A Dynamically Quantum Particle Swarm Optimization Algorithm with Adaptive Mutation

Chenyang Gao^{1*}, Ning Chen¹, Yuelin Gao², Jiajiang Zhang²

¹ School of Information Science and Engineering, Central South University, Changsha, China. ² Institute of Information and System Science, Beifang University of Nationalities, Yinchuan, China.

* Corresponding author. Email: 395346665@qq.com Manuscript submitted July 10, 2015; accepted January 25, 2016. doi: 10.17706/jsw.11.4.338-346

Abstract: An Dynamically Quantum Particle Swarm Optimization Algorithm with Adaptive Mutation (AMDQPSO) is given, the algorithm can better adapt to the problem of the complex nonlinear optimization search. The concept of the evolution speed factor and aggregation degree factor are introduced to this algorithm, and the inertia weight was constructed as a function of the evolution speed factor and aggregation degree factor, so that the algorithm has the dynamic adaptability in each iteration. This paper introduces the concept of the rate of cluster focus distance changing, and gives a new perturbations method. When the algorithm is found to sink into the local optimization, the new adaptive mutation operator and mutation probability are implemented at the best position of the global optimization. so that the proposed algorithm can easily jump out of the local optimization. The test experiments with six well-known benchmark functions show that the AMDQPSO algorithm improves the convergence speed and accuracy, strengthens the capability of local research and restrains the prematurity.

Key words: Quantum particle swarm optimization (QPSO), adaptive mutation, the rate of cluster focus distance changing, inertia weight.

1. Introduction

The PSO (Particle Swarm Optimization, PSO) algorithm proposed by Dr. Eberhart and Dr. Kennedy in 1995 is swarm intelligence algorithm for global optimization [1]. The PSO algorithm can be easily implemented and has a faster converges speed, and has few parameters to adjust. However, we found that the PSO algorithm has poor local search ability and has a shortcoming of premature convergence in the study of PSO. To overcome those shortcomings, a lot of research work has been done by many scholars, and a variety of improved PSO algorithms have been proposed, which have improved the convergence precision and convergence speed in varying degrees. In 2004, inspired by quantum mechanics and trajectory analysis of PSO, a new algorithm called Quantum-behaved Particle Swarm Optimization (QPSO) is proposed [2]. The QPSO also can be easily implemented and has a faster converges speed, and has better global search ability. Since its emergence, QPSO has attracted the attention of several researchers all over the word. Compared with PSO algorithm, although QPSO algorithm has better global search ability, but QPSO also has the same problem of premature convergence with other intelligent evolutionary algorithm. To solve existing problems of the QPSO algorithm, the literature [3]-[6] have proposed different improvement strategies, which have improved the convergence speed in varying degrees.

This paper presents a dynamic quantum particle swarm optimization algorithm with adaptive mutation (AMDQPSO). In this algorithm, we have introduced the concept of evolution speed factor and aggregation degree factor, and the inertia weight β was constructed as a function of the evolution speed factor and aggregation degree factor. so that the algorithm has the dynamic adaptability in each iteration. The QPSO algorithm easy to fall into the local optima, when the algorithm evolute to the last time. So the concept of the population fitness variance and the rate of change focusing distance are introduced in this paper, and we give a new disturbance method. In order to help the algorithm jumped out the local optima, when the algorithm trapped in the local optima, the global optima was changed with adaptive mutation. The test experiments with six well-known benchmark functions show that the AMDQPSO algorithm can improve the convergence speed and accuracy, strengthen the capability of local research and restrain the prematurity.

2. QPSO Algorithm

In 2004, Sun Jun proposed the QPSO algorithm [2]. In the quantum physics, the state of a particle with momentum and energy can be depicted by its wave-function $\Psi(x,t)$. In the quantum space, the probability distribution function of the particle's position can be calculated through the probability density function. By employing the Monte Carlo method, the particle's position is updated according to the following equation:

$$p_{id}(t) = \varphi_{id}(t)P_{id}(t) + (1 - \varphi_{id}(t))P_{gd}(t)$$
(1)

$$X_{id}(t+1) = p_{id}(t) \pm \frac{L_{id}(t)}{2} \times \ln(\frac{1}{\mu_{id}(t)})$$
(2)

where $\varphi_{id}(t)$ and $\mu_{id}(t)$ are a random number uniformly distributed in [0,1], $p_i(t) = (p_{i1}(t), p_{i2}(t), ..., p_{iD}(t))$ is a local attractor, $P_i(t) = (P_{i1}(t), P_{i2}(t), ..., P_{iD}(t))$ is the personal best position, $P_g(t) = (P_1(t), P_2(t), ..., P_D(t))$ is the global best position.

In the update equation of the particle states, the control method of the $L_i(t)$ weakening to zero is vital to the convergence rate and performance of the algorithm.

The literature [7] proposed a more intelligent method, which is called "mainstream thought", to evaluate the parameter $L_i(t)$. The mainstream thought point or mean best position (mbest) is defined as the center of pbest position the swarm. That is

$$mbest(t) = (mbest_{1}(t), mbest_{2}(t), ..., mbest_{D}(t)) = \frac{1}{N} \sum_{i=1}^{N} P_{i}(t)$$

$$= (\frac{1}{N} \sum_{i=1}^{N} P_{i1}(t), \frac{1}{N} \sum_{i=1}^{N} P_{i2}(t), ..., \frac{1}{N} \sum_{i=1}^{N} P_{iD}(t))$$
(3)

Therefore, the parameter $L_i(t)$ is given by

$$L_{id}(t) = 2\beta \times \left| mbest_d(t) - X_{id}(t) \right|$$
(4)

where parameter is called contraction-expansion coefficient, which controls the convergence speed of the QPSO algorithm. The most commonly used control strategy of β is to initially setting it to 1.0 and reducing it linearly to 0.5.

3. QPSO Algorithm

3.1. Evolution Speed Factor and Aggregation Degree Factor [7]

In QPSO algorithm, compression - expansion factor β is the only parameter that needs to be determined, and plays an important role in the convergence process of the algorithm. Generally, the parameter decreases from 1.0 to 0.5 [8]. In the literature [7], the concept of the evolution speed factor and aggregation degree factor are given.

$$Sd = \frac{F(P_g(t))}{F(P_g(t-1))}$$
(5)

The formula (5) expressing of the evolution speed factor

$$Id = \frac{F(P_g(t))}{M(t)}$$
(6)

The formula (6) expressing of the aggregation degree factor. Where $F(P_g(t))$ and $F(P_g(t-1))$ represent the fitness of the global best position, current generation and previous generation. M(t) is the average fitness value of all the particles of the current best position, it is calculated as

$$M(t) = \frac{1}{N} \sum_{i=1}^{N} F(P(t))$$
⁽⁷⁾

By their definition, we can know the parameter *Sd* and *Jd* all belong to the interval (0, 1]. The evolution speed factor *Sd* reflects the evolution speed of the particle , the smaller of the value, the faster of the speed of evolution, when the value of *Sd* always equal to 1, which shows that the algorithm stopped or found the optimal solution; The parameter *Jd* not only reflects the current degree of aggregation of all the particles, also reflects the diversity of the particles. The value of the *Jd* is more bigger ,the swarm has a greater degree of the aggregation and smaller diversity .

In this paper, to making the algorithm to better adapt to the complex, high-dimensional, nonlinear optimization problem, we introduced the inertia weight adjustment strategy of the reference [10].

$$\beta = \beta_0 - Sd \times \beta_1 + Jd \times \beta_2 \tag{8}$$

where β_0 is the initial value of the parameter β , usually $\beta=1$, β_1 and β_2 are inertia weight, its value are $\beta_1=0.5$ and $\beta_2=0.2$.

3.2. Algorithm Premature Judgments and Treatment Methods

With the population continues to evolve, the differences between the every individuals has become more and more smaller, resulting in the algorithm prematurity. The paper [9] proposed a judgment method to determine whether the population premature.

Definition 1 if δ^2 is the fitness variance of the swarm, then δ^2 is defined as:

$$\delta^{2} = \sum_{i=1}^{N} \left(\frac{f_{i} - f_{avg}}{f} \right)^{2}$$
⁽⁹⁾

where *N* is the size of the population, f_i is the fitness value of the i-th individual, f_{avg} is the average fitness value of all the population individuals. f_i is a normalized scale factor, which is to limit the size of δ^2 , the value of f:

$$f = \begin{cases} \max_{1 \le i \le N} \left| f_i - f_{avg} \right| & \max_{1 \le i \le N} \left| f_i - f_{avg} \right| > 1 \\ 1 & others \end{cases}$$
(10)

When $\delta^2 < C$ (*C* is a given value and tends to 0), we can assume that the algorithm goes into a late search stage, prone to premature convergence.

Definition 2 The rate of cluster focus distance changing: if *k* is the rate of cluster focus distance changing, D_{max} and D_{avg} represent the maximum and average focus distance , then *k* is defined as:

$$k = \frac{D_{\max} - D_{avg}}{D_{\max}}$$
(11)

$$D_{\max} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{D} (P_g - P_{ij})^2}$$
(12)

$$D_{avg} = \max_{i=1,2,\dots,N} \sqrt{\sum_{j=1}^{D} (P_g - P_{ij})^2}$$
(13)

During QPSO algorithm operation, when the population fitness variance tends to zero and the rate of cluster focus distance changing becomes more and more smaller, the particles fall into the local convergence. In order to effectively improve the algorithm, we give a new mutation operator. Based on the above analysis we propose a new adaptive mutation probability P_m

$$p_m = \begin{cases} \exp(-k) / 4 & \delta^2 < \delta_0 \\ 0 & \delta^2 \ge \delta_0 \end{cases}$$
(14)

where δ_0 is a given value, usually it tends to 0. Generate random numbers $r \in [0,1]$, if $r < P_m$, given a random perturbations for the global optimum value with a new way of disturbance.

$$P_{p} = P_{p} (1 - 0.03 * rand) \tag{15}$$

3.3. AMDQPSO Algorithm Steps

The steps of AMDQPSO algorithm are as follows:

Step 1. set parameter, Sd=0, Jd=0, $\beta_1=0.5$ and $\beta_2=0.2$.

Step 2. Initialize the position vector of particles, the optimal value of the individual and the global optimum value, for each particles, evaluate its fitness value.

Step 3. whether the algorithm reaches the maximum number of iterations, if the algorithm iterations is reached going to step 8, otherwise going to step 4.

Step 4. Evaluate the value of the *Sd* and *Jd* based on the (5) and (6). Evaluate the value of the β according (8).

Step 5. Update the location of all the particles, evaluate its fitness value, update personal best value and the global optimum.

Step 6. Evaluate the value of the δ^2 , k and P_m according (9), evaluate the value of the k according (11), and evaluate the value of the P_m according (14).

Step 7. Generate random numbers $r \in [0,1]$, if $r < P_m$, then perform mutation operation according (15), otherwise loop to step 3.

Step 8. Output the global optimum value and the fitness value.

4. Experiments and Results

4.1. Test Functions

The six well-known benchmark functions [10] are used as test functions, which are as follows:

• $f_1(x)$: Branin

$$f_1(x) = (x_2 - \frac{5 \cdot 1}{4\pi^2} x_1^2 + \frac{5}{4\pi} x_1 - 6)^2 +$$

$$10(1 - \frac{1}{8\pi})\cos(x_1) + 10$$
(16)

where $-5 \le x_1 \le 10$ and $0 \le x_2 \le 15$, the global optimum is at $x^* = (-3.142, 12.275), (3.142, 2.275), (9.425, 2.425)$ where $f_1(x^*) = 0.398$.

• $f_2(x)$: Schaffer's F6 Function

$$f_2(x) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$$
(17)

where $|x_i| \le 1$. The global optimum is at $x^* = (0,0)$, where $f_2(x^*) = 0$.

• $f_3(x)$: Sphere Model

$$f_3(x) = \sum_{i=1}^{D} x_i^2$$
 (18)

where $|x_i| \le 1$. The global optimum is at $x^* = (0,0)$, where $f_3(x^*) = 0$.

• $f_4(x)$: Generalized Rastrigin's Function

$$f_4(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10], \quad |x_i| \le 5.12$$
(19)

Where $|x_i| \le 5.12$. The global optimum is at $x^* = (0,0)$, where $f_4(x^*) = 0$.

• $f_5(x)$: Ackley's Function

$$f_{5}(x) = -20 \exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_{i}^{2}}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_{i}) + 20 + e)$$
(20)

where $|x_i| \le 32$. The global optimum is at $x^* = (0, 0)$, where $f_5(x^*) = 0$.

• $f_6(x)$: Generalized Griewank Function

$$f_6(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1, \ \left| x_i \right| \le 600$$
(21)

Volume 11, Number 4, April 2016

where $|x_i| \le 600$. The global optimum is at $x^* = (0,0)$, where $f_6(x^*) = 0$.

4.2. The Design of Parameters

For the experiments, we used the following parameters: number of particles *N*=0, $\beta_0 = 1$, $\beta_1 = 0.5$, $\beta_2 = 0.2$, $C = 3 \times 10^6$, the inertia weights $\beta = 1$ decreases from 1.0 to 0.5 with evolution, the dimension of test functions include 2 and 30.

4.3. Discussion of Results

Table 1. Comparison Results by Experiment for $f_1(x)$ and $f_2(x)$						
Function	Solution	AMDQPSO	PSOCF	PSORW	PSOTC	PSONW
	Best	0.3979	0.3979	0.3979	0.3979	0.3979
$f_1(x)$	Mean	0.3979	0.3979	0.3979	0.3979	0.3987
	St. Dev	1.4156e-5	4.4693e-5	3.6656e-5	8.1641e-4	4.8221e-4
	Best	0	3.1500e -7	2.2637e-7	9.6383e-7	2.0763e -7
$f_2(x)$	Mean	0	0.0053	0.0042	0.0086	0.0044
	St. Dev	0	0.0048	0.0048	0.0026	0.0047

Table 2. Comparison Results by Experiment for $f_1(x)$ and $f_2(x)$

Function	Solution	AMDQPS	PSOCR	PSOCW	NPSO1	NPSO2
		0				
	Best	0.3979	0.3979	0.3979	0.3979	0.3979
$f_1(x)$	Mean	0.3979	0.3984	0.3979	0.3987	0.3979
	St. Dev	1.4156e-5	4.8221e-4	4.5790e-5	4.2979e-4	4.2236e-4
	Best	0	2.3110e-4	2.8296e-8	1.6866e-7	2.2946e-8
$f_2(x)$	Mean	0	0.0082	0.0013	7.9980e-4	2.0430e-4
	St. Dev	0	0.0028	0.0033	0.0027	0.0014

Table 3. Comparison results by Experiment for $f_3(x)$, $f_4(x)$, $f_5(x)$ and $f_6(x)$

Function	Solution	AMDQPSO	PSOCF	PSORW	PSOTC	PSONW
<i>f</i> ₃ (<i>x</i>)	Best	4.4872e-57	6.6004e-6	7.3585e-6	7.8768e+3	6.9023
	Mean	1.2425e-50	9.4159e-6	9.4340e-6	1.0508e+4	712.7469
	St. Dev	8.1529e-50	5.8005e-7	5.8001e-7	1.0727e+3	897.2576
	Best	10.9446	31.8387	25.8689	208.3870	33.6796
$f_4(x)$	Mean	21.6800	78.1592	66.3920	248.7455	67.0598
	St. Dev	6.8672	24.6041	18.4993	13.9451	18.5200
<i>f</i> ₅ (<i>x</i>)	Best	4.409e-15	9.8306e-6	9.6763e-6	14.0627	0.0231
	Mean	0.1734	2.7624	2.4637	15.6602	5.1710
	St. Dev	0.4759	1.1450	1.2705	0.4188	2.2567
<i>f</i> ₆ (<i>x</i>)	Best	0	6.8538e-6	8.4503e-6	68.9545	0.3653
	Mean	0.0554	0.0893	0.0429	97.0943	8.4631
	St. Dev	0.4759	0.1775	0.0771	9.4887	7.1134

We performed 100 independent runs for each test problem, the results obtained the "Best", "Mean" and "St.Dev". We compared our approach against eight approaches [10]. All the experiments were executed on a personal computer with the Core i7 dual-core processor, 4GB internal memory and Windows 7 operating system.

As described in Table 1 and Table 2, $f_1(x)$ is a simple 2-dimension test function, all improved approach

was able to find the global optimum; $f_2(x)$ is a complicated 2- dimension test function. In the nearby of the global optimum value, there are a lot of local minimum, so the algorithm is difficult to find the global optimum.

Table 4. Comparison Results by Experiment for $f_3(x)$, $f_4(x)$, $f_5(x)$ and $f_6(x)$						
Function	Solution	AMDQPSO	PSOCR	PSOCW	NPSO1	NPSO2
	Best	4.4872e-57	4.4455e+3	5.5001e-5	7.4440e-6	7.6941e-6
$f_3(x)$	Mean	1.2425e-50	6.0764e+3	10.1458	9.3597e-6	9.4682e-6
	St.Dev	8.1529e-50	704.9375	14.9335	6.0096e-7	4.9516e-6
	Best	10.9446	179.5670	17.1001	21.8891	21.8891
$f_4(x)$	Mean	21.6800	233.3484	65.1484	55.7506	53.2171
	St.Dev	6.8672	17.4131	26.3254	22.8229	20.5359
	Best	4.409e-15	11.7022	0.0010	8.4937e-6	8.6209e-6
$f_5(x)$	Mean	0.1734	13.6514	1.6272	0.4596	0.4867
	St.Dev	0.4759	0.4999	1.2498	1.5204	1.5098
	Best	0	32.9148	9.6178e-5	8.1673e-6	6.9701e-6
$f_6(x)$	Mean	0.0554	56.6829	0.7671	0.0128	0.0112
	St.Dev	0.1119	6.2559	0.5324	0.0144	0.0159

However, our approach can find the global optimum, other approaches failed to achieve the optimal solution, it shows that our approach has a good performance in the local search.

Results in the Table 3 and Table 4 shows that, $f_3(x)$ is a simple 30-dimension test function. Those algorithm of the PSOCF, PSORW, NPSO1and NPSO2 only found the solutions very close to the global optimum in the test function $f_3(x)$, but our approach AMDQPSO has a very high accuracy. For those function $f_4(x)$, $f_5(x)$ and $f_6(x)$, those test function are high-dimensional and complex, so it has a great difficult to achieve the global optimum. In the test function $f_4(x)$ and $f_5(x)$, our approach has a better result than other approach. For the function $f_6(x)$, even though the mean fitness value running 100 times is worse than the algorithm NPSO2, but our approach has found the global optimum. According to Table I and Table II, we found the "St.Dev" result for all the test functions were very good. This shows that our algorithm has better robustness.

5. Conclusion

The purpose of this paper was to present a novel algorithm to solve the complex optimization problem.We proposed a novel algorithm called Dynamically Quantum Particle Swarm Optimization Algorithm with Adaptive Mutation. For the inertia weight, we make it dynamic adjustment. Furthermore, the premature judgment mechanism was added. Additionally, we introduced the concept of the rate of cluster focus distance changing. Finally, the new adaptive mutation operator and mutation probability are implemented at the best position of the global optimization. So that the proposed algorithm can easily jump out of the local optimization. Although our algorithm showed better performance, but it still needs further study.

Acknowledgment

The work is supported by the Foundations of National Natural Science China (61561001) and by the key research projects of Beifang University of Nationality (2015KJ10).

References

[1] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of the IEEE international

Conference on Neural Networks (pp. 1942-1948).

- [2] Sun, J., & W. B. Xu (2004). Particle swarm optimization with particles having quantum behavior. *Proceedings of the IEEE Congress Evolutionary Computation* (pp. 325-331).
- [3] Sun, J., Fang, W., Wu, X., Palade, V., & Xu, W. (2012). Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection. *Evol. Comput.*, *20*(*3*), 349-393.
- [4] Li, W. L., Wang, Z. S., & Song, X. J. (2012). Parameters optimization of support vector machine based on simulated annealing and improved QPSO. *Proceedings of the Industrial Control and Electronics Engineering* (pp.1089-1091).
- [5] Li, Z. Y., Xu, K., Liu, S. B., & Li, K. L. (2008). Quantum multi-objective evolutionary algorithm with particle swarm optimization method. *Proceedings of the IEEE Fourth International Conference on Natural Computation* (pp. 672-676).
- [6] Gao, H., Xu, W. B., Sun, J., & Tang, Y. L. (2010). Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm. *IEEE Trans. Instrum. Meas.*, *59*(*4*), 934-946.
- [7] Zhang, X. P., Du, Y. P., Qin, G. Q., & Tan, Z. (2005). Adaptive particle swarm algorithm with dynamically changing inertia weight. *Journal of Xi'an Jiaotong University*, *39(10)*, 41-43. China.
- [8] Sun, J., Xu, W. B., & Liu, J. (2005). Parameter selection of quantum-behaved particle swarm optimization. *Advances in Natural Computation, Lecture Notes in Computer Science*, *3612*, 543-552.
- [9] Liu, J. F., & Gao, Y. L. (2011). Quantum particle swarm optimization algorithm with adaptive mutation. *Computer Engineering and Application*, *47(3)*, 41-43.
- [10] Liang, H., liu, Y. G., Xu, R. N., & Su, R. J. (2010). A new improved particle swarm optimization method. *Computer Engineering and Application*, *46*(*12*), 38-41.



Chenyang Gao was born in 1992. He received his master's degree in School of Information Science and Engineering, Central South University, Changsha, China. His research interests include intelligent information processing and intelligent control.



Ning Chen was born in 1970. She is a Ph. D., a professor, a Ph. D. supervisor. Her main research interests currently include stability analysis of complex system, decentralized robust control, distributed coordination control and advanced energy-saving control technology.



Yuelin Gao was born in 1963, He is a Ph. D., a professor, a Ph. D. supervisor. His main research interests currently include theory, methods and applications of optimization, intelligent information processing, financial computation and financial engineering.



Jiajiang Zhang was born in 1989. He received his master's degree in Institute of Information and System Science, Beifang University of Nationalities, Yinchuan, China. His research interests include intelligent computing and applications.