SCIREA Journal of Computer

http://www.scirea.org/journal/Computer

April 1, 2018

Volume 3, Issue 1, February 2018

Hybrid chaotic enhanced acceleration particle swarm optimization algorithm

Mengshan LI[1], Huaijin ZHANG[2], Bingsheng CHEN[3], Lixin GUAN[4], Yan WU[5]

[1,2,3,4,5]College of Physics and Electronic Information, Gannan Normal University, Ganzhou, Jiangxi, China, 341000

Corresponding author: Mengshan LI, jcimsli@163.com

Abstract

In view of the recently proposed acceleration particle swarm optimization with strong global search capability, a chaos enhanced particles swarm optimization algorithm based chaos theory is proposed. Hybrid chaotic sequence is introduced to adjust the global learning factor, and the algorithm can further increase the global search ability. The performance of the algorithm is verified by testing four typical multi-objective optimization functions, and compared with the classic noninferiority classification multi-objective genetic algorithm, multi-objective particle swarm optimization algorithm and acceleration particle swarm optimization algorithm. The result shows that the Hybrid chaotic acceleration particle swarm optimization algorithm has faster convergence speed and stronger ability to jump out of local optimization, and the performance is superior.

Keywords: Particle swarm; Chaotic enhanced; Acceleration algorithm

1. INTRODUCTION

Particle swarm optimization(PSO) algorithm is a swarm intelligence optimization algorithm, it is the aggregation of biological behavior. Compared with other intelligent algorithm, PSO algorithm has simple structure, less parameters and is easy to describe and implement, the global search ability is stronger, without gradient information, and many other features in the function optimization, a multi-objective problem solving, such as pattern recognition is widely used in various fields[1-4]. However, the standard PSO algorithm also has shortcomings such as precocious convergence and bad local search capability similar to other intelligent algorithms[5-7]. If the application is optimized in high dimensional complex problems, the population may have accumulated to a certain point of stagnation without searching for the global optimum, and the problem of precocious convergence is formed[8]. At the same time, in the search process of PSO algorithm, the convergence rate becomes slow when the particle is approaching or entering the most advantageous region, that is, the search capability is poor in the latter. Thus, the application of PSO algorithm is restricted[9-12].

For the deficiency of PSO algorithm, the researchers propose many improvement strategies[13] . Among them, introduce the inertia weight factor, contraction factor and adaptive mutation operator are the most representative. Such as the inertia coefficient adaptive adjustment method of linear degression method[14], fuzzy adaptive method[15] and distance information method[16, 17]; PSO algorithm with compression factor; the PSO algorithm of adaptive mutation operator. In addition, the PSO algorithm and collaborative strategy[18], chaos theory[19] and other algorithms[20] combine to form a hybrid PSO algorithm which is also attracted by the researchers. As quantum PSO algorithm with chaotic mutation operator; in addition, there are many researches on discrete PSO algorithm and multi-objective PSO algorithm[21-23]. At present, the improvement of PSO algorithm is mainly focused on two aspects: adjustment of algorithm parameters and update of particle structure and trajectory. The aim is to improve the performance of the algorithm solve or improve the problems of local search slow, precocious convergence, and improve the convergence speed and precision of the algorithm. Although the proposed particle swarm improvement algorithm improves both in performance and efficiency, but it is difficult to improve the local search ability of the algorithm while avoiding precocious convergence. To provide better, more efficient, and cheaper particle swarm algorithms, academics and industry researchers have been exploring and experimenting with new approach[24-27].

In recent years, an improved variant with extremely strong global convergence, called accelerated PSO (APSO) [25, 28-30], has attracted the attention of scholars. The main idea is to fully consider the global excellent, and the particle is only constrained by global extremum during the whole search process, thus speeding up the search speed. However, while the APSO increases the convergence speed, the algorithm still has the precocious convergence problem and may miss some extreme values. For this purpose, this paper proposes a new optimization algorithm for chaos enhancement acceleration particle swarm optimization algorithm (CAPSO) by integrating Hybrid chaos theory into APSO algorithm. The global learning factor of APSO algorithm is optimized by the chaotic sequence generated by Hybrid theory, so that the convergence accuracy can be improved when it enters precocious convergence. Finally, the algorithm of multi-objective optimization is analyzed and compared with other algorithms to verify the performance of CAPSO algorithm.

2. CAPSO

2.1 Standard PSO Algorithm

Particle Swarm Optimization algorithm (PSO) was founded in 1995[31-33]. In the optimization of PSO algorithm, the potential solution of the problem is assumed to be a "particle" in the n-dimensional space, and the particle will fly at a certain speed and direction in the solution space. In the process of iteration, all particles are expressed in two global variables to the best position of the particle itself (*pbest*), also known as the individual extreme value and the best position of all particles (*gbest*), also known as the global extremu. Suppose in an n-dimensional search space, a population X of m particles, $X = \left(x_{1,} x_{2,} \ldots, x_n\right)^T$, the position of the i particle is represented $x_i = \left(x_{i,1,} x_{i,2,} \ldots, x_{i,n}\right)^T$, the speed is represented $v_i = \left(v_{i,1,} v_{i,2,} \ldots, v_{i,n}\right)^T$. Its individual extreme value is $p_i = \left(p_{i,1,} p_{i,2,} \ldots, p_{i,n}\right)^T$, the global extremum of the particle population is $p_g = \left(p_{g,1,} p_{g,2,} \ldots, p_{g,n}\right)^T$, In the process of iteration k+1, particle by formula (1) and (2) to update their speed and position.

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

$$x_{i,d}^{k+1} = x_{i,d}^{k} + v_{i,d}^{k+1} \quad (2)$$

Among, i=1,…,m; ω is called an inertial weight factor, it keeps the particle motion inertia, make its have ability to expand the search space; C_1 and C_2 are learning factors, on behalf of each particle to the extreme value position statistical accelerate the weight of items. rand() is a random numbers between (0, 1), $v_{i,d}^{k}$, $x_{i,d}^{k}$ are velocity and position of d dimension of particle i in the k iteration. $p_{i,d}^{k}$ is position of the individual extremum of particle i in d dimension, $p_{g,d}^{k}$ is global extreme value of the group in d dimension.

The standard PSO algorithm has its own limitations, such as the algorithm implementation process and the value of parameters has a great relationship; when the algorithm is applied to the complex optimization problem of high dimension, the algorithm tends to converge to some extreme point stagnation when the global optimum position is not found, which is easy to get precocious convergence. These points can be a point in the local extreme point or in local extreme point area. In addition, the convergence rate of the algorithm becomes slow when approaching or entering the optimal solution area. The early convergence rate of the PSO algorithm is fast, but in the later stage, due to the lack of effective local search mechanism in the local polar hour, the local search speed is slow.

2.2 CAPSO algorithm

In accelerating particle swarm optimization (APSO), will not be considered inertia weight factors and cognitive factors of influence on particle, only use the contribution to the global exploration factor to improve the algorithm, the main idea of the algorithm is fully responsible for global search variables only unique rights, fully considering the exploration factor updates of the particles, particles in the process of the whole search only by global extremum constraints, so as to accelerate the search speed. The velocity formula is as follows:

$$v_{i,d}^{k+1} = v_{i,d}^{k} + c_1 r(t) + c_2 (p_{g,d}^{k} - x_{i,d}^{k}) \quad (3)$$

r is the random number in (0,1). In APSO, the velocity items can be found to be negligible, and the formula for the location update is as follows:

$$x_{i,d}^{k+1} = (1 - C_2)x_{i,d}^{k} + C_2 p_{g,d}^{k} + C_1 r \tag{4}$$

C_1r is random number, it can provide an algorithm to jump out of local optimal use.

Compared with the standard PSO algorithm, APSO use two parameters C_1 and C_2, to reduce randomness in the iterative process, C_1 is expressed as monotone decreasing function: $C_1 = \delta^t$, $0 < \delta < 1$, t is current iteration number. So, in APSO algorithm, the performance of the algorithm is mainly affected by parameter C_2, for general problems, its value is in [0.2, 0.7]. When C_2 is 1, the particle can converge to the current global extreme at any time, and this global extreme may not be the real global extreme; Conversely, when C_2 is 0, the global search speed of the algorithm is extremely slow. Therefore, it is very important to optimize the optimization of C_2 by analyzing the performance of APSO algorithm.

However, while the APSO increases the convergence speed, the algorithm still has the precocious convergence problem and may miss some extreme values. From the analysis of the change characteristics of learning factor C_2 in APSO, its characteristics can be described by chaotic mapping, and chaos theory can be used to optimize parameter C_2. Therefore, this paper uses Hybird mapping equation to generate chaotic sequences to achieve the optimization of parameter C_2. The formula is as follows:

$$x_{k+1} = \begin{cases} b(1 - \mu_1 x_k^2), -1 < x_k \leq 0 \\ 1 - \mu_2 x_k, 0 < x_k < 1 \end{cases} \tag{5}$$

When the parameters are $\mu_1 = 1.8, \mu_2 = 2.0, b = 0.85$, maps to a chaotic state.

The steps of CAPSO algorithm:

1) Initialize the particle group.

In the PSO algorithm, the particle is initialized and the optimal value is selected as the global optimal value. And it creates a chaotic value.

2) Calculate the adaptive value of group particles.

3) The adaptive value of each particle is compared with the adaptive value of the best position of its own. If it is better, the current position is the best position of the particle.

4) The adaptive value of each particle is compared with that of the global best position, and if it is better, the current position is the best position in the global.

5) The learning factor C_2 is obtained from the Hybrid chaotic sequence (derived from formula (5)), and the position of the particle is updated by formula (4).

6) If the end condition of the algorithm is satisfied, the global best position is the optimal solution, saving the result and ending. Otherwise return steps (2).

3. Numerical Experiment

3.1 Experiment function and evaluation

In order to test the performance of CAPSO algorithm, this paper selects the multi-objective optimization test function proposed by Schaffer and Deb [34-38] as an experimental case. The solutions to such problems are usually not unique, but there are a series of optimal solutions, also called non-inferior solutions, and the collection of non-inferior solutions is often call Pareto optimal solution. Because intelligent algorithm can search multiple solutions of solution space in parallel, so multi-objective optimization is more suitable to verify the performance of intelligent algorithm. The multi-objective optimization functions used in this article are shown in Table 1.

Table 1 Experimental test function

Function	Definition
SCH1	$\min f(x) = (f_1(x), f_2(x)) s.t. x \in [-5, 7]$ $f_1(x) = x^2 \quad f_2(x) = (x-2)^2$
SCH2	$\min f(x) = (f_1(x), f_2(x)), s.t. x \in [-5, 10]$ $f_1(x) = \begin{cases} -x, (x \le 1) \\ -2+x, (1 < x \le 3) \\ 4-x, (3 < x \le 4) \\ -4+x, (x > 4) \end{cases} \quad f_2(x) = (x-5)^2$

ZDT2	$\min f(x) = (f_1(x), f_2(x)), s.t.x \in [-5, 10]$ $f_1(x) = x_1 \qquad f_2(x) = g(x)\left[1 - \left(x_1 / g(x)\right)^2\right]$ $g(x) = 1 + 9(\sum_{i=2}^{n} x_i) / (n-1)$
ZDT3	$\min f(x) = (f_1(x), f_2(x)), s.t.x_i \in [0, 1]$ $f_1(x) = x_1 \qquad f_2(x) = g(x)[1 - \sqrt{x_1 / g(x)} - \frac{x_1}{g(x)} \sin(10\pi x_1)]$ $g(x) = 1 + 9(\sum_{i=2}^{n} x_i) / (n-1)$

In order to evaluate the merits of non-inferior solutions, this paper adopts convergence index and distribution index to evaluate the performance of the algorithm, and the convergence and distribution of homogeneity are defined as follows:

1) The convergence index(GD), GD is used to describe the distance between the ungoverned solution and the optimal front end of the real Pareto algorithm.

$$GD = \frac{\sqrt{\sum_{i=1}^{N} d_i^2}}{N} \tag{6}$$

Among them, N represents the number of non-dominant solutions that the algorithm searches for, d_i^2 indicating the shortest Euclidean distance of all solutions in the non-inferior solution i and the optimal front-end of the real Pareto.

2) The distributive index (SP), SP is used to evaluate the uniformity of distribution of ungoverned solution.

$$SP = \frac{[\frac{1}{n}\sum_{i=1}^{n}(\bar{d}-d_i)^2]^{1/2}}{\bar{d}} \tag{7}$$

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i \tag{8}$$

Among them, n is the number of non-dominant solution, d_i indicating the shortest Euclidean distance of all solutions in the i non - inferior solution in the target space and the optimal front-end of the real Pareto.

3.2 Experimental Result

The proposed CAPSO algorithm was used to experiment with SCH1, SCH2, ZDT2 and ZDT3. The algorithm parameter is set to: particle size 50; the maximum iteration number is set 200. The Pareto non-inferior solutions of each function are shown in Figure 1-4.

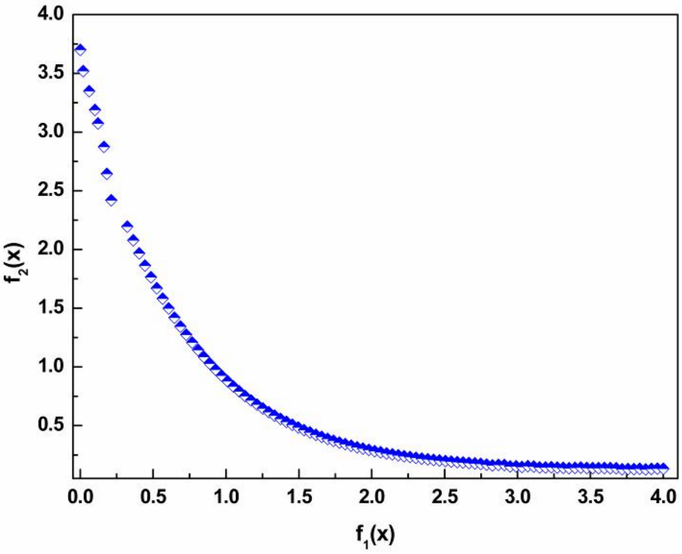

Figure 1 CAPSO algorithm solves the Pareto non-inferior solution of SCH1 function

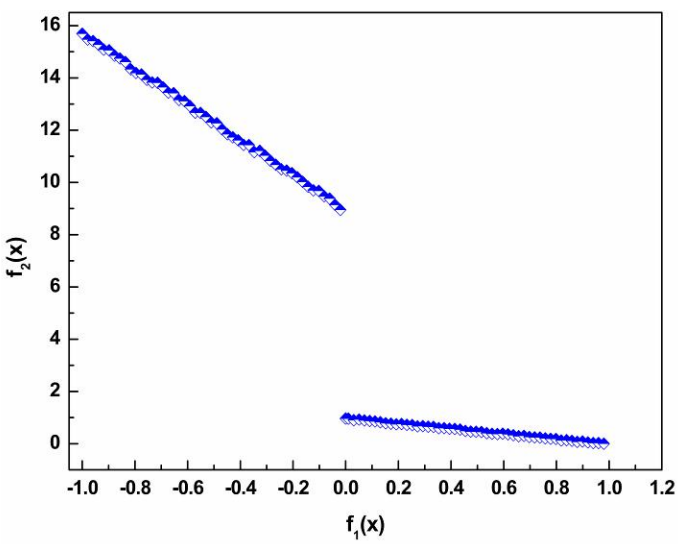

Figure 2 CAPSO algorithm solves the Pareto non-inferior solution of SCH2 function

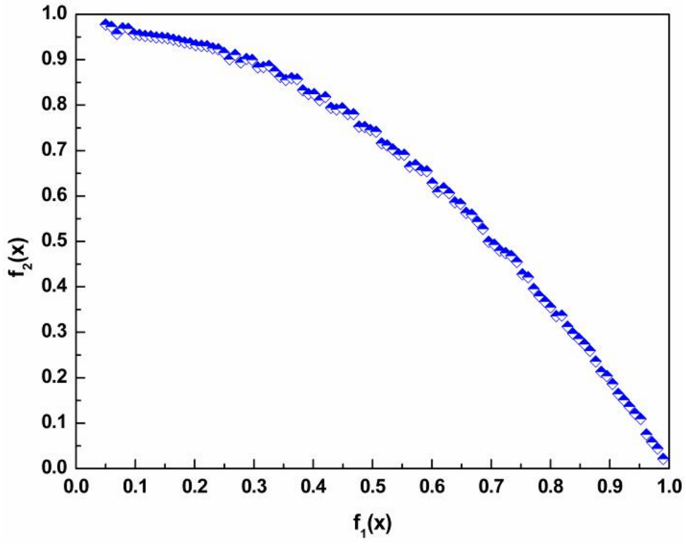

Figure 3 CAPSO algorithm solves the Pareto non-inferior solution of ZDT2 function

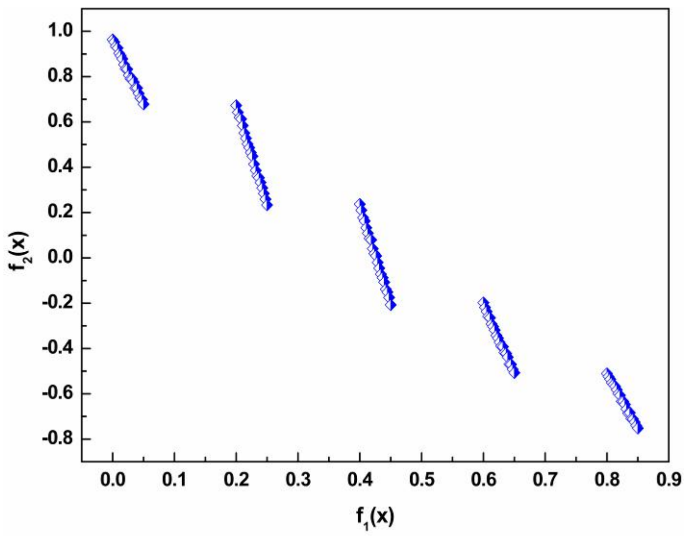

Figure 4 CAPSO algorithm solves the Pareto non-inferior solution of ZDT3 function

In the target function space, the non-inferior optimal target domain is the boundary of the fitness value region, which is the effective interface. It can be seen from the experimental results that the four test functions accurately give the effective interface, and the complete Pareto non-inferior solution can be obtained. Particularly the discrete problem of ZDT3, and the algorithm also gives an accurate non-inferior solution. In general, the algorithm has a lot of Pareto solutions, and distribution is more uniform. The accuracy and reliability of the CAPSO algorithm are illustrated.

Through CAPSO algorithm runs 30 times for each test function, statistics of the convergence index GD, distribution index SP and calculating the average time of CT, and four statistical test function evaluation index in each test function, the results are shown in Table 2.

Table 2 CAPSO optimize the performance statistics of test functions

Index	SCH1	SCH2	ZDT2	ZDT3	Average
GD[a]	0.000332	0.000315	0.000321	0.000311	0.00032
SP[b]	0.00322	0.00321	0.00312	0.00316	0.00318
CT[c]	2.1	2.4	2.8	3.2	2.6

[a] is convergence index, [b] is distribution index, [c] is computation time

The evaluation index GD, SP and CT confirmed the feasibility, accuracy and efficiency of CAPSO algorithm for solving multi-objective optimization problems. GD shows that the non-inferior solution is very close to the optimal front end of the real Pareto; SP shows that the non-inferior solution has good distribution; CT shows that the time spent running is within acceptable limits.

In order to test the superiority of the algorithm in multi-objective optimization solution. In this paper, CAPSO algorithm compare with the non-inferior multi-objective genetic algorithm(NSGA - Ⅱ), multi-objective particle swarm optimization algorithm(MOPSO) and acceleration particle swarm optimization (APSO), statistical comparison results as shown in Table 3.

Table 3 Comparison of the results of four algorithms to optimize test functions

Index	NSGA II[a]	MOPSO[b]	APSO	CAPSO
GD	0.000544	0.000811	0.000642	0.00032
SP	0.00601	0.00722	0.00631	0.00318
CT	13.8	12.7	3.1	2.6

[a] Non-inferior multi-objective genetic algorithm, [b] Multi-objective particle swarm optimization algorithm

According to GD from table 3, the convergence of CAPSO algorithm is better than other three algorithms, indicates that the optimal front-end distance between the non-inferior solution and real Pareto is smaller, that is the solution is closer to the real solution; Especially for APSO, the performance of CAPSO algorithm has made a lot of improvement, analysis its reason, CAPSO algorithm has good performance to through the introduction of chaotic sequence, the improved into local extremum problems. SP from table 3 shows that the distribution of non-inferior solutions of CAPSO algorithm is better, that is, the non-inferior distribution of the algorithm is more uniform than the other two algorithms. In the same way, good distribution is also attributed to chaotic sequences. On CT, CAPSO and APSO algorithm of computing time are much smaller than the NSGA-Ⅱ and MOPSO; Especially for MOPSO, the algorithm that introduced the acceleration mechanism reduced the time by more than half. From the calculation time analysis, APSO only introduces global factor, to search the global optimal target, the purpose of the particle search is clear, reduce the computation time. Compared with the calculation time of CAPSO and APSO, the calculation time of chaos sequence is slightly higher, because the chaos sequence expands the search scope of the particle, which is bound to increase the search time. However, in terms of performance, it is obvious that CAPSO is better than APSO, and its convergence of non-inferior solution and distribution are better than other algorithms, which can obtain a feasible solution with high quantity, high accuracy and even more uniform distribution.

In general, through CAPSO algorithm for numerical experiments of four multi-objective optimization problem, and compared with the classical multi-objective optimization NSGA-II algorithm, the MOPSO algorithm and APSO algorithm, CAPSO algorithm has a better comprehensive performance, algorithm through the chaos enhancement mechanism to improve the convergence precision, improve the premature convergence of the algorithm.

4. CONCLUSION

In this paper, a Hybrid chaotic enhanced acceleration particle swarm optimization algorithm is proposed(CAPSO). The algorithm uses chaos theory and acceleration mechanism to improve the PSO algorithm and improve the precocious convergence of PSO algorithm, which greatly improves the convergence speed. By the experiment of four standard test functions, the proposed algorithm can be used to solve the multi-target problem, and the obtained non-inferiority solution can get a good approximation of the optimal solution set of Pareto and distribute evenly. By comparing with other algorithms, CAPSO algorithm can provide practical reference value for many optimization problems in engineering.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support from the National Natural Science Foundation of China (Grant Numbers: 51663001, 51463015).

REFERENCES

[1] Zhu Q.L., Lin, Q.Z., Chen, W.N., Wong, K.C., Coello, C.A.C., Li, J.Q., Chen, J.Y., Zhang, J. An External Archive-Guided Multiobjective Particle Swarm Optimization Algorithm. IEEE Transactions on Cybernetics, 2017, 47(9): 2794-2808.

[2] Neri F., Mininno, E., Lacca, G. Compact Particle Swarm Optimization. Information Sciences, 2013, 239: 96-121.

[3] Zitzler E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G. Performance assessment of multiobjective optimizers: An analysis and review. Ieee Transactions On Evolutionary Computation, 2003, 7(2): 117-132.

[4] Shirazian S., Alibabaei, M. Using neural networks coupled with particle swarm optimization technique for mathematical modeling of air gap membrane distillation (AGMD) systems for desalination process. Neural Computing & Applications, 2017, 28(8): 2099-2104.

[5] Tsekouras G.E., Tsimikas, J. On training RBF neural networks using input-output fuzzy clustering and particle swarm optimization. Fuzzy Sets And Systems, 2013, 221: 65-89.

[6] Yan J., He, W.X., Jiang, X.L., Zhang, Z.L. A novel phase performance evaluation method for particle swarm optimization algorithms using velocity-based state estimation. Applied Soft Computing, 2017, 57: 517-525.

[7] Javidrad F., Nazari, M. A new hybrid particle swarm and simulated annealing stochastic optimization method. Applied Soft Computing, 2017, 60: 634-654.

[8] Kiran M.S. Particle swarm optimization with a new update mechanism. Applied Soft Computing, 2017, 60: 670-678.

[9] Han H.G., Lu, W., Qiao, J.F. An Adaptive Multiobjective Particle Swarm Optimization Based on Multiple Adaptive Methods. IEEE Transactions on Cybernetics, 2017, 47(9): 2754-2767.

[10] Gou J., Lei, Y.X., Guo, W.P., Wang, C., Cai, Y.Q., Luo, W. A novel improved particle swarm optimization algorithm based on individual difference evolution. Applied Soft

Computing, 2017, 57: 468-481.
[11] Liu Q.X., van Wyk, B.J., Du, S.Z., Sun, Y.X. Dynamic Small World Network Topology for Particle Swarm Optimization. International Journal Of Pattern Recognition And Artificial Intelligence, 2016, 30(9).
[12] Gong Y.J., Li, J.J., Zhou, Y.C., Li, Y., Chung, H.S.H., Shi, Y.H., Zhang, J. Genetic Learning Particle Swarm Optimization. IEEE Transactions on Cybernetics, 2016, 46(10): 2277-2290.
[13] Akay B. A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. Applied Soft Computing, 2013, 13(6): 3066-3091.
[14] Khare A., Rangnekar, S. A review of particle swarm optimization and its applications in Solar Photovoltaic system. Applied Soft Computing, 2013, 13(5): 2997-3006.
[15] Valdez F., Melin, P., Castillo, O. An improved evolutionary method with fuzzy logic for combining Particle Swarm Optimization and Genetic Algorithms. Applied Soft Computing, 2011, 11(2): 2625-2632.
[16] Marinakis Y., Marinaki, M. Particle swarm optimization with expanding neighborhood topology for the permutation flowshop scheduling problem. Soft Computing, 2013, 17(7): 1159-1173.
[17] Zhang Z.B., Jiang, Y.Z., Zhang, S.H., Geng, S.M., Wang, H., Sang, G.Q. An adaptive particle swarm optimization algorithm for reservoir operation optimization. Applied Soft Computing, 2014, 18: 167-177.
[18] Sinha A.K., Zhang, W.J., Tiwari, M.K. Co-evolutionary immuno-particle swarm optimization with penetrated hyper-mutation for distributed inventory replenishment. Engineering Applications Of Artificial Intelligence, 2012, 25(8): 1628-1643.
[19] Cheng M.Y., Huang, K.Y., Chen, H.M. K-means particle swarm optimization with embedded chaotic search for solving multidimensional problems. Applied Mathematics And Computation, 2012, 219(6): 3091-3099.
[20] Li M.W., Kang, H.G., Zhou, P.F., Hong, W.C. Hybrid optimization algorithm based on chaos, cloud and particle swarm optimization algorithm. Journal of Systems Engineering and Electronics, 2013, 24(2): 324-334.
[21] Qasem S.N., Shamsuddin, S.M., Hashim, S.Z.M., Darus, M., Al-Shammari, E. Memetic multiobjective particle swarm optimization-based radial basis function network for classification problems. Information Sciences, 2013, 239: 165-190.
[22] Zheng Y.J., Chen, S.Y. Cooperative particle swarm optimization for multiobjective

transportation planning. Applied Intelligence, 2013, 39(1): 202-216.
[23] Wang L., Yang, B., Chen, Y.H. Improving particle swarm optimization using multi-layer searching strategy. Information Sciences, 2014, 274: 70-94.
[24] Tsekouras G.E. A simple and effective algorithm for implementing particle swarm optimization in RBF network's design using input-output fuzzy clustering. Neurocomputing, 2013, 108: 36-44.
[25] Mac T.T., Copot, C., Tran, D.T., De Keyser, R. A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization. Applied Soft Computing, 2017, 59: 68-76.
[26] Jiang F., Xia, H.Y., Tran, Q.A., Ha, Q.M., Tran, N.Q., Hu, J.K. A new binary hybrid particle swarm optimization with wavelet mutation. Knowledge-based Systems, 2017, 130: 90-101.
[27] Li P.C., Xiao, H. An improved quantum-behaved particle swarm optimization algorithm. Applied Intelligence, 2014, 40(3): 479-496.
[28] Wang Y., Li, Y.Y., Chen, Z.H., Xue, Y. Cooperative particle swarm optimization using MapReduce. Soft Computing, 2017, 21(22): 6593-6603.
[29] Meng M., Rizvi, M.J., Le, H.R., Grove, S.M. Multi-scale modelling of moisture diffusion coupled with stress distribution in CFRP laminated composites. Composite Structures, 2016, 138: 295-304.
[30] Zhao L.-P., SHU, Q.-L., WU, Y., LI, M.-S. Chaos-enhanced Accelerated Particle Swarm Optimization Algorithm. Application Research of computers, 2014, 31(08): 2307-2310.
[31] Kennedy J., Eberhart, R., Particle swarm optimization, in: 1995 IEEE International Conference on Neural Networks Proceedings, Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE Australia Council, Perth, 1995, pp. 1942-1948.
[32] Kennedy J., Eberhart, R., Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks. Part 1 (of 6), IEEE, Perth, Aust, 1995, pp. 1942-1948.
[33] Eberhart R., Kennedy, J., New optimizer using particle swarm theory, in: Proceedings of the 1995 6th International Symposium on Micro Machine and Human Science, October 4, 1995 - October 6, 1995, IEEE, Nagoya, Jpn, 1995, pp. 39-43.
[34] Schaffer J.D., Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, in: International Conference on Genetic Algorithms - ICGA 85, Lawrence Erlbaum Associates, Netherlands, 1985, pp. 93-100.

[35] Zitzler E., Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the Strength Pareto approach. Ieee Transactions On Evolutionary Computation, 1999, 3(4): 257-271.

[36] Deb K. Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. Evolutionary Computation, 1999, 7(3): 205-230.

[37] Zitzler E., Deb, K., Thiele, L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation, 2000, 8(2): 173-195.

[38] Deb K., Pratap, A., Agarwal, S., Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. Ieee Transactions On Evolutionary Computation, 2002, 6(2): 182-197.