SCIREA Journal of Computer

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

January 2, 2018

Volume 3, Issue 1, February 2018

Chaos Adaptive Improved Particle Swarm Optimization Algorithm and its Multi-objective Optimization Solution

CHEN Bingsheng[1], LIU Liang[2], SU Keming[3], ZHANG Huaijin[4], LI Mengshan[5]

[1,2,3,4,5]Gannan Normal University, Ganzhou, Jiangxi 341000, China

Abstract

To overcome the problem of premature convergence on particle swarm optimization (PSO), this paper proposes an improved particle swarm optimization method (IPSO) that based on self-adaptive regulation strategy and chaos theory. For a given the effective balance of particles' searching and development ability, self-adaptive regulation strategy is employed to optimize the inertia weight. To improve efficiency and quality of search, learning factor is optimized by generating Chaotic Sequences by Chaos Theory. The proposed improved methods achieve better convergence performance and increases searching speed. Simulation results of some typical optimization problems and comparisons with typical multi-objective optimization algorithms show that IPSO has an ability of fast convergence speed, and the diversity of non-dominated and the convergence are ideal. The algorithm meets requirements of multi-objective optimization Problem.

Keywords： Particle Swarm Optimization, multi-objective optimization, Chaos Theory, self-adaptive regulation strategy

1. INTRODUCTION

Multi-objective optimization problems (MOP) is a common problem in engineering field. The problem is composed of multiple targets to be optimized. The main feature is that there is a conflict among the objectives, which cannot achieve optimal value at the same time[1-5]. For example, when designing a new product, it is usually required to consider factors such as production, quality, cost, consumption and profit. In order to achieve high yield, high quality, low consumption, low cost and high profit, it is necessary to establish optimum design model with multiple objectives. At present, the MOP method can be divided into the traditional multi-objective optimization method and the multi-objective optimization method based on group intelligent algorithm. Traditional methods include evaluation function method[6-9], interactive programming method[10], layered solution method[11], etc. Their essence is to transform each sub-objective function of the multi-objective optimization into a single objective function to solve it. These traditional methods have encountered many difficulties in solving high-dimensional complex multi-objective problem. For example, different properties of sub-target units are different and cannot be compared, and the results are relatively unsatisfactory[12]. In recent years, the swarm intelligence algorithm for multi-objective optimization is widely concerned by many scholars, and a number of multi-objective optimization algorithms based on various swarm intelligence algorithms are proposed [13-17]. For example, Fonseca[18] put forward the MOGA algorithm. This algorithm relies too much on the selection of shared function, and needs to determine the shared radius, and may produce a large selection pressure, leading to premature convergence. Horn[19] proposed NPGA algorithm. However, the algorithm is difficult to adjust and select the small habitat radius, and also choose a suitable comparison set scale, which makes the optimization result not ideal. Srinivas[20] proposed NSGA algorithm. The advantage of this algorithm is that multiple optimization targets can be selected arbitrarily, and the non-inferior optimal solution that distribution is uniform can be obtained. The disadvantage is that the computational efficiency is low and the computational complexity is O (N^3) (where M is the target number and N is the population size). Zitzler[21] proposed Multi-objective evolution algorithm SPEA algorithm. This algorithm adopts elite retention strategy, which has high computational efficiency, but its computational complexity is as high as the population's cube. Deb[22] proposed the second generation of NSGA , NSGA- II. The computational complexity is O(N^2), and the computational complexity of the algorithm is significantly reduced. This algorithm introduces fast non-inferior sorting and new diversity protection methods to overcome the shortcomings of NSGA. It is helpful to put forward the Crowded concept and reference elite strategy, which is beneficial to maintain the uniformity of good individual reconciliation and improve the overall evolution of the population. NSGA-II algorithm is a typical multi-objective optimization algorithm, many researches are based on the NSGA-II algorithm. But there are some problems with this algorithm, such as, the search space changes adaptability is poor, the algorithm is easy to precocious convergence, particle search randomness is stronger in the process of iterative, resulting in low search efficiency, etc.

In the last decade, The application of Particle Swarm Optimization algorithm(PSO) in multi-objective optimization has received wide attention from many scholars. PSO and Genetic Algorithm (GA) are the group intelligent algorithms, but compared with GA, PSO has the characteristics of easy implementation, less parameter adjustment, strong global search ability, and so on, which are favored by researchers. However, when applying particle swarm algorithm to solve multi-objective optimization problems, the researchers found that: In the optimization process, the particle swarm optimization is easy to show the problem of precocious convergence, in particular, it is more difficult to solve the complex objective function of higher dimensions. And the convergence rate is very slow when approaching or entering the most advantageous region, resulting in a local extreme value, and the result is not ideal. Aiming at the premature convergence of PSO, the scholars at home and abroad have proposed a variety of improvement schemes. For instance, Yuhui [23] proposed fuzzy adaptive particle swarm optimization algorithm. Zhou[24] adopted the fuzzy membership function and adaptive adjustment strategy to improve the PSO algorithm, and proposed the Adaptive Focusing Particle Swarm Optimization algorithm (AFPSO). He[25] introduced passive aggregation factor into particle swarm optimization algorithm to preserve the integrity of the population and proposed the Particle Swarm Optimization With Passive Congregation algorithm (PSOPC). Most of the improved algorithms are improved by adjusting the parameters of the algorithm, such as introducing various linear and nonlinear inertial weight dynamic adjustment strategies and introducing contraction factors and so on. The proposed particle swarm improvement algorithm has improved performance and efficiency, but there is still a large room for improvement, therefore, to provide a particle swarm algorithm with better performance, higher efficiency, and lower cost, academic and industry researchers have been exploring and trying new improvement way[26-31].

In view of the above research questions, this paper adopts chaos theory and adaptive weight adjustment strategy to improve the standard particle swarm optimization, and proposes a chaos adaptive Improved Particle Swarm Optimization algorithm (IPSO). The algorithm evolves the inertia weight through adaptive adjustment strategy, optimizes the learning factor by the chaotic sequence generated by chaos theory. The improved particle swarm optimization algorithm can improve the algorithm's precocious problem and improve the searching speed of the algorithm. Finally, the algorithm is applied to multi-objective optimization problem solving to verify the performance of convergence accuracy , speed , global convergence and so on.

2. Chaos Adaptive Particle Swarm Optimization algorithm

2.1 Standard particle swarm optimization algorithm

Particle Swarm Optimization algorithm (PSO) [32] is a group evolution algorithm proposed by scholars Eberhart and Kennedy[33] based on the social behaviors of birds in 1995. The PSO algorithm is derived from the behavior characteristics of biological groups and is used to solve the optimization problems. In the process of particle optimization, 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 iterative process, all particles use two global variables to represent the best position of the particle itself (pbest) and the best position of all particles (gbest). It is assumed that, in an n-dimensional search space, the particle population $X=\left(x_1,x_2,\ldots,x_n\right)^T$ is composed of m particles. The position of the ith particle is denoted as $x_i=\left(x_{i,1},x_{i,2},\ldots,x_{i,n}\right)^T$ and the velocity is denoted as $v_i=\left(v_{i,1},v_{i,2},\ldots,v_{i,n}\right)^T$. The individual extremum is $p_i=\left(p_{i,1},p_{i,2},\ldots,p_{i,n}\right)^T$, The global extreme of the population of particles is $p_g=\left(p_{g,1},p_{g,2},\ldots,p_{g,n}\right)^T$. During the k+1 iteration, the particle updates its speed and position through formula (1) and (2).

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

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

Where, ω is called an inertial weight factor, it makes the particles keep sport inertia and have the ability to expand search space; C_1 and C_2 are the learning factors, which represent the weight of each particle to the statistical acceleration item of the extremum position; rand () is a random number within (0, 1), $v_{i,d}^{k}$ and $x_{i,d}^{k}$ are respectively the velocity and position of particle i in d -dimension kth iteration; $p_{i,d}^{k}$ is the position of the individual extremum of particle i in d-dimension, $p_{g,d}^{k}$ is the position of the global extremum of the whole population in d-dimension.

2.2 Chaos adaptive particle swarm algorithm

PSO algorithm has the advantages of easy description, easy implementation, little adjustment parameter, fast convergence speed, low calculation cost ,etc. And there is no high requirement for memory and CPU speed. It has been proved to be an effective method to optimize the problem. But the standard PSO algorithm has its own limitations, such as the implementation process of the algorithm has a great relationship with the value of the parameters. In the complex optimization problem of high dimension, the algorithm is easy to converge to some extreme point and stagnates when the global optimum is not found, that is, precocious convergence. In addition, the convergence rate of the algorithm becomes slow when approaching or entering the optimal solution area.

Aiming at the above shortcomings of PSO algorithm, in order to improve the precocious convergence of the algorithm and improve the convergence speed of the algorithm, this paper uses adaptive weight adjustment strategy and chaos theory to optimize inertial weight factor ω , learning factors C_1, C_2 parameters of standard PSO algorithm, and a chaotic self-adaptive improved particle swarm optimization algorithm (IPSO) is obtained. The inertial weight factor ω is adjusted by Formula (3).

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min})\left\{\frac{(iter_{\max} - iter)^{n}}{(iter_{\max})^{n}}\right\} \tag{3}$$

Where, iter is the current iteration number of the algorithm, $iter_{max}$ is the maximum number of iterations that the PSO algorithm is allowed to perform. n is a nonlinear modulation index.

In the process of optimization iteration of PSO algorithm, the learning factor C_1 and C_2 are adjusted by chaotic sequences generated by chaos theory. Because the change of chaotic variables is random, ergodic and regular, the IPOS algorithm can maintain the diversity of population, overcome the problem of precocious convergence, and improve global search performance. This paper uses the typical Lorenz's equation to realize the evolution of chaotic variables and optimize search. As shown in Formula (4).

$$\begin{cases} \frac{dx}{dt} = -(y+z) \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + xz - cz \end{cases} \tag{4}$$

where, parameters a, r and b are controlled parameters. In the Lorenz equation, the effective values of a, r and b are respectively a= 10, r=28 and b=8/3.

The IPSO algorithm performs the following process:

(1) To initialize the particle group

The position and velocity of particles in PSO algorithm are initialized. The initial position and velocity of the particles are generated randomly. The current position of each particle is used as the particle individual extremum, and the optimal value of the individual extremum is selected as the global optimal value.

(2) To calculate the adaptive value of group particles.

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

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

(5) The learning factor C_1, C_2 and inertial weight ω were calculated respectively, and the new inertial weight and learning factor were obtained, and the velocity and position of the particles were optimized.

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 to Step (2).

3. Experimental results and analysis

3.1 Experimental results

In order to test the performance of chaotic adaptive particle swarm algorithm, this paper selects the two standard test functions proposed by Schaffer[34] and the ZDT3 proposed by Deb [13] as the test case; The test function is shown below.

Test function SCH1：

$$\min f(x) = (f_1(x), f_2(x))$$
$$s.t. x \in [-5, 7]$$

Where: $f_1(x) = x^2$， $f_2(x) = (x-2)^2$

Test function SCH2：

$$\min f(x) = (f_1(x), f_2(x))$$
$$s.t. x \in [-5, 10]$$

Where: $f_1(x) = \begin{cases} -x, (x \le 1) \\ -2 + x, (1 < x \le 3) \\ 4 - x, (3 < x \le 4) \\ -4 + x, (x > 4) \end{cases}$， $f_2 = (x-5)^2$

Test function ZDT3：

$$\min f(x) = (f_1(x), f_2(x))$$
$$s.t. x_i \in [0, 1]$$

Where: $f_1(x) = x_1$ ， $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)} - \frac{x_1}{g(x)}\sin(10\pi x_1)]$ ，

$$g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$$

Through the IPSO algorithm presented in this paper, The test function SCH1, SCH2 and ZDT3 are respectively carried out the simulation optimization experiment. The number of particles and the number of iterations are set to 100 and 50 times, the inertia weight ω and the learning factor C_1 and C_2 are calculated according to the Formula (3) and (4). The test results are shown in Fig. 1- Fig. 3.

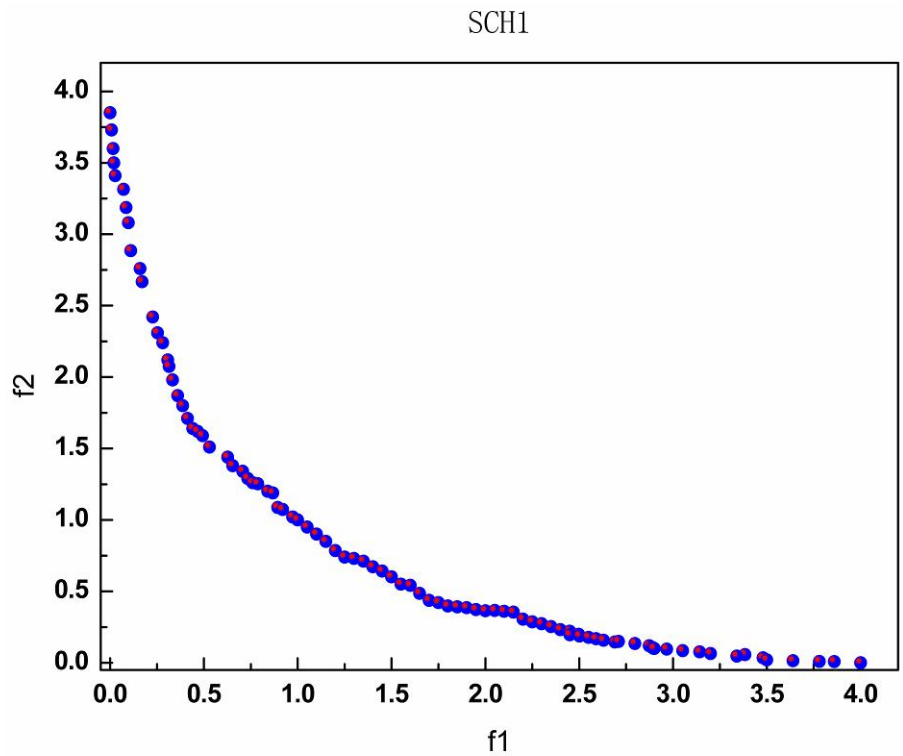

Fig. 1 IPSO solves the SCH1 function Pareto non - inferior solution

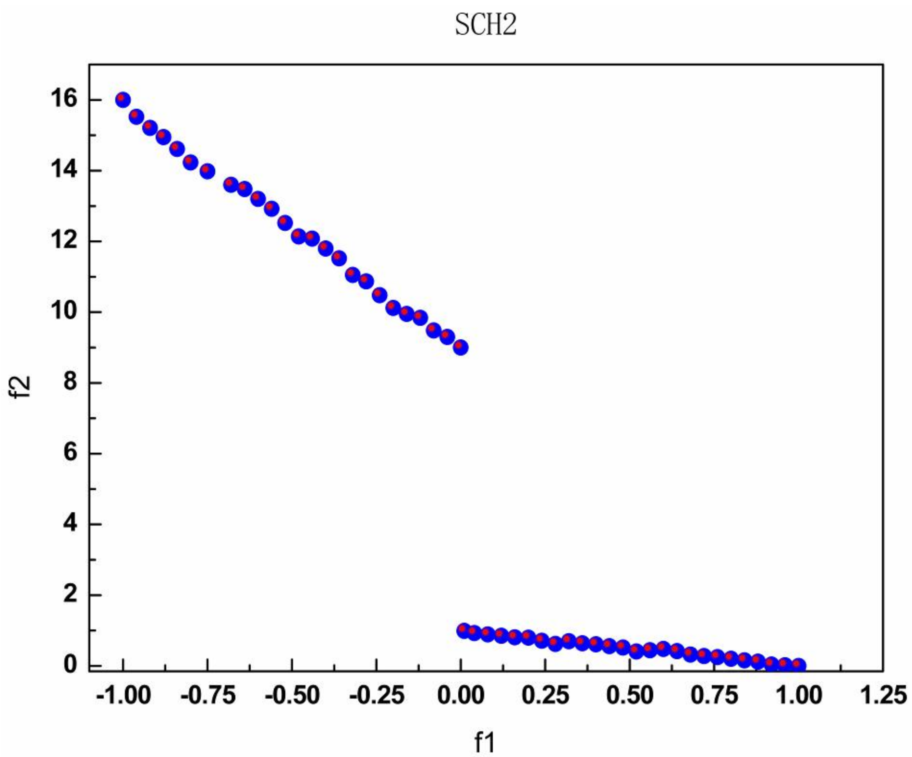

Fig. 2 IPSO solves the SCH2 function Pareto non - inferior solution

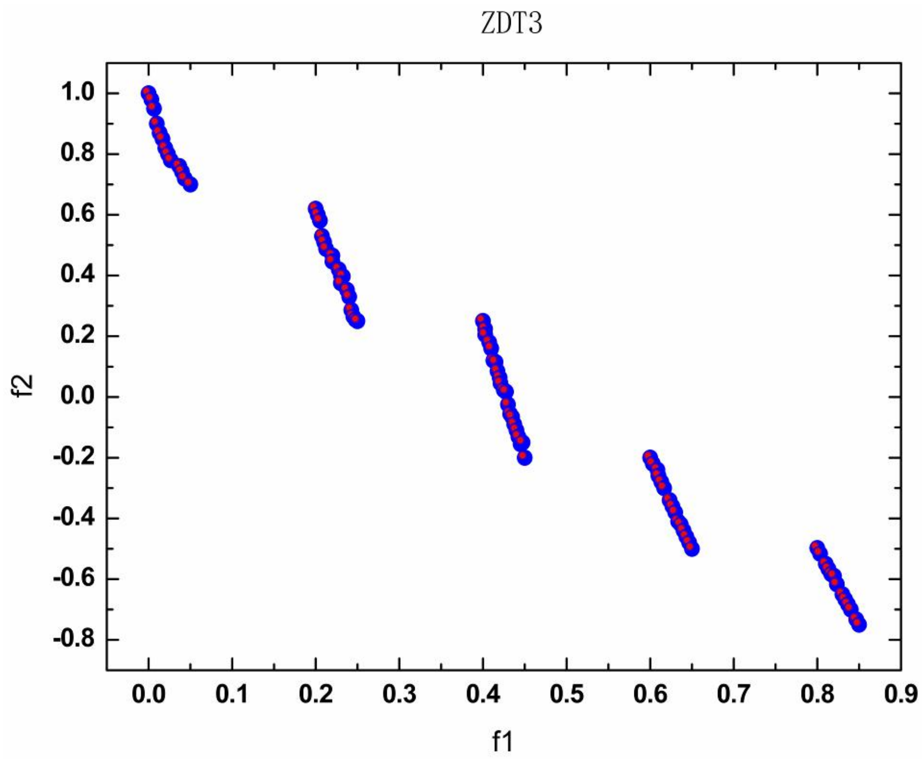

Fig. 3 IPSO solves the ZDT3 function Pareto non - inferior solution

From the experimental results Fig. 1- Fig. 3 , we can see that the three test functions accurately give the effective interface, and the algorithm in this paper obtains complete Pareto curve through the three test function simulation experiments. Especially for the difficult ZDT3 test functions, the target vector distribution is even. Therefore, it has practical reference value for multi-objective optimization problem in engineering.

3.2 Analysis of the results

In order to further evaluate the convergence and distribution uniformity of the non-inferior solutions, this paper adopts convergence index and distribution index to evaluate the performance of the algorithm, which is defined as follows[26,27]:

（1）Convergence index (Gonvergence Distance, GD), GD is used to describe the distance between the non-dominated solutions that the algorithm finds and the optimal front end of the real Pareto algorithm.

$$GD = \frac{\sqrt{\sum_{i=1}^{N} d_i^2}}{N} \tag{5}$$

Where, N is the number of non-dominated solutions that the algorithm finds, d_i^2 represents the shortest Euclidean distance between non-inferior solution i and all solutions in the optimal front end of the real Pareto.

（2）Distribution index SP, is to evaluate the uniformity of distribution of non-dominated solutions.

$$SP = \frac{\left[\frac{1}{n}\sum_{i=1}^{n}(\bar{d}-d_i)^2\right]^2}{\bar{d}} \tag{6}$$

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i \tag{7}$$

Where, N is the number of non-dominated solutions, d_i represents the shortest distance between the ith non-inferior solution in the target space and all solutions in the optimal front end of the real Pareto.

The algorithm IPSO proposed in this paper runs 30 times per test function. The mean of convergence index GD , distribution index SP and computed time CT of each test function is calculated separately. And the average of GD, SP and CT of 3 standard test functions is calculated. The statistical results are shown in Table 1.

performance index	SCH1	SCH2	ZDT3	average value
GD	0.000344	0.000336	0.000327	0.000336
SP	0.0034	0.0037	0.0031	0.0034
CT	2.023	2.134	2.213	2.123

Table 1 IPSO optimizes test function performance statistics

The evaluation index GD, SP, CT confirmed the accuracy of the IPSO algorithm to solve the multi-objective non-inferior solution. GD indicates 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 elapsed time is within the allowable range.

The IPSO algorithm of this paper is compared with classic non-poor classification multi-objective no domination sorting genetic algorithms(NSGA-Ⅱ), the convergence index GD of three test function solved by multi-objective particle swarm optimization algorithms

(MOPSO), the mean value of the distributed index SP and the computed time CT. The results are shown in Table 2.

performance index	NSGA II	MOPSO	IPSO
GD	0.000353	0.000355	0.000336
SP	0.0036	0.0040	0.0034
CT	2.402	0.076	2.123

Table 2 The comparison of optimize test function results of three algorithms

According to the results obtained in Table 2, the IPSO algorithm GD is significantly better than NSGA II and MOPSO GD, and the optimal front-end distance between the non-inferior solution and the real Pareto decrease by 4.8% and 5.4% respectively. SP is to evaluate the distribution of the solution set in the target space by calculating the distance change between each individual and neighbors individual, and the smaller the value, the better the distribution. Table 2 shows that the SP value of IPSO is minimal, indicating that the distribution of non-inferior solutions of IPSO algorithm is more uniform than that of the other two algorithms. For computed time CT, IPSO takes less time than NSGA II in the process of running, but it is more time-consuming than the MOPSO algorithm. The reason is that the improved algorithm is not searched by equal step length in the search process, and the standard PSO algorithm is searched by equal step length and in a single direction. Obviously the IPSO algorithm takes more time than the standard PSO algorithm, but the time spent in IPSO is also within the allowable range. To sum up, through the GD and SP performance compared with other algorithms, the algorithm proposed in this paper is proved to be feasible and effective, is an important method to solve the multi-objective optimization problem.

4. Conclusions

This paper presents a chaotic adaptive Improved Particle Swarm optimization algorithm(IPSO). The algorithm uses chaos theory and adaptive adjustment strategy to optimize the parameters in PSO algorithm, overcame the precocious convergence of PSO algorithm, and improve the convergence speed, so that The dispersion of the solution set is better. The experimental results of three standard test functions show that when the algorithm proposed in this paper is used to solve the multi-objective problem, the obtained non-inferiority solution can approach Pareto optimal solution set and distribute evenly. Comparing the proposed algorithm with other algorithms, the IPSO algorithm has better performance.

ACKNOWLEDGMENTS

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

References

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

[2] ZHU, Q. L., LIN, Q. Z., CHEN, W. N., ET AL. An External Archive-Guided Multiobjective Particle Swarm Optimization Algorithm[J]. IEEE Transactions on Cybernetics, 2017, 47(9): 2794-2808.

[3] WANG, L., YANG, B., ORCHARD, J. Particle swarm optimization using dynamic tournament topology[J]. Applied Soft Computing, 2016, 48: 584-596.

[4] LIU, J. H., MEI, Y., LI, X. D. An Analysis of the Inertia Weight Parameter for Binary Particle Swarm Optimization[J]. Ieee Transactions On Evolutionary Computation, 2016, 20(5): 666-681.

[5] ZHENG, L. M., WANG, Q., ZHANG, S. X., ET AL. Population recombination strategies for multi-objective particle swarm optimization[J]. Soft Computing, 2017, 21(16): 4693-4705.

[6] KAMPOLIS, I. C., GIANNAKOGLOU, K. C. A multilevel approach to Single- and multiobjective aerodynamic optimization[J]. Computer Methods in Applied Mechanics and Engineering, 2008, 197(33-40): 2963-2975.

[7] YAN, J., HE, W. X., JIANG, X. L., ET AL. A novel phase performance evaluation method for particle swarm optimization algorithms using velocity-based state estimation[J]. Applied Soft Computing, 2017, 57: 517-525.

[8] QIN, Q. D., CHENG, S., ZHANG, Q. Y., ET AL. Particle Swarm Optimization With Interswarm Interactive Learning Strategy[J]. IEEE Transactions on Cybernetics, 2016, 46(10): 2238-2251.

[9] MOHIUDDIN, M. A., KHAN, S. A., ENGELBRECHT, A. P. Fuzzy particle swarm optimization algorithms for the open shortest path first weight setting problem[J]. Applied Intelligence, 2016, 45(3): 598-621.

[10] LI, L. S., LAI, K. K. A fuzzy approach to the multiobjective transportation problem[J]. Computers & Operations Research, 2000, 27(1): 43-57.

[11] GUNZBURGER, M. D., LEE, J. A domain decomposition method for optimization problems for partial differential equations[J]. Computers & Mathematics with Applications, 2000, 40(2-3): 177-192.

[12] ZITZLER, E., DEB, K., THIELE, L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results[J]. Evolutionary Computation, 2000, 8(2): 173-195.

[13] DEB, KALYANMOY. Multi-objective genetic algorithms: Problem difficulties and construction of test problems[J]. Evolutionary Computation, 1999, 7(3): 205-230.

[14] LAUMANNS, M., THIELE, L., DEB, K., ET AL. Combining convergence and diversity in evolutionary multiobjective optimization[J]. Evolutionary Computation, 2002, 10(3): 263-282.

[15] JIANG, F., XIA, H. Y., TRAN, Q. A., ET AL. A new binary hybrid particle swarm optimization with wavelet mutation[J]. Knowledge-based Systems, 2017, 130: 90-101.

[16] JAVIDRAD, F., NAZARI, M. A new hybrid particle swarm and simulated annealing stochastic optimization method[J]. Applied Soft Computing, 2017, 60: 634-654.

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

[18] FONSECA, CARLOS M, FLEMING, PETER J. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization [C]. // Proceedings of the Fifth International Conference on Genetic Algorithms. San Mateo, California, 1993.416-423.

[19] REY HORN, JE, NAFPLIOTIS, NICHOLAS, GOLDBERG, DAVID E. Multiobjective optimization using the niched pareto genetic algorithm[J]. IlliGAL report, 1993, (93005): 61801-62296.

[20] SRINIVAS, NIDAMARTHI, DEB, KALYANMOY. Muiltiobjective optimization using nondominated sorting in genetic algorithms[J]. Evolutionary Computation, 1994, 2(3): 221-248.

[21] Zitzler, E., Thiele, L. Multiobjective optimization using evolutionary algorithms - A comparative case study. In: Eiben AE， Back T， Schoenauer M,Schwefel HP, (eds.). Parallel Problem Solving from Nature - Ppsn V1998, p. 292-301.

[22] DEB, K., PRATAP, A., AGARWAL, S., ET AL. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. Ieee Transactions On Evolutionary Computation, 2002, 6(2): 182-197.

[23] YUHUI, SHI, EBERHART, R. C. Fuzzy adaptive particle swarm optimization[J]. Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), 2001: 101-106 vol. 101.

[24] ZHOU, SONG-HUA, OUYANG, CHUN-JUAN, LIU, CHANG-XIN, ET AL. Adaptive fuzzy particle swarm optimization algorithm[J]. Computer Engineering and Applications, 2010, 46(33): 46-48.

[25] HE, S., WU, Q. H., WEN, J. Y., ET AL. A particle swarm optimizer with passive congregation[J]. Biosystems, 2004, 78(1-3): 135-147.

[26] TAN, K. C., LEE, T. H., KHOR, E. F. Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons[J]. Artificial Intelligence Review, 2002, 17(4): 253-290.

[27] ZITZLER, E., THIELE, L., LAUMANNS, M., ET AL. Performance assessment of multiobjective optimizers: An analysis and review[J]. Ieee Transactions On Evolutionary Computation, 2003, 7(2): 117-132.

[28] 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[J]. Neural Computing & Applications, 2017, 28(8): 2099-2104.

[29] MAC, T. T., COPOT, C., TRAN, D. T., ET AL. A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization[J]. Applied Soft Computing, 2017, 59: 68-76.

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

[31] GONG, Y. J., LI, J. J., ZHOU, Y. C., ET AL. Genetic Learning Particle Swarm Optimization[J]. IEEE Transactions on Cybernetics, 2016, 46(10): 2277-2290.

[32] KENNEDY, J., EBERHART, R. Particle swarm optimization[J]. 1995 IEEE International Conference on Neural Networks Proceedings (Cat. No.95CH35828), 1995: 1942-1948 vol.1944.

[33] KENNEDY, J., EBERHART, R. Particle swarm optimization [C]. // 1995 IEEE International Conference on Neural Networks Proceedings, Proceedings of ICNN'95 - International Conference on Neural Networks. Perth: IEEE Australia Council,

1995.1942-1948.
[34] SCHAFFER, J DAVID. Multiple objective optimization with vector evaluated genetic algorithms [C]. // Proceedings of the 1st international Conference on Genetic Algorithms. L. Erlbaum Associates Inc., 1985.93-100.