Scientific Research Publishing

Journal of Computer and Communications, 2017, 5, 9-20
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

# Dynamic Self-Adaptive Double Population Particle Swarm Optimization Algorithm Based on Lorenz Equation

**Yan Wu, Genqin Sun, Keming Su, Liang Liu, Huaijin Zhang, Bingsheng Chen, Mengshan Li**

Gannan Normal University, Ganzhou, China
Email: jcimsli@163.com

**How to cite this paper:** Wu, Y., Sun, G.Q., Su, K.M., Liu, L., Zhang, H.J., Chen, B.S. and Li, M.S. (2017) Dynamic Self-Adaptive Double Population Particle Swarm Optimization Algorithm Based on Lorenz Equation. *Journal of Computer and Communications*, **5**, 9-20.
https://doi.org/10.4236/jcc.2017.513002

**Received:** October 11, 2017
**Accepted:** November 7, 2017
**Published:** November 10, 2017

## Abstract

In order to improve some shortcomings of the standard particle swarm optimization algorithm, such as premature convergence and slow local search speed, a double population particle swarm optimization algorithm based on Lorenz equation and dynamic self-adaptive strategy is proposed. Chaotic sequences produced by Lorenz equation are used to tune the acceleration coefficients for the balance between exploration and exploitation, the dynamic self-adaptive inertia weight factor is used to accelerate the converging speed, and the double population purposes to enhance convergence accuracy. The experiment was carried out with four multi-objective test functions compared with two classical multi-objective algorithms, non-dominated sorting genetic algorithm and multi-objective particle swarm optimization algorithm. The results show that the proposed algorithm has excellent performance with faster convergence rate and strong ability to jump out of local optimum, could use to solve many optimization problems.

## Keywords

Improved Particle Swarm Optimization Algorithm, Double Populations, Multi-Objective, Adaptive Strategy, Chaotic Sequence

## 1. Introduction

Particle swarm optimization (PSO) algorithm, which is an intelligent optimization algorithm, has many characteristics such as simple structure, less parameter, easy description and realization, strong global search ability and no need for gradient information, widely used in many fields, such as function optimization, multi-objective problem solving, pattern recognition, etc., especially applicable to solving the problem of nonlinear, multi-extreme value and non-differentiable and multivariable complexity optimization [1] [2] [3] [4]. However, similar to other intelligent algorithms, the standard PSO algorithm also has shortcomings such as precocious convergence and local search capability [5] [6]. For example, when it is used in the high dimensional complex optimization problems, population may when no search to the global optimal point form the premature convergence problem, then can't guarantee the algorithm can converge to the global extreme value point. At the same time, in the search process, when the particle is approaching or entering the most advantageous area, the convergence speed is obviously slow.

For the lack of PSO algorithm, the researchers propose many improvement strategies [7] [8] [9]. At present, the improvement of PSO algorithm mainly focuses on the adjustment of the algorithm parameters and the updating of the particle structure and trajectory [10] [11] [12], aiming to make the algorithm solve or improve the problem of local search slow, precocious convergence, and improve the convergence speed and precision of the algorithm to improve the performance of the algorithm [13] [14]. Among them, the introduction of inertia weight factor [15], shrinkage factor and adaptive mutation operator is the most representative, such as a linear gradient method [16], the fuzzy adaptive method [17] [18] and distance information such as inertia coefficient adaptive control method [19], the compression factor of PSO algorithm, PSO algorithm of adaptive mutation operator, etc. In addition, standard PSO algorithm and the hybrid PSO algorithm combined with collaborative strategy [20], chaos theory [21] and other algorithms are also attracted by researchers [22] [23], such as the quantum PSO algorithm with chaotic mutation operator [24]. Besides, There are also many researches on discrete PSO algorithm, multi-objective PSO algorithm and so on [25] [26].

Although the improved PSO algorithm improves both performance and efficiency, it is difficult to improve the local search ability of the algorithm while avoiding precocious convergence. There are many academics and industry researchers have been exploring and experimenting with new approaches to provide better performance, higher efficiency and lower cost particle swarm optimization [27] [28]. Therefore, in this paper, a self-adaptive double population PSO algorithm based on Lorenz equation was proposed, called LSA-DPSO. In the LSA-DPSO algorithm, Lorenz equation and the dynamic adaptive weight adjustment strategy were introduced. In order to verify the feasibility and effectiveness of the proposed algorithm, the convergence speed and accuracy are discussed compared with standard PSO algorithm and classical multi-objective algorithm.

## 2. LSA-DPSO Algorithm

### 2.1. Standard PSO

PSO is an evolution algorithm proposed in 1995 by scholars Eberhart and Kennedy [29]. 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 are represented by two global variables to represent the best position of the particle itself (pbest) and the best position of all particles (gbest), Let's say in a n dimensional search space, The population of m particles is $X = \left(x_{1,} x_{2,} \cdots, x_n\right)^{\mathrm{T}}$, The position of the *i*th particle is expressed $x_i = \left(x_{i,1,} x_{i,2,} \cdots, x_{i,n}\right)^{\mathrm{T}}$, The velocity is expressed as $v_i = \left(v_{i,1,} v_{i,2,} \cdots, v_{i,n}\right)^{\mathrm{T}}$. Its individual extremum is represented as $p_i = \left(p_{i,1,} p_{i,2,} \cdots, p_{i,n}\right)^{\mathrm{T}}$ ,The global extremum of the particle population is $p_g = \left(p_{g,1,} p_{g,2,} \cdots, p_{g,n}\right)^{\mathrm{T}}$, during the k + 1 iteration, the particle updates its speed and position through formula (1) and (2).

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

$$x_{i,d}^{k+1} = x_{i,d}^{k} + v_{i,d}^{k+1} \tag{2}$$

where $i = 1, \cdots M$; $\omega$ is called an inertial weight factor, which keeps the particles in motion and has the ability to expand the search space. $c_1$ and $c_2$ are the learning factors, which represent the weight of each particle to the extreme position statistical acceleration item. $v_{i,d}^{k}$, $x_{i,d}^{k}$ are respectively the velocity and position of the d dimension in the $k_{th}$ iteration of the particle *i*; $p_{i,d}^{k}$ is the position of the individual extremum of particle *i* in d dimension, $p_{g,d}^{k}$ is the position of the population in the global extreme value of d dimension.

## 2.2. LSA-DPSO

There are three factors that determine the change in particle velocity in PSO algorithm. 1) Inertial weight factor, instruct the speed information of the previous moment which compares current speed with front speed. 2) Cognitive factors, which is the development capacity factor, not only the optimal error of the particle itself, but also the partial excavation and development capability of the particle. 3) Exploration factor, the social sharing ability coefficient, expresses its error that distances the best and how the particle share and cooperate. Under the circumstances, the inertia coefficient determines the search step length, which is better for global search and smaller for local exploration. Cognitive factors and exploration factors are collectively called learning factors, which represent the optimal and global optimal proportion of particles themselves. By adjusting the learning factors properly, the global and local search steps of the particles can be weighed and the exploration factors can be changed to realize the local optimal algorithm, when the algorithm is in precocious convergence.

Based on the above, the authors developed LSA-DPSO algorithm by mixing She mixed Lorenz equation and dynamic adaptive strategy into a dual population PSO algorithm. LSA-DPSO using adaptive evolutionary inertia weight adjustment strategy, improve the convergence speed, chaotic sequence produced by Lorenz equation optimizing algorithm of learning factor $c_1$ and $c_2$, jump out of local optimum when entering the premature convergence and using a dual population to improve the diversity of it. The inertial weight factor is dynamically adjusted by formula (3).

$$\omega = \omega_{\max} - Pgbest(k) / Plbest_{ave} - (\omega_{\max} - \omega_{\min}) \times k / k_{\max} \tag{3}$$

Among them, omega max and omega min respectively indicate the maximum and minimum value of the inertial weight; *Pgbest*(*k*) represents the global optimum of the *k-th* iteration; $Plbest_{ave}$ represents the local optimal average of all particles; $k_{\max}$ represents the maximum number of iterations and *k* is the current iteration number.

The learning factor $c_1$ and $c_2$ produce chaotic sequences through the typical Lorenz equation, as shown in formula (4).

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} = -a(x-y) \\ \frac{\mathrm{d}y}{\mathrm{d}t} = rx - y - xz \\ \frac{\mathrm{d}z}{\mathrm{d}t} = xy - bz \end{cases} \tag{4}$$

In the formula, the parameters *a*, *b* and *r* are the positive control parameters, which are respectively 10, 8/3, and 28, and the learning factor ($c_1$, $c_2$) has a chaotic state, which is defined as:

$$\begin{cases} c_1 = x(t) \\ c_2 = y(t) \end{cases} \tag{5}$$

Because of the randomness and regularity, the change of chaotic variables making the algorithm can maintain the diversity of the population, improve the problem of premature convergence and improve the global search performance.

The steps of LSA-DPSO algorithm are as follows:

1) Initialize the particle group

The particles are initialized to two subpopulations *A* and *B*, and the positions and velocities of all the particles of *A* and *B* are initialized, and the initial positions and velocities of the particles are randomly generated. The current position of each particle is used as the maximum value of the particle, and the optimal value of the subgroup *A* and *B* is selected as the global optimal value of subgroup A and B.

2) Calculate the adaptive value of population particles.

3) Compare the adaptive value of each particle with the adaptive value of the best position of its own, and make the better current position as 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 the better current position is the best position.

5) To compares the best position of the global best position in subgroup B with the global best position in subgroup A, and USES A better position as the global extreme value of A group. Then we can compare the good position as the

global extreme value of group B.

6) Obtain the learning factor $c_1$, $c_2$ and inertial weight respectively, and update the velocity and position of the particles.

7) 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 Experiments

### 3.1. Experiment Functions and Evaluation

Multi-objective optimization is the most typical optimization problem. As the goals are often contradictory and constrained, it is difficult to achieve optimal simultaneous optimization, which means that one of the objectives must be optimized at the expense of other goals. The solution to such a problem is usually not unique, but the existence of a series of optimal solutions, known as non-inferior solutions, which are often referred to as Pareto optimal solutions. Because swarm intelligence algorithm can parallel search multiple solutions in the search space, some classical multi-objective optimization problem can verify the performance of the algorithm very well. In this paper, the multi-objective optimization test function proposed by Schaffer [30] and Deb [31] is used as an experimental case, and the test function is shown in **Table 1**.

The advantages and disadvantages of non-inferior solutions are evaluated by means of convergence and distribution indicators:

Convergence index (GD) [32], which is used to describe the distance between the non-dominant 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 ungoverned solutions that the algorithm searches for, $d_i$ Represents the shortest Euclidean distance of all solutions in the optimal front end of the true Pareto and the $i$-th pare to solutions.

Distribution index ($SP$) [33], $SP$ is used to evaluate the uniformity of distribution of ungoverned solution concentrated solutions. Among them, the algorithm parameters are set to:

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

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

Among then, $N$ is the number of ungoverned solutions, $d_i$ Represents the shortest distance between the $i$ th non-inferior solution in the target space and all solutions in the optimal front end of the real Pareto.

**Table 1.** Experimental test function.

| Function | Define | Feature |
|---|---|---|
| SCH1 | $\min f(x)=(f_1(x), f_2(x))\, s.t. x\in[-5,7]$<br>$f_1(x)=x^2$<br>$f_2(x)=(x-2)^2$ | Convex type |
| SCH2 | $\min f(x)=(f_1(x), f_2(x)), s.t. x\in[-5,10]$<br>$f_1(x)=\left\{\begin{matrix} -x,(x\le 1) \\ -2+x,(1<x\le 3) \\ 4-x,(3<x\le 4) \\ -4+x,(x>4) \end{matrix}\right\}$<br>$f_2=(x-5)^2$ | Discontinuous |
| ZDT2 | $\min f(x)=(f_1(x), f_2(x)), s.t. x\in[-5,10]$<br>$f_1(x)=x_1$<br>$f_2(x)=g(x)\left[1-\left(x_1/g(x)^2\right)\right]$<br>$g(x)=1+9\left(\sum_{i=2}^{n}x_i\right)\Big/(n-1)$ | Concave type |
| ZDT3 | $\min f(x)=(f_1(x), f_2(x)), s.t. x_i\in[0,1]$<br>$f_1(x)=x_1$<br>$f_2(x)=g(x)\left[1-\sqrt{x_1/g(x)}-\frac{x_1}{g(x)}\sin(10\pi x_1)\right]$<br>$g(x)=1+9\left(\sum_{i=2}^{n}x_i\right)\Big/(n-1)$ | Discrete type |

## 3.2. Experimental Results

It proposed the LSA-DPSO algorithm to simulate the multi-objective test function SCH1, SCH2, ZDT2 and ZDT3. The population size is 60; the maximum iteration number is set to 500; the maximum and minimum values of inertia weight are 0.9 and 0.3 respectively. Pareto non-inferior solution was obtained by algorithm, and the distribution of Pareto non-inferior solutions of each function was plotted in **Figures 1-4**.

To solve the multi-objective function, the optimal target domain is the boundary of the fitness value region, which is called the effective interface. As can be seen from the Pareto non-inferior solution drawn from the graph, the four test functions accurately give the effective interface, and the complete Pareto non-inferior solution can be obtained. It can be seen from the distribution of solution that the solution distribution of each function is even.

When solving complex multi-objective functions such as discrete and discontinuities, the algorithm also gives an accurate and non-inferior solution set, such as SCH2 and ZDT3.

The above results show that the LSA-DPSO algorithm is superior in solving the multi-objective problem, and obtains a non-inferior solution which is very close to the real Pareto frontier, and has a large number of non-inferior solutions with good distribution, what is verifies the feasibility and accuracy of the algorithm.

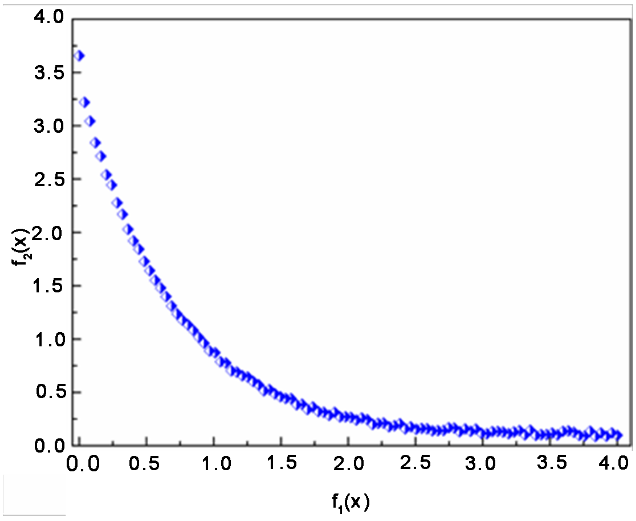

**Figure 1.** Pareto non-inferior solution of SCH1.

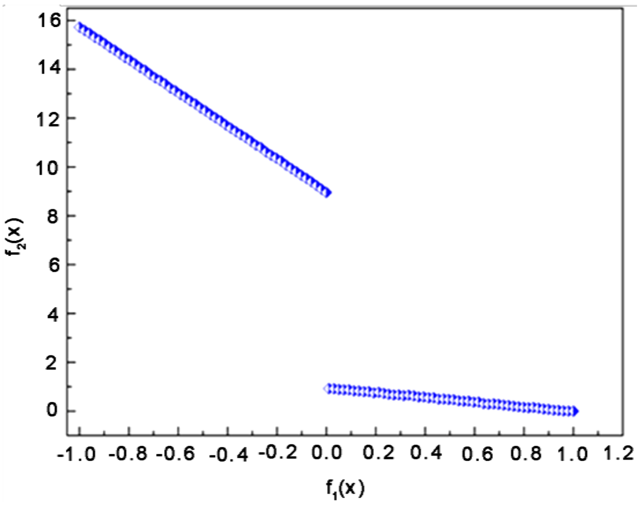

**Figure 2.** Pareto non-inferior solution of SCH2.

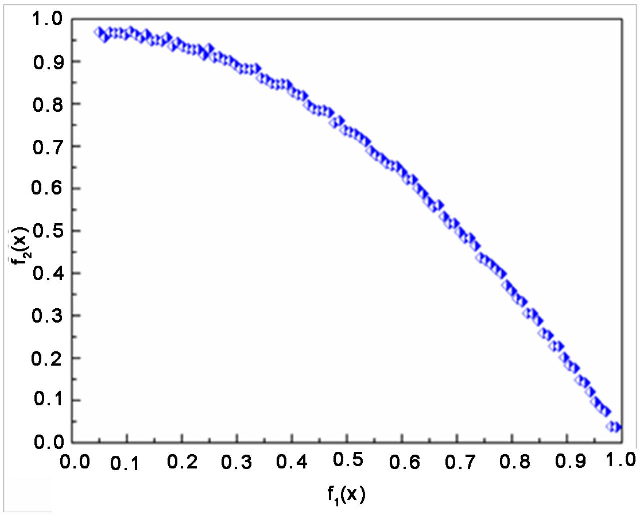

**Figure 3.** Pareto non-inferior solution of ZDT2.

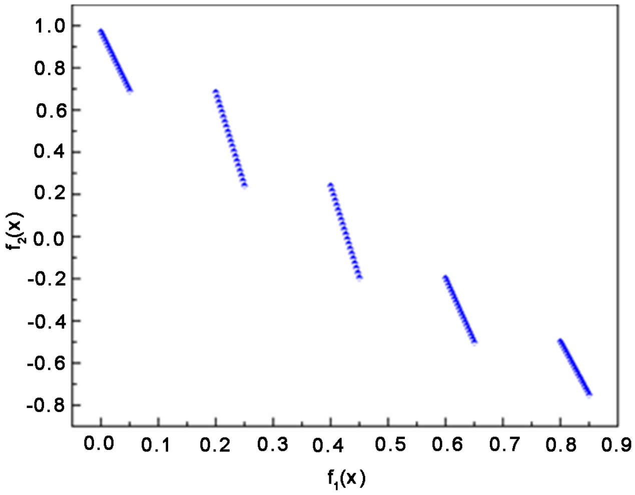

**Figure 4.** Pareto non-inferior solution of ZDT3.

In order to further verify the robustness and stability of the algorithm, each test function with the LSA-DPSO algorithm run 5 times, statistics of each test function index of the convergence of GD, SP distribution index and calculating the average time of CT, and four statistical evaluation index of the test function, and the results are shown in Table 2.

The data show that the LSA-DPSO algorithm performs well in solving multi-objective problems that confirms the feasibility, accuracy and efficiency of LSA-DPSO algorithm. GD indicates that noninferior-solutions are very close to the real Pareto. And through the data of SP, noninferior-solutions are well distributed. Meanwhile, from the differentia of CT index, the difference between functions is within acceptable limits.

To compare the advantages of each algorithm. We contrast LSA-DPSO algorithm with the NSGA-II algorithm and the MOPSO algorithm by decomposing each algorithm five times using various target test functions. The statistical results are shown in Table 3.

In terms of convergence accuracy and distribution, the LSA-DPSO algorithm is superior. At the point of GD and SP, The optimal front-end distance that is obtained through the LSA-DPSO algorithm is closer than other two algorithms, what indicates LSA-DPSO has better convergence and accuracy. At the point of CT, the execution time of LSA-DPSO algorithm is lower than NSGA II algorithm but higher than that of MOPSO. The reason is that the LSA-DPSO algorithm is dynamically adjusted with the flight process of the particle, which will surely consume more time than the standard PSO algorithm, which is long and single-directional flight search. As an inevitable result, the convergence and distribution of the non-inferior solutions obtained by the LSA-DPSO algorithm are better than those of the other two algorithms.

Through the LSA-DPSO algorithm for four multi-objective optimization function of numerical experiments show that the LSA-DPSO algorithm has a better comprehensive performance, introducing Lorenz equation of chaotic

Table 2. Performance statistics of LSA-DPSO optimization test functions.

| Performance indicators | SCH1 | SCH2 | ZDT2 | ZDT3 | Average |
|---|---|---|---|---|---|
| GD | 0.000335 | 0.000336 | 0.000353 | 0.000340 | 0.000341 |
| SP | 0.00338 | 0.00335 | 0.00336 | 0.00325 | 0.00334 |
| CT[a] | 17.7 | 17.6 | 18.4 | 19.8 | 18.4 |

[a]Computation time.

Table 3. The comparison results of three algorithms are optimized

| Performance indicators | NSGA II[a] | MOPSO[b] | LSA-DPSO |
|---|---|---|---|
| GD | 0.000366 | 0.000465 | 0.000341 |
| SP | 0.00465 | 0.00521 | 0.00334 |
| CT | 22.5 | 17.5 | 18.4 |

[a]non-inferior classification multi-objective genetic algorithm, [b]multi-target particle swarm optimization algorithm.

sequence to improve the premature convergence problem of PSO algorithm, through dynamic adaptive mechanism of multiple species and improve the algorithm convergence speed and accuracy.

## 4. Conclusion

This paper presents an algorithm that is based on the dynamic adaptive dual population particle swarm of Lorenz equation. By using the chaos sequence generated by Lorenz equation, the learning factor is optimized and the dynamic adaptive adjustment strategy is adjusted to the inertia weight, it improved the convergence speed, accuracy and the problem that the convergence of PSO algorithm is premature. Through the test function experiment, the proposed algorithm is used to solve the multi-target problem, and the obtained non-inferiority solution can approach the optimal solution set of Pareto very well, and the distribution is even. It indicates the performance of LSA-DPSO algorithm is better. In this paper, the algorithm is used to solve the problem of complex optimization problems such as non-linear, multipolar and multivariable, which can provide references for many practical engineering applications.

## Acknowledgements

The authors gratefully acknowledge the support from the National Natural Science Foundation of China (Grant Numbers: 51663001, 51463015, 51377025) and the science and technology research project of the education department of Jiangxi province (Grant Numbers: GJJ150983, GJJ151012).

## References

[1] Wang, L., Yang, B. and Chen, Y.H. (2014) Improving Particle Swarm Optimization Using Multi-Layer Searching Strategy. *Information Sciences*, **274**, 70-94.

https://doi.org/10.1016/j.ins.2014.02.143

[2] Neri, F., Mininno, E. and Lacca, G. (2013) Compact Particle Swarm Optimization. *Information Sciences*, **239**, 96-121. https://doi.org/10.1016/j.ins.2013.03.026

[3] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M. and da Fonseca, V.G. (2003) Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *Ieee Transactions On Evolutionary Computation*, **7**, 117-132. https://doi.org/10.1109/TEVC.2003.810758

[4] Zhao, X.L., Turk, M., Li, W., Lien, K.C. and Wang, G.Z. (2016) A Multilevel Image Thresholding Segmentation Algorithm Based on Two-Dimensional K-L Divergence and Modified Particle Swarm Optimization. *Applied Soft Computing*, **48**, 151-159. https://doi.org/10.1016/j.asoc.2016.07.016

[5] Tsekouras, G.E. and Tsimikas, J. (2013) On Training RBF Neural Networks Using Input-Output Fuzzy Clustering and Particle Swarm Optimization. *Fuzzy Sets and Systems*, **221**, 65-89. https://doi.org/10.1016/j.fss.2012.10.004

[6] del Valle, Y., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J.C. and Harley, R.G. (2008) Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. *Ieee Transactions On Evolutionary Computation*, **12**, 171-195. https://doi.org/10.1109/TEVC.2007.896686

[7] Akay, B. (2013) A Study on Particle Swarm Optimization and Artificial Bee Colony Algorithms for Multilevel Thresholding. *Applied Soft Computing*, **13**, 3066-3091. https://doi.org/10.1016/j.asoc.2012.03.072

[8] He, G. and Huang, N.J. (2012) A Modified Particle Swarm Optimization Algorithm with Applications. *Applied Mathematics and Computation*, **219**, 1053-1060. https://doi.org/10.1016/j.amc.2012.07.010

[9] Taghiyeh, S. and Xu, J. (2016) A New Particle Swarm Optimization Algorithm for Noisy Optimization Problems. *Swarm Intelligence*, **10**, 161-192. https://doi.org/10.1007/s11721-016-0125-2

[10] Lynn, N. and Suganthan, P.N. (2015) Heterogeneous Comprehensive Learning Particle Swarm Optimization with Enhanced Exploration and Exploitation. *Swarm and Evolutionary Computation*, **24**, 11-24. https://doi.org/10.1016/j.swevo.2015.05.002

[11] Jiang, S.J., Shi, J.J., Zhang, Y.M. and Han, H. (2015) Automatic Test Data Generation Based on Reduced Adaptive Particle Swarm Optimization Algorithm. *Neurocomputing*, **158**, 109-116.

[12] Esmin, A.A.A., Coelho, R.A. and Matwin, S. (2015) A Review on Particle Swarm Optimization Algorithm and Its Variants to Clustering High-Dimensional Data. *Artificial Intelligence Review*, **44**, 23-45. https://doi.org/10.1007/s10462-013-9400-4

[13] Gandomi, A.H., Yun, G.J., Yang, X.S. and Talatahari, S. (2013) Chaos-Enhanced Accelerated Particle Swarm Optimization. *Communications in Nonlinear Science and Numerical Simulation*, **18**, 327-340.

[14] Nickabadi, A., Ebadzadeh, M.M. and Safabakhsh, R. (2011) A Novel Particle Swarm Optimization Algorithm with Adaptive Inertia Weight. *Applied Soft Computing*, **11**, 3658-3670.

[15] Zhan, Z.H., Zhang, J., Li, Y. and Chung, H.S.H. (2009) Adaptive Particle Swarm Optimization. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, **39**, 1362-1381. https://doi.org/10.1109/TSMCB.2009.2015956

[16] Shang, L., Zhou, Z. and Liu, X. (2016) Particle Swarm Optimization-Based Feature Selection in Sentiment Classification. *Soft Computing*, **20**, 3821-3834. https://doi.org/10.1007/s00500-016-2093-2

[17] Valdez, F., Melin, P. and Castillo, O. (2011) An Improved Evolutionary Method with Fuzzy Logic for Combining Particle Swarm Optimization and Genetic Algorithms. *Applied Soft Computing*, **11**, 2625-2632.

[18] Thong, P.H. and Son, L.H. (2016) A Novel Automatic Picture Fuzzy Clustering Method Based on Particle Swarm Optimization and Picture Composite Cardinality. *Knowledge-Based Systems*, **109**, 48-60.

[19] Marinakis, Y. and Marinaki, M. (2013) Particle Swarm Optimization with Expanding Neighborhood Topology for the Permutation Flowshop Scheduling Problem. *Soft Computing*, **17**, 1159-1173. https://doi.org/10.1007/s00500-013-0992-z

[20] Sinha, A.K., Zhang, W.J. and Tiwari, M.K. (2012) Co-Evolutionary Immuno-Particle Swarm Optimization with Penetrated Hyper-Mutation for Distributed Inventory Replenishment. *Engineering Applications of Artificial Intelligence*, **25**, 1628-1643.

[21] Cheng, M.Y., Huang, K.Y. and Chen, H.M. (2012) K-Means Particle Swarm Optimization with Embedded Chaotic Search for Solving Multidimensional Problems. *Applied Mathematics and Computation*, **219**, 3091-3099.

[22] Liu, J.H., Mei, Y. and Li, X.D. (2016) An Analysis of the Inertia Weight Parameter for Binary Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, **20**, 666-681. https://doi.org/10.1109/TEVC.2015.2503422

[23] Harrison, K.R., Engelbrecht, A.P. and Ombuki-Berman, B.M. (2016) Inertia Weight Control Strategies for Particle Swarm Optimization. *Swarm Intelligence*, **10**, 267-305. https://doi.org/10.1007/s11721-016-0128-z

[24] Mohiuddin, M.A., Khan, S.A. and Engelbrecht, A.P. (2016) Fuzzy Particle Swarm Optimization Algorithms for the Open Shortest Path First Weight Setting Problem. *Applied Intelligence*, **45**, 598-621. https://doi.org/10.1007/s10489-016-0776-0

[25] Zheng, Y.J. and Chen, S.Y. (2013) Cooperative Particle Swarm Optimization for Multiobjective Transportation Planning. *Applied Intelligence*, **39**, 202-216. https://doi.org/10.1007/s10489-012-0405-5

[26] Qasem, S.N., Shamsuddin, S.M., Hashim, S.Z.M., Darus, M. and Al-Shammari, E. (2013) Memetic Multiobjective Particle Swarm Optimization-Based Radial Basis Function Network for Classification Problems. *Information Sciences*, **239**, 165-190.

[27] Civicioglu, P. (2013) Backtracking Search Optimization Algorithm for Numerical Optimization Problems. *Applied Mathematics and Computation*, **219**, 8121-8144.

[28] Tsekouras, G.E. (2013) A Simple and Effective Algorithm for Implementing Particle Swarm Optimization in RBF Network's Design using Input-Output Fuzzy Clustering. *Neurocomputing*, **108**, 36-44.

[29] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. *IEEE International Conference on Neural Networks Proceedings*, *Proceedings of International Conference on Neural Networks*, IEEE Australia Council, Perth, 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

[30] Schaffer, J.D. (1985) Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. *International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, 93-100.

[31] Deb, K. (1999) Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation*, **7**, 205-230. https://doi.org/10.1162/evco.1999.7.3.205

[32] Cheng, S.X., Zhan, H. and Shu, Z.X. (2016) An Innovative Hybrid Multi-Objective

Particle Swarm Optimization with or without Constraints Handling. *Applied Soft Computing*, **47**, 370-388.

[33] Qiu, C.Y., Wang, C.L. and Zuo, X.Q. (2013) A Novel Multi-Objective Particle Swarm Optimization with K-Means Based Global Best Selection Strategy. *International Journal of Computational Intelligence Systems*, **6**, 822-835. https://doi.org/10.1080/18756891.2013.805584