Particle Swarm Optimization Algorithm Based on the Idea of Simulated Annealing

DONG Chaojun, QIU Zulian

(Automation Control Research Institute, School of Electronic & Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Summary

Particle swarm optimization (PSO) algorithm is a new population intelligence algorithm and has good performance on optimization. After the standard PSO algorithm and the idea of simulated annealing algorithm had been analyzed, the acceptance of Metropolis rule by probability in the simulated annealing algorithm was introduced in the algorithm of PSO. The simulated annealing-particle swarm optimization was presented. Simulation result shows that the disadvantage of getting in the local best point of standard PSO was overcome effectively and the ability of global optimality are toned up.

Keywords particle swarm optimization; Simulated annealing; global optimality; Metropolis rule

1. Introduction

Particle swarm optimization (PSO), a kind of evolvement-computation technology based on swarm intelligence, was raised by Kennedy and Eberhart who were aroused by the research results about artificial life in 1995. The basic idea of PSO came from the research on the behavior of bird swarms looking for food. Every particle always follows the two best positions—the best position in the whole swarm and itself in iteration computation, so it converges very fast, but there are several shortcomings in PSO below: (1) getting in a local best point easily; (2) It is difficult to deal with the constraints of the optimization problem. From then the inertia coefficient was discussed [2-6]

2. Sa-PSO Algorithms

2.1 Basic PSO Algorithm [2-3, 5-6]

There are m particles in a swarm that is in a space of D dimensions, the ith particle's position in the space is below: $\mathbf{X}_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$, $i = 1,2,\cdots,m$, which is a latent solution. The ith particle's flit speed is below: $\mathbf{v}_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$, $i = 1,2,\cdots,m$, and until now the best position of the ith particle is below: $\mathbf{P}_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$, $i = 1,2,\cdots,m$, the best position in the whole swarm until now is below: $\mathbf{P}_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$, the PSO algorithm is below

$$x_{id} = x_{id} + v_{id} \tag{1}$$

$$v_{id} = v_{id} + c_1\gamma_1(p_{id} - x_{id}) + c_2\gamma_2(p_{gd} - x_{id}) \tag{2}$$

Where, $i = 1,2,\cdots,m$, $d = 1,2,\cdots,D$, c_1 and c_2 are, respectively, the study coefficients of cognizing and society, and are both positive constants. The relative value of c_1 and c_2 expresses the relative importance-degree of $\mathbf{P}_i$ and $\mathbf{P}_g$ with evolvement. γ_1 and γ_2 are both random numbers between 0 and 1, $v_{id} \in [-v_{\max}, v_{\max}]$, $v_{\max}$ is decided by the user.

Equations (1)-(2) can be changed as

$$x_{id} = \alpha x_{id} + v_{id} \tag{3}$$

$$v_{id} = w v_{id} + c_1\gamma_1(p_{id} - x_{id}) + c_2\gamma_2(p_{gd} - x_{id}) \tag{4}$$

where α and w are, respectively , constraint factor and inertial factor (w >0).

Many scholars regard the equations (3)-(4) as the basic PSO algorithm, It is obvious that the PSO algorithm is very simple, but the algorithm itself includes the principles of sociology, psychology and bionomics, so the particle swarm including some simple particles behaves a complex action, which is the main reason that many peoples are sure that the PSO algorithm has a good foreground.

2.2 Sa-PSO Algorithm

The idea of simulated annealing algorithm is presented by Metropolis in 1953, and was used in compounding optimization by Kirkpatrick in 1983. It accepts the current optimal solution at a probability after searching, which called Metropolis law. And Sa-PSO algorithm become a global optimal algorithm by using this new acceptance rule, the theory has been proved.

The basic idea of simulated-annealing particle swarm optimize algorithm (Sa-PSO) is shown below:

at the beginning, the individual best point and the global best point were accepted by the Metropolis rule, the hypo-best point was accepted at probability, the aim function is allowed to become worse at a certain extent, the acceptance rule was decided by the coefficient *T*, *T* is the anneal temperature. With the *T* descending, the searching region would be around the best point, the accepted probability of the hypo-best point become small also, when the T descend to the lower limit, the accepted probability of the hypo-best point is zero, the algorithm only accept the best solution as the basic PSO algorithm. The relation between the annealing temperature and the inertial weight was built, the inertial weight change with the temperature, and then the searching precision was changed following the inertial weight, so the searching speed was increased.

2.2.1 Metropolis accept probability law

The accepted probability function was got from the extent Boltzman-Gibbs distributing, shown blow

$$P = \left[1 - (1-h)\Delta f / T\right]^{1/(1-h)} \quad (5)$$

Where $\Delta f = f(X_i) - P_i$; $f(X_i)$ is the *i*th particle solution, P_i is the historical best solution, T is the anneal temperature; h is real number that regulate the relation between margin and temperature, when $h \to 1$, then

$$p = \exp(-\Delta f / T) \quad (6)$$

This is the general SA algorithm accept probability function, it is the special one of function (5), and the Sa-PSO algorithm also use it.

2.2.2 Cooling schedule

This shows how the temperature parameter is decremented at each step. The fast simulated annealing (VFSA) schedule used widely was chosen , which shown blow

$$T(k) = T_0 \exp\left(-ck^{1/N}\right) \quad (7)$$

Where T_0 is the initialized temperature, k is annealing-time; c is constant defined; N is number of syllogism coefficient. The function (7) also could be shown as below

$$T(k) = T_0 \alpha^{k^{1/N}} \quad (8)$$

Where α is rate of cooling, $0.7 \le \alpha \le 1.0$. In real use the $1/N$ in the function (8) is replaced by 1.5 or 1.0.

2.2.3 Inertial Factor Function

The inertial factor in the function (3) is used to control the effect between the previous step speed and the current particle speed, the bigger inertial factor w can enhance the global searching ability, and the smaller inertial factor w could enhance the local searching ability. In the simulated annealing algorithm, the temperature T has the similitude ability, it could enhance the global searching ability at higher temperature and enhance local searching ability at lower temperature. A new mapping function was presented to describe relation between the two factors. It is used to enhance the global the searching ability at the beginning and improve the searching speed at end of algorithm. The function shown below

$$w = w_0\left(1 - \frac{T_0 - T(k)}{\beta * T_0}\right) \quad (9)$$

Where β is a adjust factor, it make sure that the inertial factor at a logical region. w_0 is the basic PSO initialization inertial factor.

2.3 The astringency of Sa-PSO algorithm

Assume the $P_g and P_i$ keep unchanged in the evolution, when

$\sqrt{2(1+w-\phi)^2-4w}<2$

$(\phi=\phi_1+\phi_2,\phi_1=c_1r_1,\phi_2=c_2r_2)$, the particles of PSO algorithm converge to the weight center of P_g, P_i,

$X_i \to \frac{\phi_1 p_i + \phi_2 p_g}{\phi}$. But in real use, the P_g, P_i keep on changing in the evolution, it could also be proved that if the algorithm fits the condition above, the astringency of PSO algorithm can be sure.

The Sa-PSO algorithm's configuration is similitude to the PSO, the different between two algorithms is at the beginning of algorithm, with the temperature cooling, the Sa-PSO degenerate to PSO, so the algorithm content the condition above, the astringency of Sa-PSO would be sure.

2.4 Algorithm

The function of (3) and (4) is the basic PSO algorithm, the steps of Sa-PSO shown below:

Step1: To initialize the coefficient, it includes the annealing temperature T and w、c_1、c_2. To initialize the particle swarm, it includes the particle random position and the first speed;

Step2: To evaluate every particles adaptive value $f(X_i)$;

Step3: For each particle, the adaptive value $f(X_i)$ is compared with one of the historical best position P_i, if the adaptive value is better than one of P_i. Then, X_i is consider as the best position P_i, otherwise, using the accept-probability law function (6) to decide if this point be accepted.

Setp4: For each particle, the best point P_i itself was compared with the whole best point P_g, if P_i is better than P_g, then reset P_g, otherwise, the global best point is accepted according to the probability function $p=\exp(|P_i-P_g|/T)$

Step5: The position and speed of each particle were changed following the function (3) and (4), several times later, to adjust the temperature T following the function (8) and the inertial weight w following the function (9);

Step6: if haven't get the stop condition (the condition is the best adaptive value or setting a maximum iteration times in generally), then back to step2, else the algorithm stops.

3. Simulation experiment

The general Griewank function and Schaffer's f6 function were used to be test function. They are the classical optimize test function which often be used to compare the ability with different algorithm.

To compare the algorithm ability, three algorithms are used here: the first one is the basic particle swarm optimization (std-PSO) algorithm; the second is particle swarm optimization with the inertial weight reducing linearly (Iw-PSO); the third is Sa-PSO presented in this paper.

The general Griewank function shown below:

$$\min f(X)=\frac{1}{4000}\sum_{i=1}^{30}x_i^2+\prod_{i=1}^{30}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1 \qquad (10)$$

$x_i \in [-600,600]$

Fig.1 shows the Griewank function in 3 dimension coordinate. The function has the global best point $X^*=(0,\cdots,0)$, in which the function value is 0.

The Griewank function is similar to the Sphere function with adding the noise part $\prod_{i=1}^{30}\cos\left(\frac{x_i}{\sqrt{i}}\right)$, with dimension increasing, this noise part tends to 0, so the function has more local best points when the dimension is lower, and it's hard to convergence to the global best point.

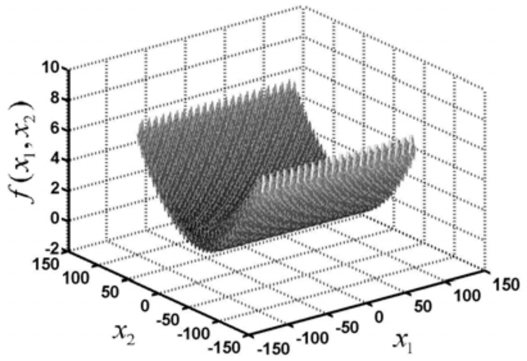

Fig.1 Griewank function

The experiment parameters are set below:

The number of particles is 30, run random 20 times, The results are shown in Table1 below.

Table 1 *Simulation result.*

dimension	Times	Griewank mean value $\times 10^{-3}$		
		Std-PSO	Iw-PSO	Sa-PSO
2	200	1.262	0.887	0.258
10	2000	12.764	8.917	6.594
20	2000	37.431	15.664	13.572
30	2000	55.322	35.007	21.361

Simulation results show: both in high dimension and in low dimension, the Sa-PSO get the better solution than the other algorithm, and has a powerful optimization ability.

To use Schaffer's f6 function for testing the validity of C-PSO algorithm, shown below

$$f(x, y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1 + 0.001(x^2 + y^2))^2} \qquad (11)$$

$$s.t. -100 < x, y < 100$$

The function has the best point $X^* = (0,0)$, in which the function value $f(X^*)$ is 1. There are infinite local best points around the global best point, all values of them are 0.990284, it is shown in the Fig.2.

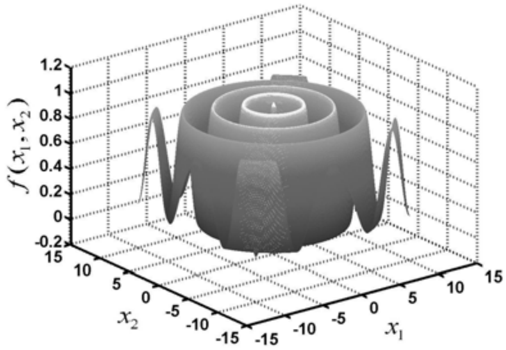

Fig. 2 Schaffer's f6 function

The experiment parameters are set below:

The number of particles is 30, the error of iteration is 0.00001, run random 200 times, and the statistical results are shown in Table2 below.

Table 2 *Simulation result.*

Algorithm	Average time	Max times	Min times
Std-PSO	803.32	2904	100
Iw-PSO	781.25	2830	140
Sa-PSO	650.91	1692	65

Simulation results show: to get the global best solution the Sa-PSO needs less times than the other two algorithms, in other words, it constringent speed is fast. Fig3 is one example of Sa-PSO algorithm searching. It is obvious that the searching curve of Sa-PSO is not always smooth, there are several shakes in it that were caused by the acceptance probability law, but in

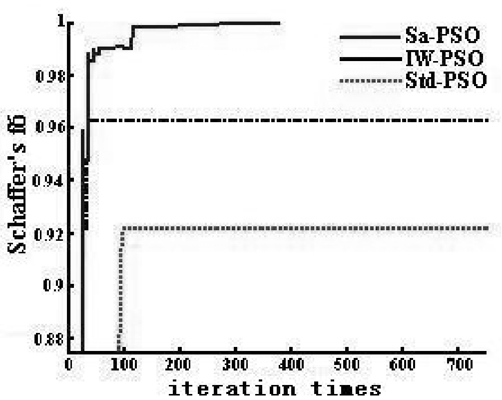

Fig. 3 An example of simulation

the anaphase of algorithm, the curve became smooth ,and the effect of acceptance probability law became lower.

The more researches show that the PSO algorithm is easy for getting in local best points easily when the number of particles is little, and it needs much inertial times to get out of local best point region. Because of the Metropolis law, the Sa-PSO algorithm still do well at little particle swarm. To confirm this, another simulation been done. The number of particle only 20, the maximum generation is 3000 and 10000, it failed if the loop generations is more than the maximum generation. Run random 200 times. The statistical results are shown in Table3 below. Simulation results show: the Sa-PSO fails at least times.

Table 3 *Simulation result.*

Algorithm	Failed searching times	
	Gmax=3000	Gmax=10000
Std-PSO	5	1
Iw-PSO	2	0
Sa-PSO	0	0

4. Conclusion

(1) It is meaningful to use Metropolis rule to improve the basic particle swarm optimization algorithm, the hypo-best point was accepted at a certain extent probability in the algorithm improved, it's useful to get out of the local best points and accelerate the constringency speed of the algorithm.

(2) The mapping relation between in inertial weight and annealing temperature is built based on their character, the searching precision and speed was improved.

(3) The result of simulation indicates that the disadvantage of PSO is conquered; the ability of global optimality is toned up.

(4) The research of PSO is just at the beginning, there are still much problems for studying thoroughly

Acknowledgments

Supported by Education office of Guangdong Province Natural Science Emphasis Research Item (05Z025) and Guangdong Provincial Natural Science Foundation (06029813) in China.

References

[1] Eberhart, R. C, Kennedy, J. A new optimizer using particle swarm theory [A]. In: Proceedings of the Sixth International Symposium on Micromachine and Human Science [C]. Nagoya, Japan, 1995, 39～43

[2] Shi Y, Eberhart R C. A modified particle swarm optimizer [R]. IEEE International Conference of Evolutionary Computation, Anchorage, Alaska, 1998

[3] Clerc M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization [A]. In: Proceedings of the Congress on Evolutionary Computation [C]. Piscataway, NJ: IEEE Service Center, 1999,1951～1957

[4] Lovbjerg M, Rasmussen T K, Krink T. Hybrid particle swarm optimiser with breeding and subpopulations [A]. In: Proceedings of the Genetic and Evolutionary Computation Conference [C]. San Francisco, USA , 2001,469-476

[5] Kennedy J. Stereotyping: Improving particle swarm performance with cluster analysis [A]. In: Proceedings of the Congress on Evolutionary Computation [C]. San diego,CA, 2000, 1507～1512

[6] Venter, G, Sobieszczanski-Sobieski, J. Multidisciplinary optimization of a transport aircraft wing using particle swarm[j]. Structural and Multidisciplinary Optimization, 2004, 26(1-2):

121～131.

[7] Lester Ingber. Very fast simulated re-annealing [J]. Mathematical Computer Modeling, 1989, 12(8): 967～973.

[8] Parsopoulos K E, Plagianakos V P, Magoulas G D, et al. Improving particle swarm optimizer by function ”stretching” [A]. In: Hadjisavvas N, Pardalos P. Advances in Convex Analysis and Global Optimization [M]. The Netherlands: Kluwer Academic Publishers, 2001. 445～457

[9] Suganthan P. N. Particle swarm optimiser with neigbourhood operator[A]. In: Proc of the Congress on Evolutionary Computation[C]. Piscataway , NJ : IEEE Service Center , 19991 1958～1962

[10] Ozcan E, Mohan C. K. Particle swarm optimization: Surfing the waves [A]. In: Proc of the Congress on Evolutionary Computation[C]. Piscataway , NJ : IEEE Service Center , 1999,1: 1939～1944

[11] Clerc M, Kennedy J. The particle swarm: Explosion, stability and convergence in a multi-dimensional complex space [J]. IEEE Transactions on Evolutionary Computation , 2002 , 6 (1) : 58～73

Authors Biography

DONG Chaojun is a Ph.D. student at Xi’an Jiaotong University in China, and also an associate professor in control science at Wuyi University in China. He was born in 1967 in Jiangling county, Hubei province, China. He got his Master’s Degree in 1993 from Central South University. His research interests are intelligent control theory and its application, intelligent traffic control.

QIU Zulian is a professor in control science at Xi’an Jiaotong University in China, He was born in 1937 in Jiangsu province of China. His research interests are intelligent control, intelligent traffic control.