

An Improved PSO Algorithm Based on Chaos and Population Core

Zhongyong Wu

Teaching Technology center of Fujian Medical University,
Fuzhou, China
e-mail: 394964465@qq.com

Lili Gan

Department of information science and Technolo-gy,
University of Xingtai
Xingtai, China
e-mail: baicao003@163.com

Abstract—particle swarm optimization (PSO) algorithm is often trapped in local optima and low accuracy in convergence. Following an analysis of the cause of the premature convergence, a novel particle swarm optimization algorithm based on neighborhood explored and chaos is proposed, which is called PCC-PSO. Chaos is introduced to initialized the particle's position to improve the diversity, the population core learning mechanism and global extreme mutation operator is also introduced to enhance the global search ability. Compared with other three improved algorithms, the PCC-PSO converges faster, prevents the premature convergence problem more effectively.

Keywords: particle swarm optimization; population core; self-adaptive; position of mutation

I INTRODUCTION

Particle Swarm Optimization (PSO) algorithm proposed by Eberhart and Kennedy in 1995 [1], which is an emerging global based on swarm intelligence optimization evolutionary algorithm, which originated from the movement of birds foraging behavior analog. Compared with other evolutionary algorithms, the main advantages are: the concept is simple, the parameters can be real-coded, and does not require gradient information, easy to implement [2,3]. PSO algorithm has been proposed, immediately aroused widespread concern in the field of evolutionary computation, and the emergence of a large number of research results. As a potential parallel optimization algorithm can be used to solve the non-linear, non-differentiable and multi-peak complex optimization problems [4], and has been widely used in science and engineering fields, such as: Eq. optimization, fuzzy system control, pattern recognition , neural networks [5-8].

The velocity V_{id} and position X_{id} of d th dimension of the i th particle are updated as follows:

$$V_{id}(k+1) = \omega * V_{id}(k) + c1 * r1 * (P_{id} - X_{id}(k)) + c2 * r2 * (P_{gd} - X_{id}(k)) \quad (1)$$

$$X_{id}(k+1) = X_{id}(k) + V_{id}(k+1) \quad (2)$$

Where: $i=1,2,\dots,N$; $d=1,2,\dots,D$; ω is inertia weight; X_{id} is the position of the i th particle; V_{id} represents velocity of the i th particle; $pbest_i$ is the best previous position yielding the best fitness value for the i th particle; and $gbest$ is the best position discovered by the whole population; $c1$ and $c2$ are

the acceleration constants reflecting the weighting of stochastic acceleration terms that pull each particle toward $pbest_i$ and $gbest$ positions, respectively. $r1$ and $r2$ are two random numbers in the range $[0, 1]$.

II PPC-PSO

A. Initialize The Particle Swarm Initial Position

Chaos is no fixed cycle cyclic behavior, gradual self-similar order of aperiodic. The chaos has its unique properties: the randomness; ergodicity; regularity. Given the chaos of these three characteristics, taking into account the movement characteristics of the chaotic sequence, randomness and ergodicity and regularity, it introduced this feature to initialize the position, so better to reflect the diversity of the initial population find the optimal solution for the optimization space and speed up the convergence to lay a more solid foundation.

Logistic map is a typical chaotic system, first use of the Logistic mapping chaotic sequence:

$$L(d+1) = \mu * L(d) * (1 - L(d)) \quad (3)$$

Eq. (3) is a constant, $\mu \in [3.56, 4.0]$ referred to as a control parameter, and mainly used to control the degree of chaos system, $L(d) \in (0,1), d=1,2,3,\dots,D$.

The second step in D -dimensional intermediate N particles, first generates N initial value: $L_1(1), L_2(1), \dots, L_i(1), L_N(1)$, then Eq. (4) is:

$$X_{i,d} = \frac{L_i(d) * (Max_d - Min_d) * i}{N} + Min_d \quad (4)$$

B. Population Core Learn Mechanism

In paper [2]: "population core refers to the best position of each particle mean and the mean of the current optimal position, this approach has two shortcomings: only reflects the evolution trend of the " history "of the population, and does not reflect the lost using the mean value calculated by the assessment of the significance of the fitness of the individual particles, and consider only the distribution of the populations position posture, so that in a certain extent equivalent to the contribution of all individuals of the population evolutionary same, i.e. "current"; to some extent with the evolution of egalitarian thought populations, reducing the "elite individual plays an important role in promoting the contribution of the evolution of the

population [25]. Therefore, to reflect population core of the current evolutionary trend given calculated as follows:

$$Pecent^i(k) = \frac{Fit(X_i(k))}{\sum_{i=1}^N Fit(X_i(k))} \quad (5)$$

$$PopC(k) = \sum_{i=1}^N Pecent^i(k) * X_i(k) \quad (6)$$

$$V_{id}(k+1) = \omega * V_{id}(k) + c1 * r1 * (P_{id} - X_{id}(k)) + c2 * r2 * (\beta_{id} * PopC(k) - X_{id}(k)) \quad (7)$$

Eq. (5) in $Pecent_i(k)$ denotes the k th iteration, the proportion corresponding to the adaptation of the i -th particle population; Eq. (6), in $PopC(k)$, which means that the population in the k th iteration, according to the Eq. (6) calculated for guiding the first evolution of the $(k+1)$ th iteration. Visible, the greater the degree of adaptation $Fit(X_i(k))$ is higher $Pecent^i(k)$ to the contribution of particles $X_i(k)$ the $PopC(k)$, the greater, the population core consider the fitness of each particle, but also take into account the population distribution of the particles, a combination of the current situation of the population. β_{id} for the center of mass coefficient, the value of $[1/N, 1)$ random number between said of population core information on the single particle output. Early in the evolution of the particle distribution is relatively uniform the optimal particle population in a local optimum or near local optimum likely $gbest$ to guide the evolution of populations search, is likely to guide the population and rapid access to local optimal region; in the later stage of evolution, if there is a $gbest$ local optimum operating speed of each particle decreases rapidly (and may even be reduced to 0, to stop evolution), it is more difficult to escape from local optima. Therefore, in the early stage of evolution, the individual particles in addition to consider its own historical best, but also consider the trend of the evolution of the population, try to use the population core $PopC(k)$ to guide the evolution; be further clarified in the later stage of evolution, population particle convergence, population core is calculated in $PopC(k)$ with the groups optimal $gbest$ small difference, both to guide the evolution of populations difference is not too large, may be considered directly $gbest$ to guide the evolution, in order to reduce the computational amount.

For the realization of the evolutionary approach to learning, we design a linear decreasing Eq. (8):

$$R_k = 1 - \frac{k}{numofGm} \quad (8)$$

For each particle in the evolution process of k th generation, a random number P in $(0, 1)$ is produced, calculated R_k with Eq.(8), if $P \geq R_k$, then update velocity by Eq.(1), else calculated Eq.(5) and Eq.(6), then take Eq.(7) to update the velocity.

C. Position Mutation Mechanism

The evolution of the particle swarm when the optimal solution is no change in continuous Max iterations can be that the algorithm may fall into the local extremes, indicating that the particle swarm according to the existing trajectory or about to slide into a local optimum. Max value based on the scale of the problem solving pre-specified or dynamically determined according to the problem using some adaptive way. The larger the value of Max , show that the judgment premature convergence criteria more relaxed. A counter recorded so far stalled algebra MSG , can be set: the same twice in a row to get the optimal solution of the MSG is incremented by one, otherwise it is cleared; when $MSG \geq Max$, then the algorithm may stagnate Max iterations, the particle does not have the ability to escape from local optima. At this point, to change the trajectory of the particles, the current best solution $P_{gd} = (pg_1, pg_2, \dots, pg_d, \dots, pg_D)$ for an "alternative" disturbance, namely:

$$if (MSG \geq Max) \text{ then } p_{gd} = rand(\) * p_{gd} \quad (9)$$

III EXPERIMENT AND ANALYSIS

A. Experimental Design

The algorithm: PSO, sPSO[8], tPSO [8], HPSO-TVAC[11], compared on four typical benchmark functions to verify the feasibility of PPC-PSO. The function form, scope and dimension, theoretical limit (minimum) and target precision shown in paper [10], sPSO and tPSO, HPSO-TVAC parameter setting as paper [8] and [11] unanimously, PPC-PSO parameter : $\omega_0 = 1.2$, $c1 = c2 = 2$, $\omega = \omega_0 * \exp(-0.5 * k * k)$, $\mu = 3.99$, $Max = 10$.

B. Analysis Of Experimental Results

1) Compared the number of iterations under the fixed convergence accuracy

Defined as follows: population size $N = 15$, $numofGm = 3000$, under the giving accuracy in TABLE 1 provides the result by 50 independently times run, where: Min, Mean, Max, Var, SR, EI, "∞" is the same as [5].

TABLE I. THE NUMBER OF ITERATIONS FOR THE GOAL AFTER 50 RUNS

Experiment code	Iteration				SR	EI	
	Min	Mean	Max	Var			
f_1	PSO	3000	3000	3000	0	0	∞
	PPC-PSO	9	12	16	2.74	1	623
	sPSO	16	26	42	21.94	1	1301
	tPSO	51	80	105	129.04	1	3984
f_2	PSO	3000	3000	3000	0	0	∞
	PPC-PSO	7	10	14	1.8	1	490
	sPSO	23	28	33	5.94	1	1412
	tPSO	60	87	108	157.92	1	4328
f_3	PSO	3000	3000	3000	0	0	∞
	PPC-PSO	9	63	132	2432.23	1	3150
	sPSO	1832	2543	3888	399793	0.18	706419
	tPSO	2827	2827	2827	0	0.02	7067500
f_4	PSO	3000	3000	3000	0	0	∞
	PPC-PSO	8	1003	1934	225402.79	1	50161
	sPSO	23	796	3688	880321.96	0.8	49731
	tPSO	123	315	618	12064.10	1	15734

From Table 1: PSO algorithm for optimization were not converge to the target accuracy; the HPSO-TVAC performance is better than the PSO algorithm, but it is also the only f_3 only a 10% success rate of convergence, and the convergence to the target accuracy number of iterations, and the other three functions have failed to converge to the target accuracy; PPC-PSO in the optimization of all the functions of a success rate of 100%, while the success rate of sPSO f_3 and f_4 were 18% and 80%, 2% and 100% success rate tPSO f_3 and f_4 , respectively.

In the optimization f_4 , consolidated can be seen: than PPC-PSO, the tPSO in Mean, Max, Var (measure of the stability of the convergence of the algorithm), EI aspect smaller, which is because: tPSO algorithm where extreme disturbance operator design is more suitable for this type of function optimization: using $rand()*pbest_i$ and $rand()*gbest$ the gbest disturbance, so that in the case of increasing the number of iterations, the individual extreme $\|pbest_i\|$ and global extreme $\|gbest\|$ are faster and closer to the theoretical extreme points $(0, 0, \dots, 0, \dots, 0)$, is conducive to that type of Eq. optimization; except tPSO, PPC-PSO is better than Min, Mean, Max, Var, EI and SR algorithms.

In summary can be obtained: learning algorithm by adding the population core operator and position variation operator, can help to improve the convergence speed and accuracy, efficiency and can avoid premature convergence.

IV CONCLUSION

Base the analysis of the basic PSO algorithm premature convergence on the PPC-PSO algorithm, first introduced chaos initialization, the particle distribution as uniform as possible in the search space to lay a solid foundation for the full search algorithm optimization space; followed by the introduction of a the population core learning mechanism to strengthen the detection and search of the search space, and the last to join the global extremum mutation operator. Improved PSO algorithm with four typical benchmark

functions with the other three more representative test, the experimental results show that the effectiveness of the PPC-PSO.

REFERENCES

- [1] Kennedy J, Eberhart R C. Particle swarm optimization [C]. In: Proc. of the IEEE International Conference on Neural Networks, Perth, Australia. 1995,1942-1948.
- [2] Shi Y, Eberhart R C. A modified particle swarm optimizer [C]. In: Proc. of the IEEE International Conference on Evolutionary Computation. 1998,69-73.
- [3] Clerc M, Kennedy J. The particle swarm: Explosion stability and convergence in a multi-dimensional complex space [J]. IEEE Trans. on Evolutionary Computation. 2002,6(1):58-73.
- [4] Ji Z, Zhou J R, Liao H L et al. A new intelligent single particle optimizer [J]. Chinese Journal of Computers. 2010,33(2):556-561.
- [5] Hu W, Li Z S. A simpler and more effective particle swarm optimization algorithm [J]. Journal of Software. 2007,18(4): 861-868.
- [6] He R, Wang Y J, Wang Q et al. An improved particle swarm optimization of self-adaptive escape velocity [J]. Journal of Software. 2005,16(12):2036-2044.
- [7] Peram T, Veeramachaneni K. Fitness-Distance-Ratio based particle swarm optimization [C]. In: Proc. of the IEEE Swarm Intelligence Symp. Indianapolis. 2003,174-181.
- [8] WU Z Y, GOU J, CUI C C. An Improved Particle Swarm Optimization Algorithm Applied to the Unified Evaluation of Circularity Error [J]. Applied Mechanics and Materials, 2011, 44(47):3937-3941 .
- [9] Wu Z Y, Gou J, Zhao Z Q. An improved particle swarm optimization based on self-adaptive neighborhood explored [J]. Journal of Chinese Computer Systems, 2010, 31(9):1938-1945.
- [10] Ratanaweera A, Halgamuge S K, Watson H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients [J]. IEEE Trans. on Evolutionary Computation, 2004, 8(3):240-255.
- [11] GOU J, WU Z Y, CHEN Hong-Guang. A simple particle swarm optimization algorithm based on self-adaptive neighborhood explored [J]. Information-An International Interdisciplinary Journal, 2011, 14(3): 865-870.