2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

A New Dynamic Probabilistic Particle Swarm Optimization with Dynamic Random Population Topology

Qingjian Ni* † ‡, Cen Cao* and Xushan Yin‡
* School of Computer Science and Engineering, Southeast University, Nanjing, China
† Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, China
‡ College of Software Engineering, Southeast University, Nanjing, China
Email: nqj@seu.edu.cn

***Abstract*—Population topologies of Particle Swarm Optimization algorithm (PSO) have direct impacts on the information sharing amony particles during the evolution, and will influence the PSO algorithms' performance obviously. The canonical PSO algorithms usually use static population topologies, and the majority are the classic population topologies (such as fully connected topology and ring topology). In this paper, we present the strategies of dynamic random topology based on the random generation of population topologies. The basic idea is as follows: various random topologies are used at different stages of evolution in the population, and the solving performance of PSO algorithms is enhanced by improving the information exchange of population in different evolutionary stages. This provides a new way of thinking for the improvement of the PSO algorithm. Experimental results on a relatively new variant of dynamic probabilistic particle swarm optimization show that our strategies can achieve better performance compared with traditional static population topologies. Experimental data are analyzed and discussed in the paper, and the useful conclusions will provide a basis for further research.**

I. Introduction

As a bio-simulated evolutionary algorithm, Particle Swarm Optimization (briefed as PSO) is rooted in the imitation of the behavioral mechanisms such as fish and bird flocks. Currently, PSO algorithm has been widely applied in practical engineering fields [1][2][3].

In the evolution process of PSO, particles shares their individual experiences with others. And this kind of information sharing is achieved through the population topology. Particles update their velocity and position by referring to the shared experience among particles. Therefore, population topology decides the experience sharing form between the particles, and has significant impact on the optimization performance.

For the population topologies of PSO algorithms, the fully connected topology (Gbest model) and the ring topology (Lbest model) are the usual static population topologies which are the earliest to be proposed and widely used [4]. After that, different population topologies have been proposed. Kennedy analyzed four static population topologies [5]. Mendes and Stutzle analyzed the effect of several population topologies on PSO algorithms' performance [6][7]. Furthermore, Clerc firstly proposed the random topologies [8], and Ni et al. detailedly studied the random topologies and introduced the random topologies to a new variant of PSO [9].

However, the researchers usually focus on the static population topologies, the research on dynamic population topologies are relatively few. Population topologies have direct impacts on information communication among particles. So, in different periods of evolution, the inner information sharing mechanisms should have different focus. Therefore, it is necessary to have in-depth discussions on dynamic population topologies.

In this paper, based on the random topologies [9], we proposed several possible strategies of dynamic population topology. And the proposed dynamic random population topologies are applied to a relatively new PSO variant of dynamic probabilistic particle swarm optimization, the effectiveness of the proposed strategies are analyzed in depth. The rest of paper is organized as follows. Section II describes the PSO variants which will be used in the latter experiment. In section III, the strategies of dynamic random population topologies are proposed and analyzed. Section IV discusses and analysis the effectiveness of the proposed strategies through adequate experiments. Section V make a summary of the paper.

II. Variants of Particle Swarm Optimization

A. The Particle Swarm Optimization Algorithm with Inertia Weight

As a population based method, particles with velocities and positions evolve through various operations. Based on the earlier variant of PSO, Shi proposed the PSO with inertia weight [10]. The PSO with inertia weight and its variants are widely used in practical engineering applications, and the particles' position and velocity update is according to the equation 1 and 2.

$$\begin{aligned} v_{id} = w * v_{id} + c_1 * rand() * (p_{id} - x_{id}) \\ +c_2 * Rand() * (p_{gd} - x_{id}) \end{aligned} \tag{1}$$

$$x_{id} = x_{id} + v_{id} \tag{2}$$

The symbols of equation 1 and 2 are described in Table I. The 3 parts on the right hand of equation 1 usually can be considered as self-memory, self-cognition and social-cognition of a particle. And the particles' velocities and positions evolve through the interactions of the 3 parts.

B. The Dynamic Probabilistic Particle Swarm Optimization

A usual PSO algorithm has two attributes, velocity and position which is similar to the PSO with inertia weight. Kennedy first proposed a new variant of PSO without the velocity attribute , which is known as Gaussian dynamic particle swarm optimization [11]. Ni conducted further research on this variant of PSO [12] [9]. This variant of PSO usually can be defined as Dynamic Probabilistic Particle Swarm Optimization (DPPSO). Unlike the usual PSO algorithms, particles have no velocities in the DPPSO algorithms, and the particles' position update are according to equation 3.

$$\begin{aligned} X_i(t+1) &= X_i(t) + \alpha * (X_i(t) - X_i(t-1)) \\ &+ \beta * CT_i(t) + \gamma * Gen() * OT_i(t) \end{aligned} \tag{3}$$

$$CT_{id}(t) = \sum_{k=1}^{K} P_{kd}/K - X_{id}(t) \tag{4}$$

$$OT_{id}(t) = \sum_{k=1}^{K} |P_{id} - P_{kd}|/K \tag{5}$$

The symbols of equation 3, 4 and 5 are described in Table II. In the equation 3, 4 and 5 , $CT_i(t)$ and $OT_i(t)$ are calculated by the particle's current position and the optimal positions of its neighborhood particles. $Gen()$ is a random number generator which should satisfy a specific distribution such as a Logistic distribution or a Cauchy distribution, etc.

According to the position update equation 3, the position of a particle's new generation is decided by four factors. The first factor is its memory to its own position. The second factor represents its trend from the former movement direction. The third factor, represents the particle's neighbors affect to the new generation. The forth factor means the impact of the difference between particle and its neighbors' optimal positions on the next generation.

DPPSO has different performance when adopting different dynamic probabilistic evolutionary operator $Gen()$. As can be seen from equation 4 and 5, the calculation of the two important values ($CT_i(t)$ and $OT_i(t)$) would use each neighborhood particle's experience, so the optimal information of the whole population could be utilized. As a result, the research on population topology of DPPSO is particularly important.

III. Dynamic Random Population Topology Strategy

A. The Classical Static Population Topologies

In the current applications of PSO, the most popular population topologies are the fully connection topology and the ring topology, which are also known as Gbest model and Lbest model. Figure 1 shows the schematic connections of the above two population topologies.

For a particle in the fully connected topology, a particle's neighborhood includes all the other particles in the whole swarm. And during the evolution, only the information of the best position which is obtained by the whole swarm will be rapidly spread. When adapting this topology, the convergence speed is very fast, but particles are easy to be trapped into local optima. As for the ring topology, one or several immediate neighborhood could be accessed by a particle. And when adopting this topology, the communication speed in the swarm is relatively slow. But once one of them found an optimum, eventually the information would be spread throughout the whole swarm slowly.

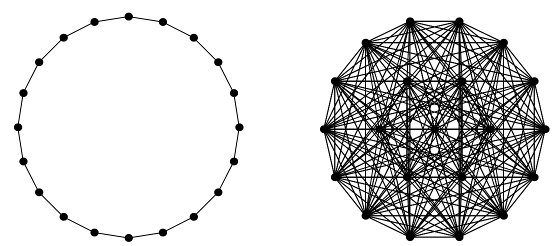

Fig. 1. The Ring topology and the Fully connected topology

B. The Random Population Topologies

In real social groups, the communication mechanisms are never as invariable as the above two models in the whole evolutionary period, but usually are somehow stochastic, dynamic and changeable. For convenience of analysis, a population topology in PSO could be usually abstracted into an undirected connected graph, which is determined by $G(V, E)$, where V stands for vertex set, and E stands for edge set, and the number of vertexes is n. For two vertices u and v in G, use $d(u, v)$ to determine the distance, i.e. the shortest path between them.

The average degree of a population topology means the average number of neighborhood particles that one particle maintains, which stands for the socializing degree of the swarm. The less neighborhoods one particle has, the harder for it to get information from the population and have effect on others; and those with many neighborhoods, on the other hand, can get much available information and have a greater influence on the population. Fully connected topology has the largest average degree which is $size - 1$, while ring topology is with the least average degree, and it is 2 in Figure 1, for example.

Definition 1 (Average degree): The Degree of a vertex is the number of points in its neighborhood, and is defined as k_v. The Average Degree of undirected connected graph for a population topology can be formulated as equation 6.

$$K = \frac{\sum_{v \in V} k_v}{|V|} \tag{6}$$

Mendes firstly conducted the research on random population topology and confirmed that the structures of population topology of PSO have direct and crucial effect on its performance [7]. Clerc proposed a method of Random Population topology, and the basic idea is: generate random topologies by means of choosing neighborhood for particles randomly [8].

TABLE I. DESCRIPTION OF THE SYMBOLS IN THE EQUATION 1 AND 2

Symbol	Description
v_{id}	The dth dimension of the particle i's velocity
x_{id}	The dth dimension of the particle i's position
p_{id}	The dth dimension of the best position which the particle i obtained
p_{gd}	The dth dimension of the best position which the population obtained
c_1, c_2	The positive factor
$rand()$, $Rand()$	The random number generator between 0 and 1
w	The inertia weight

TABLE II. DESCRIPTION OF THE SYMBOLS IN THE EQUATION 3, 4 AND 5

Symbol	Description
t	The number of particle's evolution generation
i	The particle's index number
$X_i(t)$	The position vector of particle i in the tth generation
k	The index number of particle's neighborhood
K	The quantity of particles in its neighbor area
P_k	The optimum position among the particle k's neighborhood particles
d	The dimension's index of the particles i's position vector
$CT_i(t)$	An abbreviation of Centralized Tendency, which is a D-dimensional vector and is determined by equation 4
$OT_i(t)$	An abbreviation of Outlier Trend, which is a D-dimensional vector and is determined by equation 5
α, β, γ	The positive constant factor
$Gen()$	A dynamic probabilistic evolutionary operator (a random number generator that is satisfy a specific distribution)

And based on Clerc's idea, Ni et al. improved the generating method of random population topology [9], and the improved method of generating random population topology could be described as algorithm 1.

Algorithm 1: The improved method of generating random population topology

For a population with the size of S, set up a matrix L of $S \times S$, and let $L(i,i) = 1$;
Determine a value of K, for every row i in the L matrix, generate a random number $m (m \neq i)$, where m is evenly distributed in $\{1...S\}$ and could be repeatedly selected. Let $L(m,i) = L(i,m) = 1$;
Use Dijkstra algorithm to calculate the distances between particles and store these distances in a $S \times S$ matrix D;
while *The graph representing the generating random population topology is unconnected* **do**
    Scan the distance matrix D;
    IF there are two particles u and v are unconnected;
    Let $L(u,v) = 1$;

In the matrix L obtained eventually, if $L(u,v) = 1$, then u and v is connected. This method could produce a random population topology whose average degree is slightly greater than K. Ni's improvement is to guarantee the connectivity of the corresponding undirected graph of the generating random population topology.

C. Strategies of Dynamic Random Population Topology

During the evolution of a population, it takes long time for those distant particles to transmit information. When one particle find a local optimum and itself as well as its neighborhood converge to this local best solution, another part of particles faraway may converge to another local best solution. Because these two parts of particles are relatively isolated. In the case of static population topology, communication becomes tough, which can lead them to continuous mining of local optima and ignoring the exploration of global optima.

However, if dynamic population topology is adopted and population topology is changing dynamically during the evolution, information could be exchanged sufficiently between a particle and its new neighborhoods. Although some particles may have been converged to local optimum gradually, it still can be newly informed by other particles in the population as soon as a new population topology is constructed. This will update the particles' direction of exploration and help them escape from local optima and reach global best solution.

In this paper, the proposed dynamic population topology strategies are based on random population topology which is described in section III-B. The basic idea is as follows: when initializing a population, produce a random population topology and set certain conditions for the reconstitution of population topology; identify these conditions by every evolved generation, and if conditions are met, produce a new population topology using the method described in algorithm 1. That is to say, during the evolution, the neighborhoods of particles would be changed and this could enable them to exchange information with different particles. In different period of evolution, we choose appropriate index of graph theory characteristics (an average degree of K), and produce a random population topology holding such characteristics, and make it better for the population to accommodate the evolution.

In this paper, three strategies are designed for population topology reconstitution: Fixed Cycle Based Dynamic Population Topology, Optimum Updating Status Based Dynamic Population Topology and Hybrid Strategy Dynamic Population Topology. When using these three strategies, the K value would be set upper and lower limits according to population size and and other indicators. These strategies can be expressed as follows.

(1) Fixed Cycle Based Dynamic Population Topology

Fixed Cycle Based Dynamic Population Topology (*DynamicA* for short), means generating a new population topology randomly by every certain number of generations which can be represented by a variable M. To avoid premature convergence at the early stage of evolution, the average degree of random population topology is lesser (K is set to 1) in the initializing period, which gives particles more possibility to explore the whole solution space. With the evolution generations growing, particles begin to converge to an optimum. In this case, the average degree will be increased gradually, we could use a greater K to produce a random population topology. This could deliver the information of optimal solution to particles as much as possible and enable a deeper exploitation and insure a better result. The detail strategy of *DynamicA* is as algorithm 2.

Algorithm 2: Fixed Cycle Based Dynamic Population Topology (*DynamicA*)

Generate a random population topology with the initial K ($K = 1$);
Set the evolution generation counter to 0;
while *The termination condition is not satisfied* **do**
  Evolve the particles of population ;
  Generation counter increases 1;
  if *The generation counter is evenly divisible by* M **then**
    Let $K = K + 2$;
    Produce a new random population topology;

(2) Optimum Updating Status Based Dynamic Population Topology

Optimum Updating Status Based Dynamic Population Topology (*DynamicB* for short), means changing the population topology when the present optimum haven't been updated for a certain time (configurable, and 200 in this paper) of evolution. If the present optimum keeps still throughout many generations, the average degree of generating random population topology will decrease to avoid being trapped into local optimum and stopping new exploration, that is to produce random population topology with lesser K value. Because of the decrease of K during the evolution, it is initialized with a greater value, which may lead to a rapid convergence at the beginning of evolution. The detailed strategy of *DynamicB* is as algorithm 3.

Algorithm 3: Optimum Updating Status Based Dynamic Population Topology (*DynamicB*)

Generate a random population topology with the initial K ($K = 9$);
Set a variable b to count the unchanging times of an optimum, b is initialized to 0;
while *The termination condition is not satisfied* **do**
  Evolve the particles of population;
  if *The optimum has been updated* **then**
    Set b to 0;
  else
    $b = b + 1$;
  if b *has reached 200* **then**
    Let $K = K - 2$;
    Produce a new random population topology;
    Set b to 0;

(3) Hybrid Strategy Dynamic Population Topology

Hybrid Strategy Dynamic Population Topology (*DynamicC* for short) combines the characteristics of *DynamicA* and *DynamicB*, and once one condition either in *DynamicA* or *DynamicB* is met, generate a new random population topology. The detail strategy of *DynamicC* is as algorithm 4.

Algorithm 4: Hybrid Strategy Dynamic Population Topology (*DynamicC*)

Generate a random population topology with the initial K ($K = 1$);
Set the evolution generation counter to 0;
Set the unchanging times b to 0;
while *The termination condition is not satisfied* **do**
  Evolve the particles of population;
  Generation counter increases 1;
  if *The generation counter is evenly divisible by* M **then**
    Let $K = K + 2$;
  else if *The optimum has been updated* **then**
    Set b to 0;
    else
      $b = b + 1$;
      if b *has reached 200* **then**
        Let $K = K - 2$;
  Produce a new random population topology;
  Set b to 0;

IV. Experiment Results and Analysis

A. Experiment Settings

In this paper, comparison and evaluation are conducted among the fully connected topology (Gbest), the ring topology (Lbest) and the three dynamic random population topologies (*DynamicA*, *DynamicB* and *DynamicC*), five population topology types in total. The PSO algorithm adopts the DPPSO-Logistic, which is a variant of dynamic probabilistic particle swarm optimization, and $Gen()$ is a random number generator which satisfies the Logistic distribution. Five benchmark functions are tested which are Rastrgin, Schaffer F6, Ackley, Schwefel and Sphere. The details of these benchmark functions can be seen in Table III.

In this experiment, size of population is set to 20, and except Schaffer F6 is tested in 2 dimension, all other functions are tested in dimension of 30. Experiment was repeated 50 times. The evaluation indicators of the experiment are concerned as follows. Firstly, by a certain number of generations, compare the precision of the optimal values obtained in each algorithm, i.e. the values of optima obtained finally, the indicator is denoted as *Perform.*, which could reflect the quality of the solution each algorithm obtained. Secondly, by a certain number of generations, set an accuracy value for each benchmark function which is given as *Accepted error* in Table III, and compare the success rate that the fitness value reached the accuracy during the evolution, the success rate is denoted as *Prob.*, which shows the stability of an algorithm.

B. Results and Analysis

Data tables and evolutionary trend figures are used to compare and evaluate the performance of DPPSO-Logistic

TABLE III. THE BENCHMARK FUNCTIONS USED IN THIS EXPERIMENT AND INSTRUCTIONS

Sphere	Formula	$f(\vec{x}) = \sum_{i=1}^{n} x_i^2$			
	Dimension	Optimal solution	Optimal value	Range	Accepted error
	30	$(0, 0, 0..., 0)$	0	$(-100, 100)$	0.01
Schaffer F6	Formula	$f(\vec{x}) = \frac{sin^2\sqrt{x_1^2+x_2^2}-0.5}{[1+0.001(x_1^2+x_2^2)]^2} - 0.5$			
	Dimension	Optimal solution	Optimal value	Range	Accepted error
	2	$(0, 0)$	0	$(-100, 100)$	0.00001
Schwefel	Formula	$f(\vec{x}) = 418.9829 \cdot n + \sum_{i=1}^{n} x_i sin\sqrt{\lvert x_i \rvert}$			
	Dimension	Optimal solution	Optimal value	Range	Accepted error
	30	$(0, 0, 0..., 0)$	0	$(-500, 500)$	6000
Ackley	Formula	$f(\vec{x}) = -20 \cdot exp(-0.2\sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n} \cdot \sum_{i=1}^{n} cos(2\pi x_i)) + 20 + exp(1)$			
	Dimension	Optimal solution	Optimal value	Range	Accepted error
	30	$(0, 0, 0..., 0)$	0	$(-30, 30)$	5
Rastrigin	Formula	$f(\vec{x}) = \sum_{i=1}^{n}[x_i^2 - 10cos(2\pi x_i) + 10]$			
	Dimension	Optimal solution	Optimal value	Range	Accepted error
	30	$(0, 0, 0..., 0)$	0	$(-5.12, 5.12)$	100

adopting different population topologies. The obtained *Perform.* and *Prob.* data are shown in Table IV, and boldface values in the table indicate the dominant.

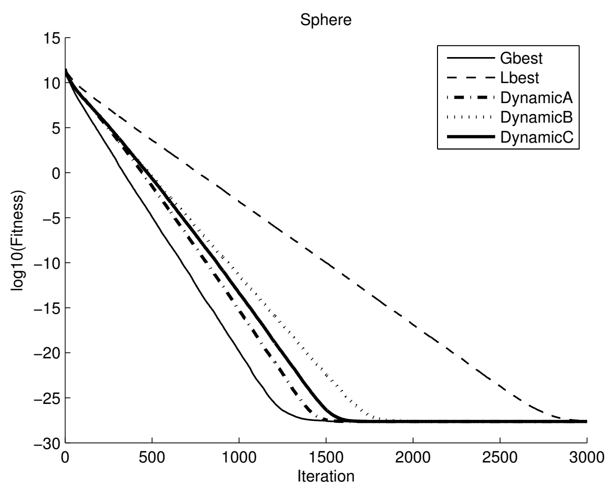

Fig. 2. Comparison of evolutionary trend about five topologies (Sphere)

For Sphere Function which is unimodal, as can be seen in Table IV and Figure 2, in terms of index values, the three dynamic random population topologies had a slight lead, but not much notable; as for the evolutionary trends, the fully connected topology (Gbest) took the lead at the early stage, while these three dynamic topologies showed their excellence in the middle and latter period.

For Schaffer F6 Function, Table IV and Figure 3 illustrated that, *DynamicA* performed best in the three dynamic random population topologies and was better than anyone else. Overall, dynamic random ones performed better than static ones.

For Schwefel Function, from Table IV and Figure 4 we can see, fully connected topology (Gbest) had a fast convergence at early stage of the evolution, but in the middle and later period, *DynamicA* and *DynamicB* could still maintain a better ability of exploration. In a word, from both index values and evolutionary trends, dynamic random population topologies were all superior to static ones.

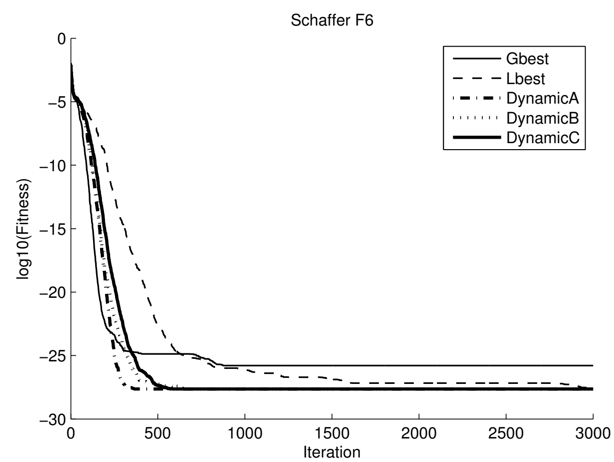

Fig. 3. Comparison of evolutionary trend about five topologies (Schaffer's F6)

For Ackley Function, which can be seen in Table IV and Figure 5, *DynamicA* performed best among all. Generally speaking, in view of evolutionary trends, the evolutionary speeds and the value of solutions obtained by the three dynamic random population topologies were all ahead of those of static.

For Rastrigin Function, as can be seen in Table IV and Figure 6, *DynamicC* and *DynamicA* performed better among the three dynamic random population topologies, *DynamicB* was slightly worse compared with the previous two topologies, but they were all better than classical static topologies. Also,

TABLE IV. PERFORMANCE COMPARISON OF DPPSO-LOGISTIC ADOPTING DIFFERENT POPULATION TOPOLOGIES

Benchmark Function	Topology	Gbest	Lbest	DynamicA	DynamicB	DynamicC
Sphere	Perform.	8.72E-32	6.11E-14	**1.32E-32**	1.70E-24	7.15E-30
	Prob.	**0.84**	0.63	0.80	0.77	0.78
Schaffer F6	Perform.	0.000777273	3.35E-16	**0**	**0**	**0**
	Prob.	0.88	0.87	**0.95**	0.94	0.94
Schwefel	Perform.	6965.39	8065.65	6570.06	**6564.42**	6609.49
	Prob.	0.14	0.00	**0.20**	**0.20**	**0.20**
Ackley	Perform.	2.1255558	**3.94E-07**	0.0459366	0.0782762	0.082886
	Prob.	**0.97**	0.94	0.96	**0.97**	0.96
Rastrigin	Perform.	130.52	112.92	**81.01**	89.57	80.43
	Prob.	0.16	0.12	**0.76**	0.60	0.67

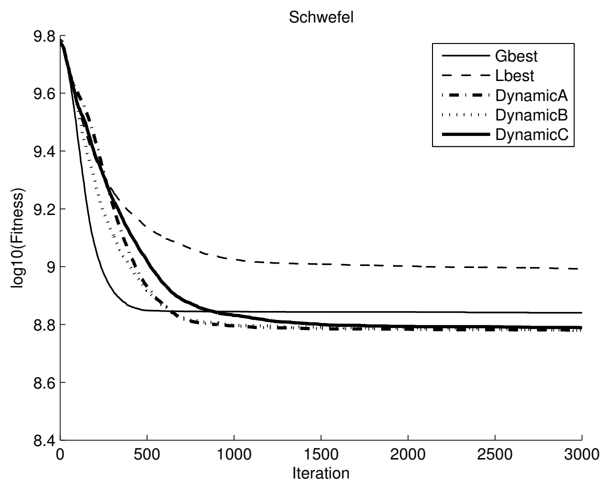

Fig. 4. Comparison of evolutionary trend about five topologies (Schwefel)

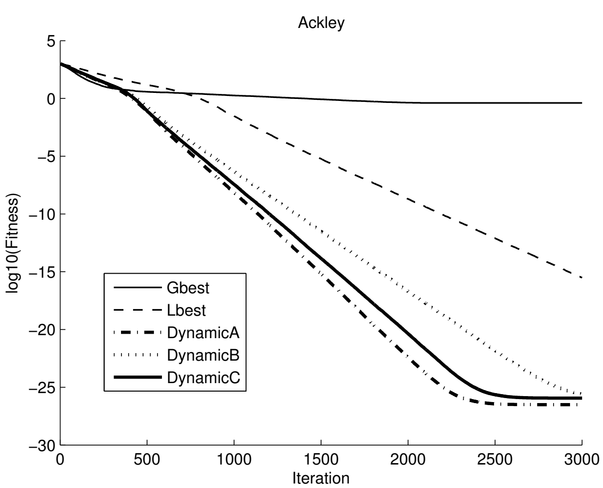

Fig. 5. Comparison of evolutionary trend about five topologies (Ackley)

for index *Prob.*, dynamic random population topologies were all better than static ones.

Experiment above compared and evaluated the three proposed dynamic random population topologies with two static population topologies (fully connected topology and ring

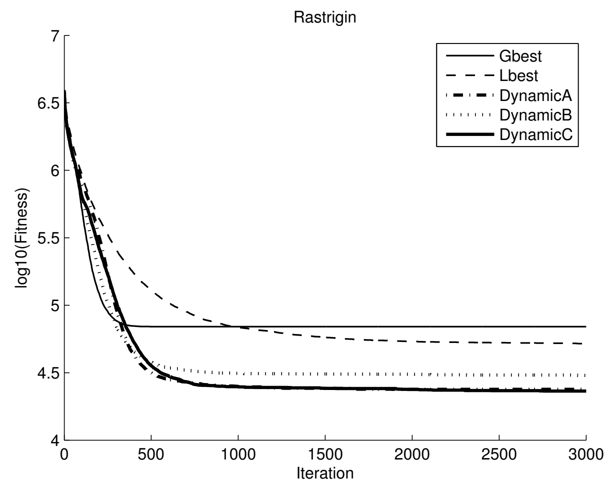

Fig. 6. Comparison of evolutionary trend about five topologies (Rastrigin)

topology). For most indicators, results demonstrate that these three proposed dynamic topologies are more dominant than static topologies. For unimodal function where there is no worry about getting trapped in local optimal, a tighter connection between particles can speed up the convergence and lead to a better solution, an increasing average degree is conducive to converging to optimum for particles. In terms of multimodal functions, population topologies are adjusted dynamically according to the average degrees at different stages, and the results are usually rather ideal.

V. CONCLUSION

Combining with graph theory characteristics of population topologies, this paper proposed three dynamic population topology strategies based on random population topology structures: Fixed Cycle Based Dynamic Population Topology, Optimum Updating Status Based Dynamic Population Topology and Hybrid Strategy Dynamic Population Topology. They are compared with classical static population topologies (the fully connected topology and the ring topology). All these topologies are tested and analyzed on five benchmark functions adopting DPPSO-Logistic algorithm. Results demonstrate that dynamic random topologies raised in this paper have an obvious superiority. That is to say, it could be of great benefit to the optimal information transfer by changing particles' neighborhoods dynamically during the evolution and making them realize information exchanging with different particles, furthermore, performance of PSO algorithm is improved. This offers a new thought to the development of PSO. Moreover, several problems deserve further discussion which include how to set index like average degree in initializing period and how to choose a suitable dynamic population topology strategy.

ACKNOWLEDGEMENTS

This paper is supported by Provincial Key Laboratory for Computer Information Processing Technology, Soochow University (KJS1223), Suzhou, China and NSFC (Grant No. 61170164).

REFERENCES

[1] A. A. Esmin, R. A. Coelho, and S. Matwin, “A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data,” *Artificial Intelligence Review*, pp. 1–23, 2013.

[2] M. AlRashidi and M. El-Hawary, “A survey of particle swarm optimization applications in electric power systems,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 913–918, 2009.

[3] R. V. Kulkarni and G. K. Venayagamoorthy, “Particle swarm optimization in wireless-sensor networks: A brief survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, no. 2, pp. 262–267, 2011.

[4] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[5] J. Kennedy, “Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance,” in *Proceedings of the Congress on Evolutionary Computation (CEC)*, vol. 3. IEEE, 1999.

[6] M. A. Montes de Oca and T. Stützle, “Convergence behavior of the fully informed particle swarm optimization algorithm,” in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. ACM, 2008, pp. 71–78.

[7] R. Mendes, “Population topologies and their influence in particle swarm performance,” Ph.D. dissertation, Universidade do Minho, 2004.

[8] M. Clerc, “Back to random topology,” *Technical report, 2007*, 2007.

[9] Q. Ni and J. Deng, “A new logistic dynamic particle swarm optimization algorithm based on random topology,” *The Scientific World Journal*, vol. 2013, 2013.

[10] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *Proceedings of IEEE World Congress on Computational Intelligence*. IEEE, 1998, pp. 69–73.

[11] J. Kennedy, “Dynamic-probabilistic particle swarms,” in *Proceedings of the conference on Genetic and evolutionary computation*. ACM, 2005, pp. 201–207.

[12] Q. Ni and J. Deng, “Two improvement strategies for logistic dynamic particle swarm optimization,” in *Proceedings of the 10th International Conference on Adaptive and Natural Computing Algorithms*. Springer, 2011, pp. 320–329.