A Novel Cultural Quantum-behaved Particle Swarm Optimization Algorithm

X. Z. Gao[2], Ying Wu[1], Xianlin Huang[1] and Kai Zenger[2]

[1] Center for Control Theory and Guidance Technology, Harbin Institute of Technology, 150001 Harbin, China
[2] Department of Automation and Systems Technology, Aalto University School of Electrical Engineering, 00076 Aalto, Finland
xiao-zhi.gao@aalto.fi, wying_hit@hotmail.com, xlinhuang@hit.edu.cn, kai.zenger@ aalto.fi

Abstract

A novel cultural quantum-behaved particle swarm optimization algorithm (CQPSO) is proposed to improve the performance of the quantum-behaved PSO (QPSO). The cultural framework is embedded in the QPSO, and the knowledge stored in the belief space can guide the evolution of the QPSO. 15 high-dimensional and multi-modal functions are employed to investigate the proposed algorithm. Numerical simulation results demonstrate that the CQPSO can indeed outperform the QPSO.

Keywords: *Quantum-behaved; PSO; Cultural Framework; Knowledge*

1. Introduction

The Particle Swarm Optimization (PSO) is a population-based optimization method, which was firstly developed by Eberhart and Kennedy in 1995. It is inspired by the social behaviors of animals and insects, such as bird flocking or fish schooling [1]. The distinguishing features of the PSO are its computation efficiency and algorithm simplicity. Unfortunately, the PSO might be stuck into local optima when dealing with multi-modal optimization problems. Numerous approaches have been introduced to enhance the optimization capability of the PSO [2, 3]. Recently, one of the novel hybridization for PSO is to apply the Quantum laws of mechanics to observe its behavior---Quantum PSO (QPSO), which has less parameter to control [4].

A novel optimization method namely Cultural Algorithm (CA) proposed by Reynolds in 1995 is a powerful solution to demanding problems, due to its flexibility and efficiency [5]. The CA is a class of computational models derived from the principles of the culture evolution in nature, and can be viewed as a dual inheritance system. In the CA, the evolution takes place in the population space under the macro-evolutionary level. Various evolutionary algorithms have been utilized in the population space of the CA [6, 7].

In this paper, a novel cultural quantum-behaved Particle Swarm Optimization, CQPSO, is proposed to improve the convergence performance of the QPSO. In the CQPSO, certain proportion of the particles in the swarm mutate based on the influence function. The mutation operator and CA can work together to increase the diversity of the swarm population, and enhance the global search capability of the QPSO. A total of four variants of the CQPSO are investigated.

The rest of this paper is organized as follows. Sections 2 briefly introduce the background knowledge of the PSO and QPSO. Section 3 proposes and discusses the underlying principle of the CQPSO. In Section 4, the optimization performance of our CQPSO is further examined using fifteen high-dimensional and multi-peak functions.

2. Basic Particle Swarm Optimization and the QPSO

2.1. Basic Particle Swarm Optimization

The basic principle of the PSO method can be explained as follows: suppose there are N particles in the particle swarm, which are initialized randomly. Each particle can fly in the D-dimension search space according to its own velocity $V_i = (v_{i1}, v_{i2}, .. v_{iD})$. The particles are associated with their positions $X_i = (x_{i1}, x_{i2}, .. x_{iD})$ standing for the possible solutions to the problems under consideration. During the iterations, every particle can update the position on the basis of the previous best position $P_i = (p_{i1}, p_{i2}, .. p_{iD})$ and global best position Pg_d of the whole swarm. The update of these particles is:

$$v_{id}^{t+1} = \chi(wv_{id}^t + c_1 r_1 (p_{id} - x_{id}^t) + c_2 r_2 (Pg_d - x_{id}^t)) . \quad \textbf{(1)}$$

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

where w is an inertial factor which is employed to balance the local and global search abilities of the PSO [8]. c_1 and c_2 are two learning factors. r_1 and r_2 are two random numbers uniformly distributed in the interval of (0,1). χ is a constriction factor used to limit the maximum velocity value [9].

2.2. Quantum-behaved Particle Swarm Optimization

In the QPSO, all the particles have the quantum behavior. The state of a particle in QPSO is stated by wavefunction $|\psi(x,t)|^2$ [10]. The particles move according to the following formulations:

$$X_i^{t+1} = \begin{cases} M_i^t + belta * \left| X_{mbest}^t - X_i^t \right| * \ln(1/u) & if \;\; k \ge 0.5 \\ M_i^t - belta * \left| X_{mbest}^t - X_i^t \right| * \ln(1/u) & if \;\; k < 0.5 \end{cases} . \quad \textbf{(3)}$$

$$X_{mbest}^t = \frac{1}{N} \sum_{i=1}^N X_i^t , \;\; X_i^t = (x_{i1}^t, x_{i2}^t \cdots x_{iD}^t) , \;\; M^t = \frac{c_1 * x^t + c_2 * Pg^t}{c_1 + c_2} . \quad \textbf{(4)}$$

where belta is the contraction-expansion coefficient, *u* and *k* are uniformly random number.

4. Cultural Quantum-behaved Particle Swarm Optimization

As proposed by Reynolds, the CA is composed of population space, belief space and the communication protocol [5]. The belief space is the place, where cultural knowledge is formed and stored. In this paper, two typical kinds of knowledge are used: situational knowledge, normative knowledge.

The acceptance function determines which individuals and their performances can have impact on the knowledge in the belief space. The number of the individuals accepted for the update of knowledge is obtained by the following function:

$$f_a(N,t) = N \cdot \beta + \lfloor N \cdot \beta / t \rfloor . \quad \textbf{(8)}$$

In our previous work, we concentrate on how to combine the cultural framework and the particle swarm optimization. Although some better results can be obtained not only in the function optimization but also in some real applications, there is no property way to guide the iteration for the velocity in the population. The emergency of the quantum-behaved particle swarm optimization can solve this problem, because the position is the only iteration term in the QPSO. In this paper, four kinds of influence functions are utilized to decide the iteration for the QPSO to improve the performance of the QPSO.

If the normative knowledge is used to determine the size of the mutation change, our CQPSO is named as CQPSO (Ns). The corresponding influence function is defined as:

$$x'_{i,j} = x_{i,j} + size(I_i) \cdot N(0,1) . \quad \textbf{(9)}$$

where $size(I_i) = u_i - l_i$ is the size of the belief interval for the i^{th} variable, and $N(0,1)$ is a normally distributed random variable.

If the situational knowledge is used to guide the direction of the mutation, our CQPSO is named as CQPSO (Sd). The corresponding influence function is defined as:

$$x'_{j,i} = \begin{cases} x_{j,i} + \left|\sigma_{j,i} \cdot N(0,1)\right| & if\ x_{j,i} < s_i \\ x_{j,i} - \left|\sigma_{j,i} \cdot N(0,1)\right| & if\ x_{j,i} > s_i \\ x_{j,i} + \sigma_{j,i} \cdot N(0,1) & otherwise \end{cases} \qquad \sigma_{j,i} = \sqrt{f(x_{j,i})} . \quad \textbf{(10)}$$

where $\sigma_{j,i}$ represents the individual mutation step size for the ith variable of the jth individual. As a general rule, $\sigma_{j,i}$ is set to the square root of the fitness value of $x_{j,i}$. s_j is the best exemplar value for variable i in the belief space.

If the normative knowledge and situational knowledge are used to determine the size of the mutation change and direction of the mutation respectively, our CQPSO is named as CQPSO (NsSd). The corresponding influence function is defined as:

$$x'_{j,i} = \begin{cases} x_{j,i} + \left|size(I_i) \cdot N(0,1)\right| & if\ x_{j,i} < s_i \\ x_{j,i} - \left|size(I_i) \cdot N(0,1)\right| & if\ x_{j,i} > s_i \\ x_{j,i} + size(I_i) \cdot N(0,1) & otherwise \end{cases} . \quad \textbf{(11)}$$

If the normative knowledge is used to determine both the size and direction of the mutation, our CQPSO is named as CQPSO (NsNd). The influence function is given as:

$$x'_{j,i} = \begin{cases} x_{j,i} + \left|size(I_i) \cdot N(0,1)\right| & if\ x_{j,i} < l_i \\ x_{j,i} - \left|size(I_i) \cdot N(0,1)\right| & if\ x_{j,i} > u_i \\ x_{j,i} + \beta \cdot size(I_i) \cdot N(0,1) & otherwise \end{cases} . \quad \textbf{(12)}$$

The iteration steps of our CQPSO are described as follows:

1) Initialize N particles in the swarm with random initial positions.

2) Evaluate all the particles using the fitness function.

3) Initialize the belief space.

4) Choose c particles randomly, and mutate them according to a preset influence function, which is employed to determine the mutation based on the knowledge stored in the belief space:

$$c = \lfloor N \cdot ratio \rfloor . \tag{13}$$

where *ratio* is the proportion of the particles to be mutated in the population. The mutation proportion *ratio* is not fixed, and it can linearly decrease from 0.8 to 0.2 in the CQPSO. The influence functions used here are explained in the previous section. Evaluate the $2c$ particles using the fitness function f, and randomly select c competitors. Conduct the pairwise competition between the particles and their competitors. Select only c particles that have the largest number of 'wins'.

5). Update the belief space based on the selected acceptance function.

6). Return to step 4) until a termination criterion is satisfied.

5. Simulation Results

A total of 15 nonlinear functions are used to investigate the optimization capability of our CQPSO. All these functions here are multi-modal functions and with 30 dimension, as given in paper [11]. The four versions of our CQPSO are compared with the QPSO. The parameters of algorithms are: $N = 40$, $\chi = 0.729$, $c_1 = c_2 = 2.05$ and belta linearly decrease from 1 to 0.5 with iterations. The optimization results are provided in Table 1. It can be figured that the performance of the CQPSO is much better than that of the QPSO for almost all the functions except for the Sal function and Schwefel function.

Table 1. Function Optimization Performance Comparison

Functions	PSO	CQPSO (NsSd)	CQPSO (NsNd)	CQPSO (Sd)	CQPSO (Ns)
Ackley	1.2436	0.8856	0.2310	1.1877×10^{-4}	0.5860
CM	-0.3453	-1.3899	-2.0098	-2.0676	-1.3456
DeJongf4	0.0246	4.5454×10^{-322}	6.5711×10^{-322}	9.4242×10^{-18}	1.9994×10^{-293}
Expfun	1.2266	1.0000	1.0000	1.0000	1.0000
Griewank	0.0190	1.2212×10^{-16}	6.6613×10^{-17}	1.1102×10^{-17}	1.8874×10^{-16}
Hyperellip	20.9658	7.4660×10^{-277}	1.4426×10^{-273}	6.4493	2.8919×10^{-225}
LM1	0.0962	1.5705×10^{-32}	1.5705×10^{-32}	1.5705×10^{-32}	1.5705×10^{-32}
LM2	1.6626	0.0099	0.0065	0.0011	0.0312
Neumaier	-133.3331	-4930	-4930	-4928.6	-4930
Rastrigin	57.2142	23.3815	22.6850	19.2635	23.8790
Rosenbrock	65.8476	1.1960	0.4036	28.3834	0.8098
Sal	0.2679	0.3199	0.3199	0.3199	0.3399
Schwefel	6462.2058	6791.9461	7.0318568	6312.3102	6514.5500
Schaffer	19.1880	19.0074	14.8900	19.6145	17.0914
Sphere	0.1425	4.0166×10^{-272}	5.6356×10^{-281}	1.4878×10^{-144}	1.2513×10^{-228}

As some illustrative examples, Figures 1-4 show the comparison of convergence performance among the four versions of the CQPSO and QPSO. A logarithmic (base 10) scale is used for the vertical axis. The mean best fitness is the average over 10 independent trials for each algorithm, the number of iterations is 10,000 in each trial. It can be figured out that the QPSO is trapped into a local optimum soon, and the CQPSO combat this well compared with the QPSO.

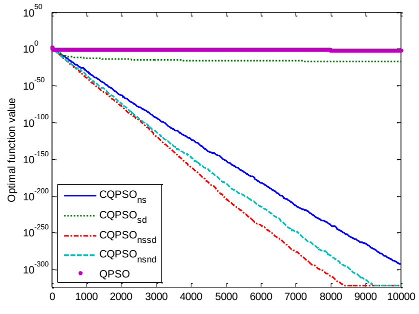

Figure1. Optimization of DeJongf4

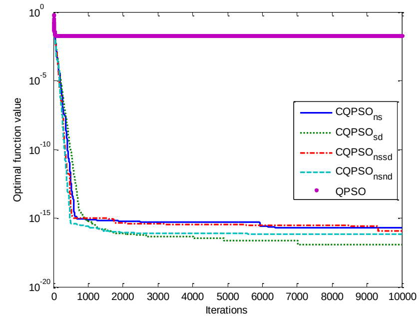

Figure 2. Optimization of Griewank

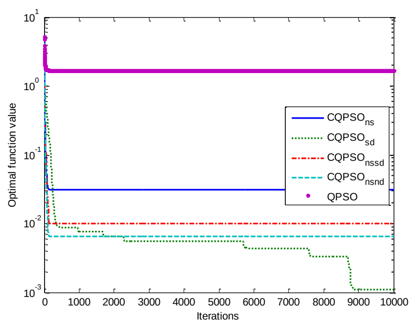

Figure 3. Optimization of LM2

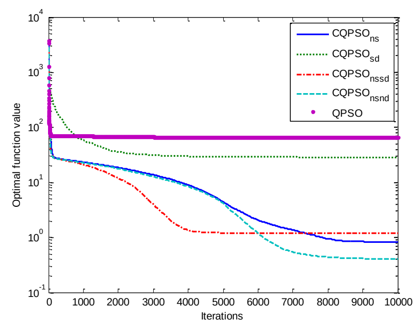

Figure 4. Optimization of Rosenbrock

Acknowledgements

This work is supported by the Academy of Finland under Grants No. 135225 and No. 127299 and the NSFC under Grant No. 60874084. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory", Proceedings of the 6th International Symposium on Micro Machine and Human Science, **(1995)** October 4-6;Nagoya.

[2] R. Battiti and S. Pasupuleti, "The Gregarious Particle Swarm Optimizer (G-PSO)", Proceedings of the ACM Genetic and Evolutionary Computation Conference, **(2006)** July 8-12, Washington DC.

[3] Y. H. Shi and R. C. Eberhart, "Fuzzy Adaptive Particle Swarm Optimization", Proceedings of the IEEE Conference on Evolutionary Computation, **(2001)**, Seoul.

[4] J. Sun, B. Feng and W.B. Xu, "Particle Swarm Optimization with Particles Having Quantum Behavior", Proceedings of the Congress on Evolutionary Computation, **(2004)** June 19-23, San Diego.

[5] R. G. Reynolds, "An Introduction to Cultural Algorithms", Proceedings of the 3rd Annual Conference on Evolutionary Programming, **(1994)** February 24-26, SanDiego.

[6] R. G. Reynolds and C. J. Chung, "Knowledge-based Self-adaptation in Evolutionary Programming Using Cultural Algorithms", Proceedings of the 1997 IEEE International Conference on Evolutionary Computation, **(1997)** April 13-16, Indianapolis.

[7] M. Lovbjerg, T. K. Rasmussen and T. Krink, "Hybrid Particle Swarm Optimizer with Breeding and Subpopulations", Proceedings of the 3rd Genetic and Evolutionary Computation Conference, **(2001)** July 7-11, San Francisco.

[8] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer", Proceedings of the 1998 IEEE World Congress on Computational Intelligence, **(1998)** May 4-9, Anchorage.

[9] M. Clerc, "The Swarm and the Queen: Towards A Deterministic and Adaptive Particle Swarm Optimization", Proceedings of the 1999 Congress on Evolutionary Computation, **(1999)** July 6-9, Washington.

[10] L. Jing, W.B. Xu and J. Sun, "Quantum-behaved Particle Swarm Optimization with Mutation Operator", Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence, **(2005)** November 14-16, Hong Kong.

[11] Y. Wu, X. Z. Gao and K. Zenger, "Knowledge-based artificial fish-swarm algorithm", Proceedings of the 18th IFAC World Congress, **(2011)** August 28 – September2, Milano, Italy.

Authors

X. Z. Gao (1972-)

Professor Gao works in Aalto University. His research focus on soft computing methods and application.

Ying Wu (1984-)

Ph.D student. She studies in Harbin Instute of Technology. Her research focus on natural computing.

Xianlin Huang (1956-)

Professor Huang works in Harbin Institute of Technology. His research focus on System identification and optimal control.

Kai Zenger (1958-)

Professor Zenger works in Aalto University, His research focus on rotor system and control engineering.