Research on quorum sensing particle swarm optimization based on chaos

Benyan Yu¹, Hongwei Kang¹*, Yong Shen¹, Xingping Sun¹ and Qingyi Chen¹

¹School of software, Yunnan University, Yunnan, China

*Corresponding author e-mail: hwkang@ynu.edu.cn

Abstract: particle swarm optimization (PSO) is a kind of swarm optimization algorithm with fast search speed, high efficiency and suitable for practical optimization problems. Once it was put forward, it has attracted extensive attention of scholars in various fields. A quorum sensing particle swarm optimization (CPSOQS) algorithm based on chaos is proposed to solve the shortcomings of traditional PSO algorithms, such as poor handling of discrete optimization problems and the loss of searching ability due to the fast particle velocity decline in the later stage. Firstly, the search ability of the algorithm is improved by generating chaotic sequences and mapping them to the solution space of the definition domain; Secondly, the quorum sensing mechanism of bacteria is introduced into PSO. Chaos search was used twice in the algorithm, which improved the global search ability. CEC 2005 benchmark is used to test the performance of the algorithm. The experimental results show that among the 14 selected test functions, 11 test functions have better experimental results than the comparison algorithm.

1. Introduction

In 1995, Kennedy and Eberhart [1] proposed a random search particle swarm optimization (PSO) algorithm. The algorithm mainly simulates the foraging behaviour of the bird groups, which shows that they will gather in the process of finding food. In PSO algorithm, each solution set is called a "particle", and each particle flies to a better position in the solution of the problem according to its own experience and the best experience of the group to search for the best solution.

Over the past two decades, many researchers have focused on PSO algorithm and improved its performance in different ways. Shi [2, 3] et al. proposed an adaptive inertia weight method to reduce linearly from 0.9 to 0.4 in the process of operation and improve the performance of PSO algorithm. Marco [4] et al. proposed the fuzzy self-correcting particle swarm optimization (FST-PSO) algorithm, which uses fuzzy logic to independently calculate the inertia, cognitive and social factors, minimum
and maximum velocities of each particle. Nasir [5] et al. proposed a dynamic neighbor learning particle swarm algorithm (DNLPSO), which selects a sample particle from the optimal location of the neighbor including itself, and updates the particle velocity by using the historical optimal information of all particles. Cheng [6] et al. introduced the social learning mechanism into the PSO algorithm, developed a social learning particle swarm optimization (SL-PSO), and further modified the implicit selection mechanism of the particle swarm optimization algorithm. Hu [7] proposed a hybrid particle swarm optimization algorithm (PSO-BFO) to improve the global convergence reliability and speed of the algorithm. Aydilek [8] made full use of the advantages of firefly and particle swarm optimization algorithm, and proposed a hybrid algorithm combining firefly algorithm and particle swarm optimization algorithm.

In 1963, Lorenz [9] pointed out in his paper "deterministic aperiodic flow" that if a definite system is very sensitive to the initial conditions in the evolution process, the result of evolution is dependent on the initial state, then it is determined that small differences will be amplified. Lorenz said that the tiny air flow caused by a butterfly's occasional flapping wings may turn into a tornado, which is known as the "Butterfly Effect" of weather. Chaotic motion [10] can achieve all states without repetition only according to its own law. Therefore, in the process of random search, chaos variables are ergodicity, randomicity and regularity, and the core of chaos optimization lies in ergodicity random search of chaos.

At present, some researchers have integrated various biological behavior mechanisms in nature into PSO algorithm to solve the shortcomings of some algorithms. Inspired by this idea, this paper embeds the relevant biological behavior mechanism into PSO to improve the performance of the algorithm, and proposed the induced particle swarm optimization algorithm based on the chaotic microbial behavior mechanism. The experimental results show that it has great improvement in dealing with local optimization, global search capability and algorithm convergence.

2. Algorithm improvement

2.1. Basic particle swarm optimization

In PSO algorithm, the particle finds the optimal solution by iterating a certain formula for a certain number of times. The updated formula for the particle’s velocity and position is shown in formula (1-2). The particle updates its velocity and position through individual extremum and global extremum, realizing the information exchange between particles. In 1998, Shi and Eberhart [11] formally presented the mathematical description of PSO algorithm as follows:

$$v_{i,d}(k+1) = \omega v_{i,d}(k) + c_1 \varepsilon_1 (p_{i,d}(k) - x_{i,d}(k)) + c_2 \varepsilon_2 (p_{g,d}(k) - x_{i,d}(k)) \quad (1)$$

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad (2)$$

Where $\omega$ is inertia weight coefficient, set $\omega=0.6$; $c_1$ and $c_2$ are acceleration control coefficients, set $c_1=1.7$; $\varepsilon_1$ and $\varepsilon_2$ are uniform random numbers in the range of [0,1].

2.2. Chaos and motion characteristics

Chaos is a new direction of nonlinear dynamic system. Although chaos is a system described by uncertainty theory, it does not mean a chaos, but contains accurate internal structure. There are many types of chaotic systems. Logistic mapping is selected in this paper. Simple as it is, it is a very widely used one-dimensional chaotic map, and the most typical chaotic system. Its mapping relationship
formula 3 is shown in [12]:

$$x_{n+1} = f(x_n, u) = ux_n(1 - x_n)$$

(3)

Where $u$ is the control parameter, and the range is $[0, 4]$, $x$ values for $[0, 1]$. Starting from $u=3$, the chaotic system starts to bifurcate twice the period from the period window, until $u > 3.56994$ generates chaos and enters the chaos region.

2.3. Chaos quorum sensing particle swarm optimization

By studying luciferase genes in bacteria, Nealson [13] concluded that luciferase genes were inhibited or inactivated in newly inoculated cultures. During the exponential phase of growth, it was activated, and then luciferase was rapidly activated and preferentially synthesized. They call this phenomenon "self-induction." Quorum Sensing (QS) [14] is the regulation of gene expression on cell density fluctuation. The phenomenon of self-induced communication between cells occurs in bacteria and between bacteria. The ability of mutual communication allows bacteria to coordinate gene expression of the whole community and regulating the behavior of the whole community.

Embedding the sensing mechanism of microbial population into PSO algorithm, which refers to the sensing mechanism of bacteria. Based on quorum Sensing (PSOQS) proposed in literature [15], chaos theory is introduced in this paper and we are proposed the chaos quorum Sensing particle swarm algorithm (CPQSOS), the main steps of the algorithm are as follows:

Step 1: Initializes the relevant parameters: the population size of particles, the problem dimensions, and initial population is generated by chaotic iteration;

Step 2: The initialized population was divided into two subpopulations $\text{group}_1$ and $\text{group}_2$ in proportion (1:4), initialized its velocity and position respectively;

Step 3: Under the condition of algorithm termination, carry out the following operations, otherwise turn to Step 7;

Step 4: the $\text{group}_1$ using the PSOQS algorithm, update $p\text{best}_1$ and $g\text{best}_1$, evaluate $f(\text{group}_1)$, and sort the individual optimal values, then select the worst $f_1\text{worst}$; the $\text{group}_2$ continue to chaos search, update $p\text{best}_2$, evaluate $f(\text{group}_2)$ and sort the individual optimal values, then select the best $f_2\text{best}$;

Step 5: Compare the individual historical optima of the two groups, if $f_1\text{worst} > f_2\text{best}$, then $f_2\text{best}$ replace $f_1\text{worst}$

Step 6: update $\text{group}_1$’s $p\text{best}_1$ and $g\text{best}_1$;

Step 7: output the global optimal value $p\text{best}_1$, the end.

The following is the pseudo-code of chaos quorum sensing particle swarm optimization algorithm:
Table 1. chaotic quorum sensing particle swarm optimization pseudocode

| Algorithm1: chaotic quorum sensing particle swarm optimization pseudocode |
|-----------------------------------------------------------|
| 1. Begin:                                                |
| 2. Chaos initializes the population, and divided into $group_1$ and $group_2$ in proportion (2:8) |
| 3. while ($k < maxgen$) /* main loop */                  |
| 4. Evaluate $f(group_1)$, update $pbest_1$ and $gbest_1$  |
| 5. if mod($k$, $g$) == 0                                 |
| 6. $group_1$ quorum sensing (refer [15])                 |
| 7. $group_2$ chaotic search, calculate particle fitness, and update $pbest_2$ |
| 8. Sort $group_1$ and $group_2$, pick the worst values $f_1pworst$ and best values $f_2pbest$ |
| 9. if $f_1pworst > f_2pbest$                             |
| 10. $f_2pbest$ replace $f_1pworst$                       |
| 11. end if                                                |
| 12. end if                                                |
| 13. Output the population optimal value $gbest_1$        |
| 14. end while                                            |
| 15. End                                                  |

3. experimental simulation results

3.1. experimental parameter setting

This paper compares the effect of CPSOQS algorithm with PSOQS algorithm and PSO algorithm on benchmark function. The experimental parameters are set as follows:

The population size of $group_1$ is 100, which used as a quorum sensing; and the population size of $group_2$ is 400, continued chaos research and sorted its fitness by ascend, then selected 100 particles compared with the individual optimal value of $group_1$. The maximum number of iterations is 2000, and each experiment is repeated for 30 times.

3.2. analysis of experimental results

In order to test the effectiveness of algorithm, CEC 2005 benchmark function is used in this experiment to evaluate the algorithm. Among the 25 functions, $F_1$~$F_5$ are single peak function, $F_6$~$F_{25}$ are multiple peak function. Table 2 is the experimental comparison results of various algorithms. The average fitness value, variance, optimal fitness value and maximum fitness value of the three algorithms are recorded. Among them, the algorithms that perform best on each test function are marked in bold.
Table 2. Experimental results of three algorithms.

| algorithms | $F_3$   | $F_5$   | $F_6$   | $F_7$   | $F_{16}$ | $F_{17}$ |
|------------|---------|---------|---------|---------|----------|----------|
| CPSOQS     | Mean    | Std     | Min     | Std     | Min      | Std      |
|            | $-4.50E+02$ | $2.11E-14$ | $-450$  | $2.74E+05$ | $-4.50E+02$ | $-5.06E-14$ |
|            | $1.23E+06$  | $6.66E+05$  | $5.08E+06$ | $3.57E+03$  | $3.68E+06$  | $2.54E+07$  |
|            | $3.57E+03$  | $8.06E+02$  | $1.93E+03$ | $4.35E+02$  | $5.08E+03$  | $3.68E+06$  |
|            | $4.35E+02$  | $4.75E+01$  | $3.90E+02$ | $4.52E+03$  | $5.63E+02$  | $3.83E+02$  |
|            | $4.52E+03$  | $2.92E-12$  | $1.90E+02$ | $4.52E+03$  | $1.37E+03$  | $1.51E+03$  |
|            | $3.52E+02$  | $1.08E+02$  | $2.48E+02$ | $3.52E+02$  | $1.08E+02$  | $3.52E+02$  |
| PSOQS      | Mean    | Std     | Min     | Std     | Min      | Std      |
|            | $-450$  | $5.06E-14$ | $-450$  | $7.30E+05$ | $-4.50E+02$ | $-3.60E+01$ |
|            | $1.52E+06$ | $1.06E+07$ | $2.46E+03$ | $1.19E+03$ | $1.95E+02$ | $1.77E+01$ |
|            | $4.32E+03$ | $3.77E+05$ | $3.90E+02$ | $4.94E+01$ | $3.90E+02$ | $2.46E+03$ |
|            | $4.49E+02$ | $2.46E+03$ | $4.52E+03$ | $5.04E+01$ | $4.52E+03$ | $2.46E+03$ |
|            | $4.52E+03$ | $2.68E-12$ | $2.13E+02$ | $5.04E+01$ | $2.13E+02$ | $5.04E+01$ |
|            | $4.10E+02$ | $1.19E+02$ | $2.84E+02$ | $2.84E+02$ | $1.19E+02$ | $2.84E+02$ |
| PSO        | Mean    | Std     | Min     | Std     | Min      | Std      |
|            | $2.09E+03$ | $8.98E+02$ | $3.79E+03$ | $7.30E+05$ | $-4.50E+02$ | $-3.60E+01$ |
|            | $2.54E+07$ | $1.06E+07$ | $2.23E+04$ | $1.19E+03$ | $1.95E+02$ | $1.77E+01$ |
|            | $1.36E+04$ | $3.37E+03$ | $5.16E+02$ | $5.04E+01$ | $3.90E+02$ | $2.46E+03$ |
|            | $1.20E+08$ | $9.91E+02$ | $9.44E+02$ | $9.44E+02$ | $9.44E+02$ | $9.44E+02$ |
|            | $6.27E+01$ | $9.94E+02$ | $9.45E+02$ | $9.45E+02$ | $9.45E+02$ | $9.45E+02$ |
|            | $5.45E+02$ | $1.08E+02$ | $7.12E+02$ | $7.12E+02$ | $7.12E+02$ | $7.12E+02$ |
|            | $4.22E+02$ | $1.19E+02$ | $6.33E+02$ | $6.33E+02$ | $6.33E+02$ | $6.33E+02$ |

From the results in Table 2 above, among the 14 test functions selected, CPSOQS algorithm has an average optimal value of 11 test functions that are significantly better than PSOQS and PSO, indicating that CPSOQS algorithm has a better search performance than other two algorithms, that is to see chaos search is feasible for the improved algorithm. From the optimization results of multi-modal test function, most of the optimization accuracy of CPSOQS algorithm is significantly better than PSOQS and PSO algorithm. It can be seen that chaos quorum sensing mechanism improves the optimization performance of the function obviously.

In order to reflect the convergence of the three algorithms visually, the comparison diagram of convergence curves of the CPSOQS, PSOQS and PSO algorithm on partial function is shown in the following figure 1. We are only list four comparison graphs, respectively, are single peak function ($F_3, F_5$), and multi-peak function ($F_6, F_{16}$).
4. Conclusion

Basic PSO is easy to fall into local minimum, and the algorithm of particle velocity in the late drops too fast to lose the search ability. Aiming at these problems, this paper introduces chaos search twice in the algorithm, using chaotic motion within a certain range is not repeated search neighborhood of corresponding points, improving the convergence speed and search ability. Combining with the quorum sensing based on microbial behavior mechanism, this paper proposes a quorum sensing algorithm based on chaos. Experiments show that it has significant effect on jumping out of local minimum, improving the search ability and stability.

5. Acknowledgments

This work was financially supported by Open Foundation of Key Laboratory of Software Engineering of Yunnan Province, grant number 2015SE204.

References

[1] Kennedy, James & Eberhart, Russell. (1995). Particle swarm Optimization. Proceedings of IEEE International Conference on Neural Networks. 4. 1942 - 1948 vol.4. 10.1109/ICNN.1995.488968.
[2] Shi Y, Eberhart R C. Parameter selection in particle swarm optimization[C]// International Conference on Evolutionary Programming. Springer, Berlin, Heidelberg. 1998.
[3] Shi Y, Eberhart R C. Empirical study of particle swarm optimization[M]. 2002.
[4] Nobile M S, Cazzaniga P, Besozzi D, et al. Fuzzy Self-Tuning PSO: A Settings-Free Algorithm for Global Optimization[J]. Swarm and Evolutionary Computation, 2017, 39.
[5] Nasir M, Das S, Maity D, et al. A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization[J]. Information Sciences, 2012, 209(none):0-0.
[6] Cheng R, Jin Y. A social learning particle swarm optimization algorithm for scalable optimization[J]. Information Sciences, 2015, 291:43-60.
[7] HU J. Improvement and application of bacterial foraging optimization algorithm[D]. Wuhan University of Technology, 2012.

[8] Aydilek, brahim Berkan. A Hybrid Firefly and Particle Swarm Optimization Algorithm for Computationally Expensive Numerical Problems[J]. Applied Soft Computing, 2018: S156849461830084X.

[9] Lorenz E N. Deterministic Nonperiodic Flow[J]. Journal of Atmospheric Sciences, 2004, 20(2):130-141.

[10] HU X H. Research and application of chaos optimization algorithm[D]. Liaoning Technical University, 2008.

[11] Shi Y, Eberhart R C. Parameter selection in particle swarm optimization[J]. 1998.

[12] XU H L. Research on group intelligence optimization algorithm based on chaotic system[D]. China University of Mining and Technology (Beijing), 2014

[13] Neelson K H, Platt T, Hastings J W. Cellular Control of the Synthesis and Activity of the Bacterial Luminescent System[J]. Journal of Bacteriology, 1970, 104(1):313-322.

[14] Miller M B, Bassler B L. QUORUM SENDING IN BACTERIA[J]. Annual Review of Microbiology, 2000, 55(1):165-199.

[15] CHEN J. Improvement and application of particle swarm optimization based on biological behavior mechanism[D]. South China University of Technology, 2014.