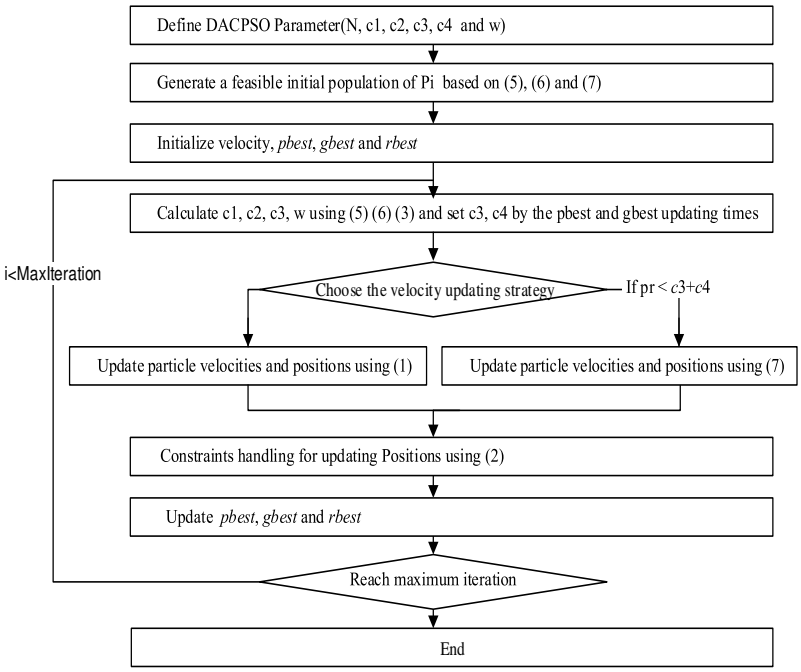

Figure 2. Flowchart of Proposed ACPSO Algorithm.

The stop criteria are that the maximum iteration number is reached or the minimum error condition is satisfied. In ACPSO algorithm, each particle of the swarm dynamically changes acceleration coefficients and execute diversity improved method during the search process.

3. Results and Analysis

The functions with dimension 30 is used to show the characteristics of the proposed PSO. The parameters are set as below. Table 1 illustrates the functions, including the best fitness value and average velocity of all particles. Table 2 illustrates the variation of parameters in the evolution, including inertia weight, $c1$, and $c2$.

Table 2. PSO Algorithms for Comparison

Algorithm	Parameters	References
SPSO	w: $0.729, c1 = c2 = 1.49445$	[1]
PSO-w	w: $0.9 - 0.4, c1 = c2 = 2$	[6]
IPSO	$w_{initial}$: $0.3, u = 1.002, c1 = c2 = 1.49445$	[8]
ACPSO	w: $0.9 - 0.4, c1_{initial} = 2, c1_{initial} = 1$	This paper

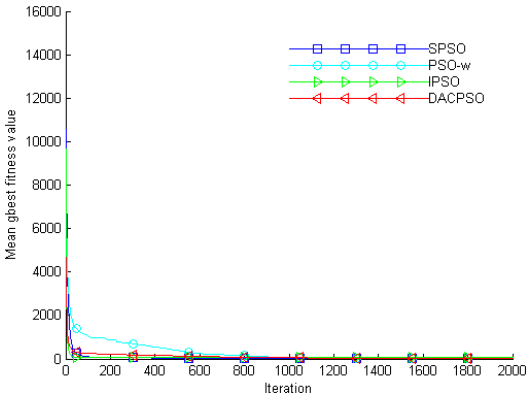

Figure 3. Rosenbrock's Function

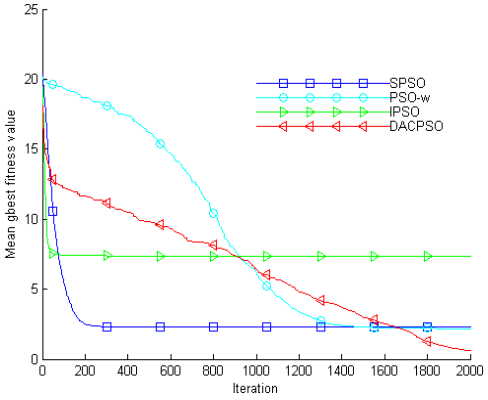

Figure 4. Ackley's Function

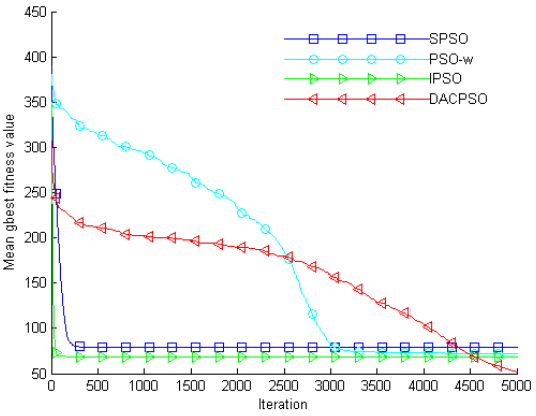

Figure 5. Rastrigin's Function

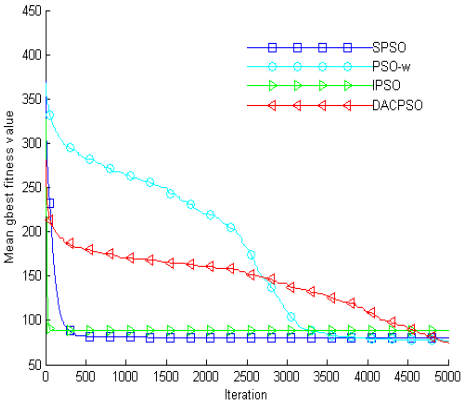

Figure 6. Nocontinuous Rastrigin's Function

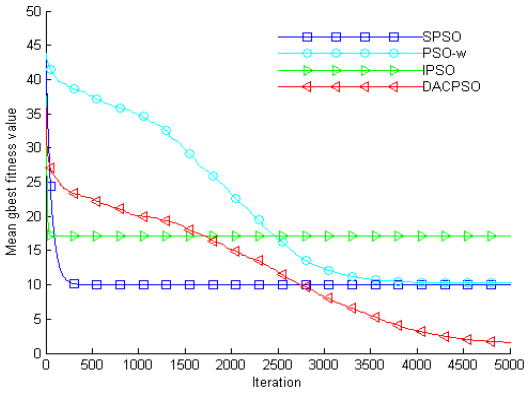

Figure 7. Weierstrass Function

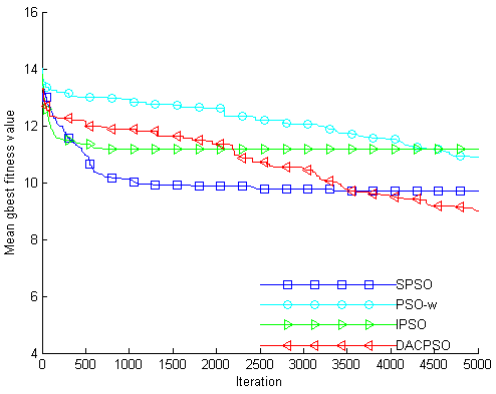

Figure 8. E_ScafferF6 Function

The result shown in Table 3 is typical. There are 8 functions tested and six of them show the best result. Mostly the best fitness value of ACPSO is better than that of others. And the Figuire 3-8 show the fitness trend of the search process and ACPSO can always find a better solution. It can be known that PSO algorithm quite depends on the inertia weight in large-scale optimization problems and the performance of standard PSO can be improved by the inertia weight of particle swarm changing along with the nonlinear ideal inertia weight. As the the parameter $c1$ starts with a large value and the particle can keep its direction and try to find more

space and at the last stage $c1$ gets smaller and smaller and but the $c2$ get bigger as a result the particle try to search the same direction and and finally improve the searching result which can be shown in Figure 3-8.

4. Conclusion

This paper presents an improved PSO algorithm called ACPSO to solve complex optimization problems. The proposed approach employs two strategies including dynamic acceleration coefficients strategy and diversity improved method. By combining the two strategies, ACPSO achieves a trade-off between the exploration and exploitation abilities. To verify the performance of ACPSO, different benchmark functions are tested in the experiments. From the experiments and analysis we observe that this strategy enables our algorithm to perform more efficiently.

Table 3. Comparison of Results on 50-time Independent Random Testing and *D*=30

Problems	SPSO	PSO-*w*	IPSO	ACPSO
	mean/std	mean/std	mean/std	mean/std
F1	3.24e+1/ 4.99e+2	2.82e+1/2.412	5.72e+1/1.41e+3	**2.74e+1**/**1.27**
F2	2.31/ 5.78e-1	2.18/04.14e-1	7.38/1.93	**5.87e-1**/**3.96e-1**
F3	1.45e-2/ 2.87e-4	**1.55e-2**/**5.15e-4**	2.57e-1/4.55e-2	1.24e-1/1.32e-2
F4	7.92e+1/ 4.43e+2	72.167/4.384e+2	68.822/2.95e+2	**52.29**/**4.75e+2**
F5	79.90/ 5.25e+02	77.17/7.14e+2	88.27/6.69e+2	**74.87**/**3.67e+2**
F6	3.60e+3/4.97e+5	**2.74e+03**/5.58e+05	5.95e+3/5.10e+5	3.07e+3/**4.10e+5**
F7	9.95/8.71	10.25/5.51	17.14/5.34	**1.59**/**1.43**
F8	9.69/0.73	10.89/ **0.63**	11.17/1.70	**8.99**/ 1.27

References

[1] R Eberhart, J Kennedy. *A New Optimizer Using Particle Swarm Theory*. Proceedings of the 6th International Symposi. Micro Machine & Human Science. Nagoya. 1995: 39-43.

[2] J Kennedy, R Eberhart. *Particle swarm optimization*. IEEE International Conference on Neural Networks. Piscataway, NJ. 1995; 4:1942-1948.

[3] R Syahputra, I Robandi, M Ashari. Reconfiguration of Distribution Network with Distributed Energy Resources Integration Using PSO Algorithm. *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*. 2015; 13(3): 759-766.

[4] Y Lu, D Niu, B Li M Yu. Cost Forecasting Model of Transmission Project based on the PSO-BP Method. *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*. 2014; 12(4): 773-778.

[5] W Wang, Z Zhang. Online Correction of the Dynamic Errors in a Stored Overpressure Measurement System. *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*. 2015; 13(3): 828-835.

[6] Y Shi, RC Eberhart. *A modified particle swarm optimizer*. Proceedings of the IEEE Congress on Evolutionary Computation. Piscataway, NJ. 1998:69-73.

[7] Y Shi, RC Eberhart. *Empirical study of particle swarm optimization*. Proceedings of the IEEE Congress on Evolutionary Computation. Piscataway, NJ. 1999:1945–1950.

[8] B Jiao, Z Lian, X Gu. A dynamic inertia weight particle swarm optimization algorithm. Chaos, Solitons & Fractals. 2008; 37(3): 698-705.

[9] J Riget, JS Vesterstom. A diversity-guided particle swarm optimizer–the ARPSO. Technical report, EVAlife, Denmark. 2002.

[10] J Sun, BW Xu, W Fang. *A diversity-guided quantum-behaved particle swarm optimization algorithm*. International Conference on Simulated Evolution and Learning. 2006: 497–504.