

Adaptive Mutation Based Particle Swarm Optimization Algorithm

Jiyang Dai, Jin Ying

Nondestructive Test Key Laboratory of Ministry Education
Nanchang Hangkong University
Nanchang, China

E-mail: djiyang@163.com, 783026713@qq.com

Abstract—In this paper an adaptive mutation based PSO (AMBPSO) is presented for improvement of deficiencies of standard PSO, which is modified by the combination of dynamic adjustment of the inertia weights, the update of position and velocity of each particle by means of randomly adaptive mutation, and the limit of the update for the change in a reasonable range. The optimization results of two standard test functions show that these modifications can enhance particles' activity to improve the algorithm's search precision and convergence speed and to keep away from easily immersing in local minima efficiently compared with standard PSO and general PSO.

Keywords—particle swarm optimization; swarm intelligent optimization; adaptive mutation; modified algorithm

I. INTRODUCTION

Particle Swarm Optimization (PSO) is a technique modeling swarm intelligence that was developed by Kennedy and Eberhart [1] in 1995, who were inspired by the natural flocking and swarming behavior of birds. Compared with other swarm intelligent optimization algorithms such as Genetic Algorithms [2], it has more intuitive simplicity, less parameter settings and higher efficiency. It is a recently invented high performance optimizer that possesses several highly desirable attributes and especially that is an effective global optimization search algorithm for dealing with the complex nonlinear optimization problem with difficulty to solve by the traditional optimization methods [3-4].

Many advances in PSO development elevated its capabilities to handle a wide class of complex engineering and science optimization problems. Summaries of recent advances in these areas are presented in [5] and [6]. The existed PSO algorithms mainly include PSO with inertia weight, PSO with contraction factor, the binary PSO, the niche PSO and hybrid PSO, etc. Different variants of the PSO algorithm were proposed but the most standard one is the global version of PSO (Gbest model) introduced by Shi and Eberhart [7], where the whole population is considered as a single neighborhood throughout the optimization process. Although a key attractive feature of the PSO approach is its simplicity as it involves only two model equations, it has some disadvantages, such as less sensitive to environmental variations, and easily immersing in local minima frequently [8, 9]. Many improved techniques

effectively increased the PSO's local disadvantages to a certain extent, but these problems need to be further resolved [10-14].

In this paper an adaptive mutation-based PSO (AMBPSO) is presented for improvement of the above deficiencies of standard PSO which is modified by the combination of dynamic adjustment of the inertia weights, update of position and velocity of each particle by means of randomly adaptive mutation, and the limit of the update for the change in a reasonable range. These modifications can enhance particles' activity to improve the algorithm's search precision and convergence speed and to keep away from easily immersing in local minima efficiently.

The paper is structured as follows. Section 2 analyses the standard PSO algorithm. Section 3 details an adaptive mutation based PSO algorithm. The optimization calculation and analysis of two standard functions using AMBPSO algorithm are given in Section 4. Conclusions are given in Section 5.

II. STANDARD PARTICLE SWARM OPTIMIZATION ALGORITHM

A. Basic Concept of PSO

In a standard PSO, a great number of particles move around in a multidimensional space, with each particle memorizing its position vector and velocity vector as well as the time at which it reached its highest level of fitness. The inspiration underlying the development of this algorithm was the social behavior of animals, such as the flocking of birds and the schooling of fish, and the swarm theory. The advantages of PSO are that it involves no evolution operators, such as crossover and mutation operators, and it does not require the adjustment of too many free parameters. PSO begins with a random population and searches for optima by continually updating this population. Moreover, each potential solution is assigned a randomized velocity. The potential solutions, called particles, are the "flown" through the problem space. Related particles can share data at the best-fitness time. The velocity of each particle is updated according to the best positions reached by all particles through iterations, and the best positions are determined by the related particles over the course of multiple generations.

It is Similar to other intelligent optimization strategies that there are individuals called as "particles", all of which

constitute a solution group in particle swarm optimization (PSO) algorithm. As they are flown through the searching space, particles are attracted towards the best solution found by the given particle's neighbors and by the particle itself. Each particle has a position vector and a velocity vector. Suppose that the searching space is N -dimensional, and M particles form the colony. The i th particle has an N -dimensional position vector X_i ($i=1, 2, \dots, M$), which means that the i th particle is located at $X_i = (x_{i1}, x_{i2}, \dots, x_{iN})^T$ in the searching space. The position of each particle is a potential result. The i th particle's "flying" velocity is also an N -dimensional vector, represented by $V_i = (v_{i1}, v_{i2}, \dots, v_{iN})^T$ ($i=1, 2, \dots, M$). The best particle among all particles can be represented by x_d^{gbest} ($d=1, 2, \dots, N$) whose fitness is $gbest$. Each particle also keeps track of the value of its best position, which is represented by x_{id}^{Pbest} ($d=1, 2, \dots, N$) corresponding to fitness $Pbset_i$.

In the standard PSO algorithm, the velocity and position of each particle are adjusted as follows:

$$v_{id}^{k+1} = w v_{id}^k + c_1 r_1 (x_{id}^{Pbest} - x_{id}^k) + c_2 r_2 (x_{id}^{gbest} - x_{id}^k) \quad (1)$$

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

where $r_1, r_2 \in (0, 1)$ are uniformly distributed random numbers between 0 and 1; c_1 and c_2 are learning rates; and w denotes the inertia weight. k ($k=1, 2, \dots, D$) is the current iteration, and D is the maximum number of iterations.

PSO randomly initializes the flock of birds throughout the searching space; every bird is called a "particle". During each individual generation, each particle adjusts its velocity vector, based on its best solution x_{id}^{Pbest} and that of its neighbors x_d^{gbest} , as calculated by means of Eq.(1) and (2). The second part $c_1 r_1 (x_{id}^{Pbest} - x_{id}^k)$ of Eq. (1) refers to the "cognitive component", which reflects the distance at which a particle is located from the best solution. The particle determines this distance on its own. On the other hand, the last part $c_2 r_2 (x_d^{gbest} - x_{id}^k)$ of Eq.(1) refers to the "social component", which reflects the distance between a particle and the best solution found by its neighbors. Eq. (1) is used to calculate the particle's new velocity on the basis of its previous velocity, the distance between its current position and its own best previous position, and the collaborative interaction between the particles. This stage features cooperation among all particles in sharing information. Finally, the particle updates its position by means of Eq. (2). The best global position, determined by taking the condition of all particles into consideration, is defined as the particle with the minimum fitness.

The role of inertia weight w is to make a compromise between local optima and global optima. Larger w is beneficial to a wide range of search and smaller w to precise search in a small area. In general, w is set as a function with linearly decreasing in the process of iterations.

$$w = w_{max} - (w_{max} - w_{min})k / D \quad (3)$$

where w_{max} and w_{min} are the initial and ultimate inertia weights, respectively.

B. Parameter Analysis

The PSO is very simple to operate that involves fewer parameters. The main parameters need to be set as follows: Target Dimensional N , Population Size m , Inertia Weight w , Velocity Coefficient c_1, c_2 , Max Iteration D . The parameters of the algorithm and meaning in the general principles of the selection are summarized below [2]:

- The target dimensional N , the number of parameters in the PSO need to be optimized, according to the definite corresponding target search space dimension.
- The population size m is generally selected from 10 to 40 suitable for solving most of optimization problems. But according to specific or very complex optimization problem, the population size can be taken between 100 and 200.
- Inertia weight w is generally selected between 0.35 and 0.9 in the practical optimization problem, where the initial and final inertia weights are usually selected as 0.9 and 0.35, respectively.
- Acceleration coefficient c_1 and c_2 are usually selected between 0 and 4, which are the main parameters for adjustment of particles "Cognitive item (Cognitive Term)" and "Social items (Social Term)". If $c_1 = 0$, then particles do not have cognitive abilities. This moment the particles have the ability to interact and can be expanded to new search space, but the particles in the iteration process are easy to fall into the local minimum; If $c_2 = 0$, the particles do not have the social information sharing capability. This time the PSO will develop random search; If $c_1 = c_2 = 0$, the particles will keep the initial velocity until they reach maximum iterating times to end;
- Maximum iteration D as the threshold value control evolution of the PSO, can reflect the evolution of the algorithm efficiency, and too much iteration will cause a tremendous waste of the algorithm, or too small iteration may not be able to reach the goal of the optimization.

C. Implementation Steps

After determining target search dimension N , population size M , the maximum number of iterations D and the learning rates c_1 and c_2 , the implemented steps of the PSO are as follows:

- 1) Initialize v_{id}^0 and x_{id}^0 .
- 2) Calculate the fitness of each particle, and store the best fitness of the individual ($pbest$) and its corresponding position x_{id}^{Pbest} , the best fitness of group ($gbest$) and its corresponding position x_d^{gbest} , the average value x_{id}^{mbest} of positions of all the best particles relative to their own.

- 3) Update velocities v_{id}^{k+1} and positions x_{id}^{k+1} according Eq. (1) and (2).
- 4) Repeat steps 2–3 until a termination criterion is satisfied.

III. ADAPTIVE MUTATION BASED PSO

A. Ideas of Improvement for PSO

Although the PSO has many advantages such as simple operation, fewer parameters, fast convergence speed, but it also has many disadvantages to be overcome, such as:

- In spite of the fact that particles can fly towards the optimal solution in search space according to all members of the group and their own experiences, the improper choice of inertia weight will lead to the lack of precise search ability of particles;
- In the process of particles' searching towards the goal of optimal solution, when a particle is close to the optimal value, its search speed will be small. As a result, the search diversity will be lost and a particle is easy to fall into the local minimum.

In order to overcome the above deficiencies, an adaptive mutation-based PSO (AMBPSO) algorithm is proposed for the improvement of the search accuracy and convergency of the standard PSO algorithm.

Firstly, the inertia weight is modified as a function with gradient descent:

$$w = (w_{\max} - w_{\min}) / k^H + w_{\min} \quad (4)$$

where H is a given positive number. If $k=1$, then $w=w_{\max}$ and if $k \rightarrow \infty$, then $w=w_{\min}$.

Secondly, we sort fitness values of all particles in each iteration. According to the ranking, the particles in the top 50% are preserved while the rest in the colony is modified by means of adaptive mutation.

$$v_{bd}^k = v_{gd}^k (1 + \beta r) \quad (5)$$

$$x_{bd}^k = x_{gd}^k (1 + \alpha r) \quad (6)$$

where,

x_{gd}^k and v_{gd}^k are position and velocity of the particles in the top 50% of fitness ranking respectively.

x_{bd}^k and v_{bd}^k are position and velocity of the particles in the rest of 50% worse particles in the colony respectively, which will be modified by adaptive mutation.

r is a uniformly distributed Gaussian random number with a range of $[0, 1]$.

α and β are two given positive numbers.

Meanwhile, each particle's position and velocity in AMBPSO algorithm are limited into the given boundaries. If

the positions or velocities are out of the given boundaries, they will be replaced by the given boundaries.

Finally, a mutation factor P is introduced as follows:

$$P = x_{id}^{mbest} - x_d^{gbest} \quad (7)$$

where x_{id}^{mbest} is a mean value of the positions of the best particles obtained in the i th iteration, and x_d^{gbest} is the global position value with the particles of the best fitness obtained so far.

The inertia weight coefficient can be dynamically adjusted on the basis of the following equation:

$$H = H(1 + \gamma r) \quad (8)$$

where γ is a positive number and r is a uniformly distributed Gaussian random number with a range of $[0, 1]$.

It is necessary to select the appropriate mutation thresholds ρ according to the actual situation. If P is less than or equal to ρ , then H is modified by Eq.(8). The aim on the introduction of mutations factor P is to improve the mutant ability of AMBPSO. Because x_{id}^{mbest} can comprehensively reflect the activities of the individual particles and x_d^{gbest} can reflect all activities of all particles, the mutations factor P is adopted not only to effectively suppress the AMBPSO premature to early fall into local minima, but also to efficiently ensure the AMBPSO to jump out of the local minima. As a result, it is critical to select an appropriate mutation threshold ρ . A large number of experiments show that, the proposed AMBPSO algorithm can effectively make up for deficiencies on convergence and premature of the standard PSO to improve the search performances of PSO.

B. Implementation Steps of the AMBPSO

In summary, The AMBPSO algorithm can be described in general as follows:

1) Determine target search dimension N , population size M , the maximum number of iterations D and the learning rates c_1 and c_2 .

2) Initialize v_{id}^0 and x_{id}^0 .

3) Calculate the fitness of each particle, and store the best fitness of the individual ($pbest$) and its corresponding position x_{id}^{pbest} , the best fitness of group ($gbest$) and its corresponding position x_d^{gbest} , the average value x_{id}^{mbest} of positions of all the best particles relative to their own.

4) Adaptively mutate 50% worse particles v_{bd}^k and x_{bd}^k according to Eq. (5) and (6).

5) Judge mutation factor P according to Eq. (7), dynamically adjust the inertia weight coefficient H according to Eq. (8) and finally calculate the inertia weight w in terms of Eq. (4).

6) Update velocities v_{id}^{k+1} and positions x_{id}^{k+1} according Eq. (1) and (2).

7) Repeat steps 3–6 until a termination criterion is satisfied.

IV. CALCULATION AND ANALYSIS OF STANDARD TEST FUNCTIONS

In order to validate the performances of the improved particle swarm optimization algorithm, the AMBPSO, the Standard PSO (SPSO) and the PSO are used to test two types of standard functions. The comparison is made from three aspects of the convergence rate, the optimal fitness value and the search precision of the three PSO algorithms. The results on the above three aspects of algorithm performances for standard test functions can indicate which algorithms are much better. Different types of standard test functions are utilized for different types of optimization problems [1].

Two types of standard test functions are used to validate the three PSO algorithms, which are defined as follows:

1) The first test function is chosen as Generalized Rastrigin function expressed as

$$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$$

Search range: $-5.12 \leq x_i \leq 5.12$

Global optimal value: $\min(f) = f(0, \dots, 0) = 0$

2) The second test function is chosen as Ackley function expressed as

$$f(x) = -20 e^{-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}} - e^{\left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i\right)} + 20 + e$$

Search range: $-32 \leq x_i \leq 32$

Global optimal value: $\min(f) = f(0, \dots, 0) = 0$

There are a great number of local minima contained in the selected test functions due to the effect of the cosine functions in the above standard test functions. If the selected intelligence group optimization algorithms have an inadequate activity for departing from local minimum values, the global optimal values cannot be achieved. These types of test functions are used to better validate the performance that the AMBPSO can make up for deficiency that both of the PSO and the SPSO are easy to get into local minima.

In order to show convincingly that the proposed novel PSO algorithm is more superior to other PSO algorithms by comparing AMBPSO with SPSO and PSO, the selection of control parameters is based on a consistency principle, that is, the basic control parameters of the three optimization algorithms are set as the same in this paper. The parameters are set as follows:

Population size $m=40$, Learning rates $c_1=c_2=2.05$, Initial inertia weight $w_{max}=0.9$, Terminal inertia weight $w_{min}=0.35$, Maximal velocity $v_{max}=1$, Minimal velocity $v_{min}=-1$, Maximal position $x_{max}=10$, Minimal position $x_{min}=0.05$, Inertia weight coefficient $H=0.5096$, Mutation Threshold $\rho=0.0001$, Mutation factor of inertia weight coefficient mutation $\gamma=0.02$, Mutation factor of position $\alpha=0.0005$, Mutation factor of velocity $\beta=0.0005$, Maximal iteration numbers $maxgen=100/500/1000$.

Initial speeds and positions in the improved particle swarm optimization algorithm are generated randomly in terms of actual optimization problems. At the same time, the numerical experiments are done on three different maximum iteration numbers. The results are illustrated by only using the fitness graphs at $maxgen=100$.

The curves of the best fitness variations with iterations generated by three PSO algorithms as to the two test functions are shown respectively in Fig.1 and Fig.2. The dimensions of the two selected functions are both set as $n=20$. In order to avoid contingency, 50 times of optimization experiments are done for all the algorithms to analyze and compare the performances of the three PSO algorithms. From Fig. 1 and 2, it can be obviously shown that The AMBPSO is of higher convergence speed than the other two algorithms for both of the selected test functions during the process of iterations.

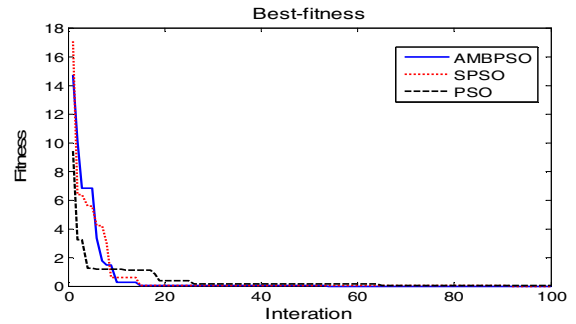


Fig. 1 Fitness curves of Generalized Rastrigin function

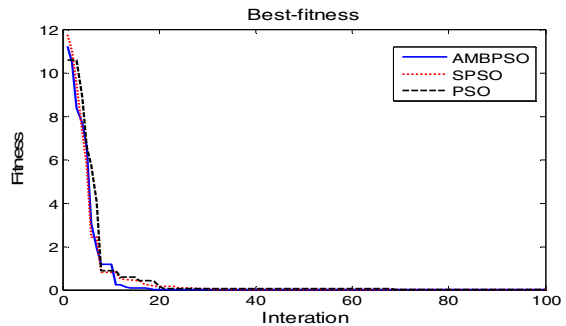


Fig. 2 Fitness curves of Ackley function

As shown in Table 1 are root mean square errors between real optimal values and calculated values by the three PSO algorithms as to the two given test functions at three different terminal iterations.

TABLE I. RMS ERRORS OF OPTIMAL VALUES FOR EACH TEST FUNCTION

	Generalized Rastrigin			Ackley		
	AMBPSO	SPSO	PSO	AMBPSO	SPSO	PSO
100	3.07×10^{-4}	4.20×10^{-3}	3.86×10^{-2}	6.22×10^{-6}	2.80×10^{-3}	1.92×10^{-2}
500	7.83×10^{-9}	1.14×10^{-4}	6.60×10^{-3}	8.64×10^{-10}	7.34×10^{-5}	4.30×10^{-3}
1000	8.88×10^{-13}	4.44×10^{-7}	1.21×10^{-4}	3.14×10^{-13}	1.52×10^{-9}	9.24×10^{-5}

From Table 1, it can be shown that the search precision increases with the increment of the maximal number of iterations given in advance, and that under the same terminal

condition the AMBPSO algorithm is of much higher precision and faster convergence rate than the other two PSO algorithms for whichever standard test functions.

V. CONCLUSIONS

In this paper an adaptive mutation-based PSO (AMBPSO) is presented for improvement of the deficiencies of standard PSO which is modified by the combination of dynamic adjustment of the inertia weights, update of position and velocity of each particle by means of randomly adaptive mutation, and the limit of the update for the change in a reasonable range. The optimization results of two standard test functions show that these modifications can enhance particles' activity to improve the algorithm's search precision and convergence speed and to keep away from easily immersing in local minima efficiently.

REFERENCES

- [1] J. Kennedy and R. Eberhart, Particle swarm optimization, in Proc. IEEE Int. Conf. Neural Network, 1995, vol. 4, pp. 1942-1948.
- [2] Zhen Ji, Huilian LIAO, Q. H. WU. Particle Swarm Optimization algorithm and application. Beijing: Science, 2009.
- [3] Ying Song, Zengqiang Chen, Zhuzhi Yuan et al. New Chaotic PSO-Based Neural Network Predictive Control for Nonlinear Process. IEEE Transactions on Neural Networks, 2007, vol. 8, no. 2, pp. 595-601.
- [4] Guo-Dong Li, Shiro Daisuke Yamaguchi et al. An Optimal Grey PID Control System. IEEE Transactions on Electrical and Electronic Engineering, 2009, vol. 4, no. 4, pp. 570-577.
- [5] H. Xiaohui, S. Yuhui, and R. Eberhart, Recent advances in particle swarm, Proc. Congr. Evol. Comput., 2004, vol. 1, pp. 90-97.
- [6] R. C. Eberhart and Y. Shi, Guest editorial, IEEE Trans. Evol. Comput. (Special Issue on Particle Swarm Optimization), 2004, vol. 8, no. 3, pp. 201-203.
- [7] Y. Shi and R. Eberhart, A modified particle swarm optimizer, Proc. IEEE World Congr. Comput. Intell., 1998, pp. 69-73.
- [8] M. R. AlRashidi, M. E. El-Hawary, A survey of particle swarm optimization applications in electric power systems, IEEE Transaction on Evolutionary Computation. 2009, vol. 13, no. 4, pp. 913-918.
- [9] Senthil Arumugam M, Rao MVC, Chandramohan A. A new and improved version of particle swarm optimization algorithm with global-local best parameters. Knowledge and Information Systems, 2008, vol. 16, no. 3, pp. 15-26.
- [10] H. M. Soliman, M. A. Awadallah, M. Nadim Emira et al. Robust Controller Design For Active Suspensions Using Particle Swarm Optimization[J]. International Journal of Modeling, Identification and Control, 2008, vol. 5, no. 1, pp. 66-76.
- [11] Zhen Ji, Huilian LIAO, Yiwei WANG, Q. H. WU. A Novel Intelligent Particle Optimizer for Global Optimization of Multimodal Functions. IEEE Congress on Evolutionary Computation(CEC 2007), 2007, pp. 3272-3275.
- [12] M. Senthil Arumugam, M.V.C.Rao. Novel hybrid approach for Real Coded Genetic Algorithm to Compute the Optimal Control of a single stage Hybrid Manufacturing Systems, International Journal of Computational Intelligence, 2005, vol 1, no. 3, pp. 231- 249.
- [13] Omkar SN, Mudigere D, Narayana Naik G, et al. Vector evaluated particle swarm optimization for multiobjective design optimization of composite structures. Computers and Structures, 2008, vol. 86, no. 12, pp. 45-56.
- [14] Shi B, Li YX, Yu XH, et al. A modified particle swarm optimization and radial basis function neural network hybrid algorithm model and its application. WRI Global Congress on Intelligent Systems(GCIS 2009), 2009, vol. 1, pp. 134-138.