2011 Fourth International Joint Conference on Computational Sciences and Optimization

An Intelligent Parameter Selection Method for Particle Swarm Optimization Algorithm

Yuntao Dai
College of Science
Harbin Engineering University
Heilongjiang, China
peach0040@126.com

Liqiang Liu, Ying Li
College of Automation
Harbin Engineering University
Heilongjiang, China
liuliqiang@hrbeu.edu.cn

***Abstract*—For the problem of particle swarm optimization parameters selection, a kind of intelligent method to optimum parameters selection using another particle swarm optimization algorithm is proposed. Firstly it analyzes the effect of each parameter on algorithm performance in detail. Then it takes parameter selection of PSO algorithm as a complex optimization problem, sets appropriate fitness function to describe optimization performance, and uses PSO-PARA algorithm to optimize the parameters selection method of PSO-OPT algorithm. Tests to the benchmark function show that these parameters are better than the experience parameters test results in the optimal fitness, the mean value of optimal fitness, convergence rate.**

***Keywords*-parameter optimization; particle swarm optimization algorithm; analysis of parameters;**

I. Introduction

In optimization algorithms, parameter is one of the key factors which influence algorithms performance and efficiency, so how to determine the optimal parameter to make algorithm performance best is a very complicated optimization problem[1]. For different kinds of optimization problem, parameter selections are different. And even for the same type of optimization problems, if problems scales are different, parameter selections are not completely the same. Particle Swarm Optimization(PSO) algorithm has less parameter to adjust, but the parameters are interrelated and coupling. Then how to determine the optimal combination for different problems is a complex optimization problem to solve[2]. So far, parameter selection of PSO algorithm still does not have perfect theoretical basis, and is mainly given through empirical analysis or a lot of simulation experiments. Mohan and Ozcan firstly analyze particle trajectories change as time changes in one-dimension search space, and give the influence of different ranges of ϕ on particle trajectories[3]. In 1999, they give particle trajectories analysis in multi-dimensional search space[4]. Clerc analyzes the iterative expressions of basic PSO algorithm, and considers random amounts as constants to analyze the state transition matrix, thus the obtains the parameters constraints to make individual particle trajectories convergence[5]. Wang Junwei makes systematic experiments for important parameters, analyzes the choice problem of fixed weight and time-varying weight, and analyzes influence of inertia weight on the algorithm performance from the dependence, population size and topological structure[6]. Gao Shang introduces partial factors test method' into the parameters optimization of PSO algorithm, and describes basic model parameters set problem of PSO algorithm as experimental design with many factors and levels in uniform design[7]. This method still uses simulations to select these parameters, although it gets the desired results with fewer tests by uniform design. Pan Guanyu proposes to determine the PSO optimal parameters using differential evolution, and result is achieved[8]. But this method merely considers the effect of parameter selection on optimal function value, while ignoring other factors. In this paper, we regard parameter selection of PSO algorithm as an optimization problem, consider those important algorithm parameters as the position information of PSO algorithm, and use other standard PSO algorithm for optimizing to obtain the optimal parameters combination for different questions.

II. PARTICLE SWARM OPTIMIZATION

PSO algorithm is firstly introduced by Kennedy and Eberhart in 1995[9][10]. In a PSO system, each individual is called a "particle", and each particle represents a potential solution. In the continuous space coordinates, PSO algorithm is described as:

Set for the D-dimensional search space, Nth particles compose a population, and the ith particle position is $x_i = (x_{i1}, x_{i2}, ..., x_{iD})^T$, the ith particle velocity vector is $v_i = (v_{i1}, v_{i2}, ..., v_{iD})^T$. fitness value of each particle in the current position (measure the merits by the function value for each particle) is described as $fitness_i = fitness(x_i)$, the best previous position of particle i is represented by $P_i = (P_{i1}, P_{i2}, ..., P_{iD})^T$, and the best position among all particles in the population is $P_g = (P_{g1}, P_{g2}, ..., P_{gD})^T$. Each particle adjusts its speed dynamically according to the comprehensive analysis individual and population flying experience, and fly to the best position that it experienced and other particles have. Each particle updates its speed and

978-0-7695-4335-2/11 $26.00 © 2011 IEEE
DOI 10.1109/CSO.2011.79

position according to (1).

$$
\begin{aligned}
v_{id}(t+1) &= wv_{id}(t) + c_1 r_1 [P_{id}(t) - x_{id}(t)] \\
&\quad + c_2 r_2 [P_{Gd}(t) - x_{id}(t)] \\
x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1)
\end{aligned} \tag{1}
$$

where t is iteration time, r_1, r_2 are random between 0 and 1, and c_1, c_2 are nonnegative constants, called learning factor and to adjust each iteration step length.

III. ANALYSIS OF PARAMETERS ON THE ALGORITHM PERFORMANCE

PSO algorithm has some important parameters, such as population size(m), inertia weight(w), learning factor(c_1,c_2) and maximum velocity(vMax). We analyzes parameters of PSO algorithm qualitatively through simulation.

Star structure is elected as PSO algorithm topology, update of particle velocity and location by (1), and each parameter's value and change range are shown in Table I.

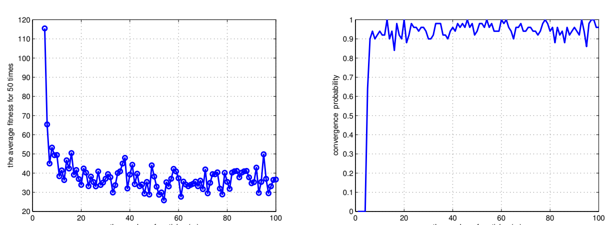
Table I
THE BOUNDARY OF PARAMETERS

parameters	criterion	boundary
Particle's Quantity(m)	20	[1, 100]
inertia weight(w)	0.729	[0, 2]
Learning Factor(c_1,c_2)	1.4962	[0, 3]
Maximum Velocity($vMax$)	$xMax$	[0.1*$xMax$, 10*$xMax$]

In each test, mean and convergence rate changes of optimization function given by PSO algorithm are compared with other parameters fixed and the variation of parameter to be analyzed.

A. Influence of Population Size

We choose 30 dimensions Rosenbrock function for test. Iteration time is set to 1000, repeated computation is 50, and the mean of 50 computations and the convergence rate are output of each parameter change. Particle number increases 1 each time and simulation results are shown in Fig.1.



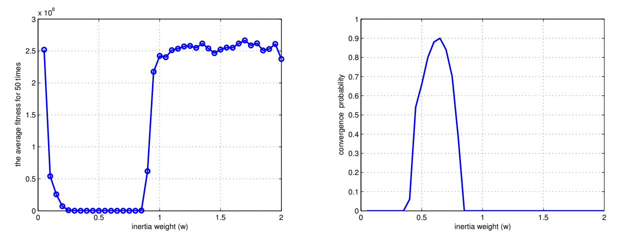
(a) (b)

Figure 1. Influence of population size m on algorithm performance. (a) and (b) shows the relation between population size and mean fitness function, convergence rate, respectively.

When particle number is too small, cooperation mechanism among particles lacks and they share less information. Although iteration is very quick, but global search capability is very weak. When particle number increased to 10, output mean significantly reduced, and optimal rate also increases more than 80% rapidly. But when particle number increases more than 20, the increase of number particles does not significantly increase convergence rate and the global search performance. What's worse, it takes more time.

B. Influence of Inertia Weight

Inertia weight w is used to control the influence of last particle velocity on the current, and its size also affects particle's global and local search capability. On the analysis of influence of inertia weight w on the algorithm performance, we take the values from 0 to 2 (interval is 0.05). The simulation result is showed in Fig.2.



(a) (b)

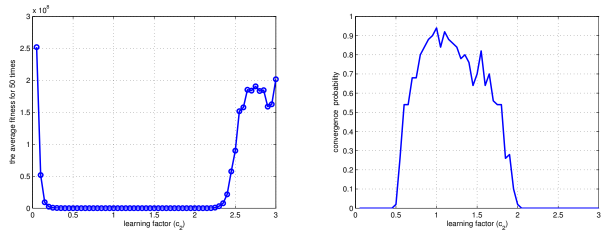
Figure 2. The simulation result of inertia weight w influence on algorithm performance. (a) and (b) shows the relation between inertia weight and mean fitness function, convergence rate, respectively.

Along with the increase of w optimal rate is presented as trend, i.e. low, high, and low again. When w is too small (w<0.4) or too big (w>0.85), algorithm hardly converges. When w is between 0.5 and 0.75, optimal rate is very high, more than 80%. So inertia weight should be appropriate, mostly we set it less than 1.

C. Influence of Learning Factors

Learning factors are used to control the relative influence among particle own memory and companions'. When c_1 is too small, particles lack cognitive ability, but it still can reach the new search space relying on the mutual cooperation between particles. Along with the increase of c_1, particles enhances its own cognitive ability. But when c_1 exceeds a certain value, its cognitive about itself slowly is more than about the society, the global search ability becomes weak, and optimal rate falls gradually. c_2 determines the global search ability of particles, and the simulation result of c_2 is shown in Fig.3.

When c_2 is smaller than 0.5, the convergence rate is almost 0, because particle search relying mainly on the cognitive ability, and share less information. Thus it's difficult to get solution. Along with the increase of c_2, Particle enhances its cognitive ability about society. But when c_2 exceeds a certain value, the particles society's cognition slowly surpass to their own cognitive awareness, and the global search ability becomes weak.

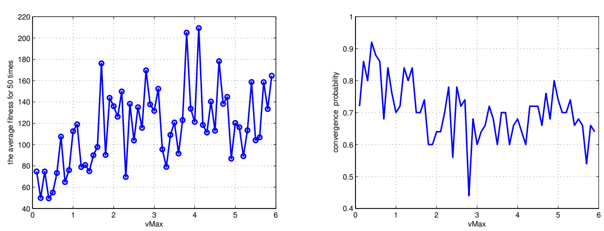


(a) (b)

Figure 3. Influence of learning factor c_2 on algorithm performance.(a) and (b) shows the relation between learning factor c_2 and mean fitness function, convergence rate, respectively.

D. Influence of Maximum Velocity

In this test, the maximum velocity is set as a function of the maximum position value of particle, $vMax = \sigma \cdot xMax$, and σ changes between 0.1-10,interval is 0.1, the simulation results show in Fig.4.



(a) (b)

Figure 4. Influence of maximum velocity on algorithm performance.(a) and (b) shows the relation between maximum velocity and mean fitness function, convergence rate, respectively.

When σ is chosen between 0.5 and 1, convergence rate is highest, and the mean of 50 computations is smaller. When σ is bigger than 1, convergence rate is tough, but the whole trend decreases, and the trend of mean fitness increases.

IV. DESIGN OF PARAMETER OPTIMIZATION

A. Algorithm Idea and Process

Parameter optimization of PSO is a complicated nonlinear problem, so we can't use obvious function to express. But the advantages of PSO algorithm can be used to solve complex nonlinear optimization problems. Therefore, we design an algorithm to optimize the parameters by other standard PSO algorithm, called PSO-PARA algorithm. AS distinction, PSO algorithm for optimization problem is called PSO-OPT algorithm.

The idea of PSO algorithm is that parameters of particle swarm algorithm is taken as objects to be optimized in PSO-PARA algorithm, i.e. the particle position, and the particle dimensions is the number of parameter. For iteration, particles' current position information (algorithm parameter) is used to run the PSO to solve optimization problem. At the same time fitness function is used to evaluate the PSO-OPT algorithm performance, and particle in PSO-PARA algorithm updates its speed and position according to individual optimal fitness and global optimal fitness, so as to realize parameter optimization.

Algorithm operation procedure is as follows:

(1) Initialize PSO-PARA algorithm parameters. Set $para\_c_1$, $para\_c_2$, and $para\_w$ of PSO-PARA algorithm within permitted, the initial position $para\_x$ and velocity $para\_v$ of particle is randomly selected.

(2) Particles' current position information $para\_x$ is used to run the PSO-OPT to solve optimization problem, and fitness function is used to evaluate the PSO-OPT algorithm performance to get the current fitness $para\_fitness$.

(3) Compare current position fitness $para\_fitness$ of each particle with individual extremum $para\_pbest$, if it's better, update individual extremum with $para\_pbest$.

(4) Compare current position fitness $para\_pbest$ of each particle with global extremum $para\_gbest$, if it's better, update global extremum with $para\_gbest$.

(5) If it meets the termination conditions, then output global extremum $para\_gbest$ and its position; otherwise, update each particle's velocity and location according to (1), and then turn to step(2).

B. Design of the Fitness Evaluation Function

Parameter optimization is a complicated nonlinear problem, so we can't use obvious function to express. Fitness function is used to evaluate the PSO-OPT algorithm parameter in PSO-PARA algorithm. Therefore, it's important to design fitness function to make it show the relationship between parameters and algorithm performance[2].

Algorithm performance index include algorithms searching capability, stability, convergence rate, computation time and some others. It is not comprehensive to evaluate an algorithm only from one. Thus, we fully consider these several performance indexes, and fitness function is expressed as a function of these indexes, namely as:

$$\begin{gathered} Fit(k) = \alpha_1 f_1(k) + \alpha_2 f_2(k) + \alpha_3 f_3(k) + \alpha_4 f_4(k) \\ \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1 \end{gathered} \tag{2}$$

where α_1, α_2, α_3, α_4 are weight coefficient.

In order to avoid the influence of the four functions $f_1(k)$, $f_2(k)$, $f_3(k)$, $f_4(k)$ output values scope on fitness function, their output values are limited on [0, 1]. In this paper, the greater fitness function is, the better PSO algorithm performance is. Each expression is as follows:

1. $f_1(k)$ presents the ability of PSO-OPT to search for the optimal solution, the greater $f_1(k)$ is, the better the optimal solution is, that is, the stronger the ability of PSO-OPT to search for the optimal solution with this group of parameters is. Expression is as follows:

$$f_1(k) = exp(f_1\_best - f_1\_avg(k)) \tag{3}$$

where f_1_best is the optimum solution or known solution. $f_1\_avg(k)$ is the optimum solution PSO-OPT finds.

2. $f_2(k)$ presents the stability of PSO-OPT algorithm. Here the optimal rate can describe algorithm stability. The higher the optimal rate is, the higher the algorithm stability is. Expression is:

$$f_2(k) = okNum/cMax \tag{4}$$

where $cMax$ is operation time, $okNum$ means algorithm convergence time under set precision.

3. $f_3(k)$ is the algorithm convergence speed. We use the PSO-OPT algorithm mean convergence iteration to describe the convergence speed, expression is as follows:

$$f_3(k) = exp(-sum\_n/okNum) \tag{5}$$

where sum_n is the total iteration.

4. $f_4(k)$ is the time PSO-OPT take to solve the optimization problems. For a given problems, parameters affects the computation time are mainly particle number and the maximum iteration $nMax$. $nMax$ is given usually, therefore, the particle number is only measured the computation time. The expression is as follows:

$$f_4(k) = exp(-particleNum/particleMax) \tag{6}$$

where $particleNum$ is particle number given by the kth particle swarm in PSO-PARA, and $particleMax$ is maximum of particle search range set by PSO-PARA. The less the number of particles is, the bigger $f_4(k)$ is.

V. SIMULATION AND ANALYSIS

A. Simulation for Parameter Optimization Algorithm

PSO-PARA algorithm is introduced inertial factor falling with linear iteration, and parameter set consult [11]. These parameters are taken as: learning factor c_1=c_2=1.4962, w_end=0.9, iteration $Tmax$=100, and particle number m=20. PSO-OPT is a standard PSO algorithm, and parameters need to optimize are w, c_1, c_2, and m. Therefore, particle dimension of PSO-PARA is four, and the variation scope of each particle is shown in Table II.

Table II
RANGE OF PARAMETERS IN PSO-OPT ALGORITHM

parameters	boundary	parameters	boundary
m	[0, 30]	c_1	[0, 4]
w	[0, 2]	c_2	[0, 4]

Consider the optimal rate is the most important algorithm performance index, therefore, coefficient of optimal rate α_1 is set as 0.6, and other coefficients are α_2=0.2, α_3=0.1, α_4=0.1, respectively. PSO-OPT algorithm run 50 times independently, and $nMax$=1000 to test each benchmark function. Table III gives the optimal parameters set of six benchmark function obtained in the algorithm.

Table III
RESULTS OF OPTIMIZED PARAMETERS

Problems	m	w	c_1	c_2
Sphere	17	0.4417	2.2684	1.5619
Rastrigin	16	0.6957	1.7791	0.6785
Griewank10	30	0.6404	2.2626	0.8442
Griewank30	16	0.5408	1.6558	2.2397
Rosenbrock	27	0.6704	1.3603	1.1183
Schaffer	30	0.6765	2.8550	0.5851

B. Performance Comparison Test

We make the six benchmark test functions run independently with parameters give by literature and this algorithm. The iteration is 1,000, and they run 100 to compare the average of algorithm performance. Comparison algorithm is standard particle swarm algorithm, and the parameters are given by [12], i.e., c_1=c_2=1.4962, w=0.729, m=30. The comparison results are shown in Table IV.

The parameters optimization simulation and performance comparison test results show that the algorithm of this article selects the parameters better than the commonly parameters greatly, especially optimal rate, because the common algorithm parameter set pays more attention to generality, Therefore, on some certain issues the convergence is not good. But this algorithm has a strong pertinence, so the performance increases.

Table IV
RESULT OF PERFORMANCE COMPARISON TEST

Problems	parameters	best function value	worst	average	variance	average iterations	probability
Sphere	[12]	0	0	0	0	333	1
Sphere	Table III	0	0	0	0	303	1
Rastrigin	[12]	14.9244	111.4349	61.2694	17.4093	165	0.97
Rastrigin	Table III	32.2841	97.5172	54.4789	13.53	86	1
Griewank10	[12]	0.0123	0.3049	0.0861	0.0428	868	0.19
Griewank10	Table III	0	0.1724	0.0509	0.0313	545	0.62
Griewank30	[12]	0	0.5971	0.0761	0.0979	716	0.54
Griewank30	Table III	0	0.1257	0.0209	0.0245	643	0.91
Rosenbrock	[12]	0.4092	224.9861	49.1006	44.1179	437	0.90
Rosenbrock	Table III	0.2627	224.7255	42.4115	39.1599	326	0.94
Schaffer	[12]	0	0.0097	0.0042	0.0048	567	0.57
Schaffer	Table III	0	0.0097	0.0004	0.0019	385	0.94

VI. CONCLUSIONS

In using particle swarm algorithm is discrete optimization process, the algorithm of optimal parameters is not the only. We simplify the PSO algorithm, and analyze the relationship among parameters under stable system, then considers PSO parameter selection as a complex optimization problem, set appropriate fitness function to describe the algorithm performance. Test results about benchmark function show that this design parameters optimization algorithm can facilitate effective realization of PSO algorithm the parameter selection, and the parameters selected by this method are better than the experience parameters in the optimal fitness, the mean of optimal fitness, convergence rate, and etc, especially optimal rate.

ACKNOWLEDGMENTS

This paper is supported by National Natural Science Foundation of China under the Grant 51009036/E091002, and the Special Fund for Basic Scientific Research of Central Colleges under the Grant HEUCF100423.

REFERENCES

[1] L. Wang, *Swarm Intelligence and its application*, Tsinghua University publisher, BeiJing; 2001.

[2] Y.T. Dai, L.Q. Liu and S.J. Wang, "Parameter optimization of ant colony algorithm based on particle swarm optimization", *Proceedings of the International Conference Information Computing and Automation*, Chengdu, 2008, pp. 1266-1269.

[3] E. Ozcan and C.K. Mohan, Analysis of a simple particle swarm optimiaion system, *Intelligent Engineering Systems through Aritificial Neural Networks*, 1998, pp. 253-258.

[4] E. Ozcan and C.K. Mohan, "Particle swarm optimization: Surfing the waves", *Proceedings of the Congress on Evolutionary Computation*, Piscataway, 1999, pp. 1939-1944.

[5] M. Clerc and J. Kennedy, The Particle Swarm: ExPlosion, Stability, and Convergence in a ComPlex Space, *IEEE Transaction On Evolutionary ComPutation*, vol. 6(1), 2002, pp 58-73.

[6] J.W. Wang and D.W. Wang, Experiments and analysis on inertia weight in particle swarm optimization, *Journal of Systems Engineering*, vol. 20(2), 2005, pp. 194-198

[7] S. Gao, K.Z. Tang, X.Z. Jiang and J.Y. Yang, Convergence Analysis of Particle Swarm Optimization Algorithm, *Science Technology and Engineering*, vol. 6(12), 2006, pp. 1625-1631.

[8] G.Y. Pan, D.Y. Dou and X.H. Liu, Convergence property and parameters2selection strategy base on differential evolution of PSO, *Journal of Jilin University (Engineering and Technology Edition)*, vol. 37(4), 2007, pp. 842-845

[9] J. KennedyR.C. Eberhart, "Particle swarm optimization proc", *IEEE Service Center ed.IEEE International Conference on Neural Networks*, Perth,Australia, 1995, pp. 1942-1948

[10] R.C. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory", *Proc of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995, pp. 39-43

[11] R.C. Eberhart and Y. Shi, "ComParing inertia weigthts and constriction factors in particle swarm optimization", *Proeeedings of the IEEE Congress on Evolutionary ComPutation*, SanDiego, 2000, pp. 84-88

[12] M. Clerc, "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization", *Proceedings of the Congress of Evolutionary Computation*, Washington, 1999, pp. 1951-1957