Neural Computing and Applications (2020) 32:10007–10032
https://doi.org/10.1007/s00521-019-04527-9

ORIGINAL ARTICLE

A reinforcement learning-based communication topology in particle swarm optimization

Yue Xu[1] · Dechang Pi[1]

Received: 31 May 2019 / Accepted: 5 October 2019 / Published online: 15 October 2019

Abstract
Recently, a multitude of researchers have considered the fully connected topology (Gbest) as a default communication topology in particle swarm optimization (PSO). Despite many earlier studies of this issue indicating that the Gbest might favor unimodal problems, the topology with fewer connections, e.g., Lbest, might perform better on multimodal problems. It seems that different topologies make PSO a problem-related algorithm, while in this paper a problem-free PSO which integrates a reinforcement learning method has been proposed, referred to as QLPSO. In the new proposed algorithm, each particle acts as an agent independently, selecting the optimal topology under the control of *Q*-learning (QL) during each iteration. Two variants of QLPSO consider the different dimensions of the communication topology, respectively. In order to investigate the performance of QLPSO, experiments on 28 CEC 2013 benchmark functions are carried out when comparing with static and dynamic topologies. The reported computational results show that the proposed QLPSO is more superior compared with several state-of-the-art methods.

Keywords Particle swarm optimization · Topology · *Q*-learning · CEC 2013 benchmark

1 Introduction

Swarm and evolutionary computation has emerged as one of the most studied branches of artificial intelligence during the last decades [1]. Hundreds of novel optimization algorithms have been reported along the years and applied successfully in many applications, e.g., DNA sequence compression [2], system of boundary value problem [3], solving mathematical equations [4, 5], wireless networks [6], object-level video advertising [7], system reliability optimization [8].

Particle swarm optimization (PSO), which was first proposed in 1995 by Kennedy and Eberhart [9], is one of the most famous swarm-based optimization algorithms. The simple structure makes it easy to be implemented, and the two salient components guarantee the excellent performance of PSO. One component is a dynamical rule which governs the movement of particles, and the other is an inter-particle communication topology [10].

Due to its simplicity and high performance, a multitude of enhancements have been presented on PSO during the last few decades, which can be simply categorized into three types: parameter selection, topology, and hybridization with other algorithms. The most common enhancement is parameter selection, for example, limiting the maximum velocity [11, 12], selecting parameters such as acceleration constant [13], constriction factor [14], and inertia constant [15, 16]. A proper parameter modification "facilitates the convergence and prevents an explosion of the swarm" [17]. Banks et al. [18] pointed out that "Hybridization is a growing area of intelligent systems research, which aims to combine the desirable properties of different approaches to mitigate their individual weaknesses." To complement on each other, PSO has been combined with many other state-of-the-art algorithms, e.g., genetic algorithm (GA) [19, 20], differential evolution (DE) algorithm [21], artificial bee colony (ABC) algorithm [22], simulated annealing (SA) [23] algorithm, memetic algorithm (MA) [24], and so on.

✉ Dechang Pi
dc.pi@nuaa.edu.cn

[1] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

As shown in [25], the topology means particles in a subset can exchange information with each other. That is, the particle can interconnect with other particles in its neighborhood and share the previous best information. The communication topology in PSO can be grossly divided into two categories: static topology and dynamic topology. Most of the recent studies have considered the fully connected topology (Gbest) as a default topology. As reported in [10], "91% papers in the 2015 proceedings of the Congress on Evolutionary Computation used the Gbest topology exclusively." The rest of the papers using different topologies have focused on solving multimodal optimization problems. The PSO with Lbest or Gbest topology seems to become a problem-related algorithm. For example, it seems that Gbest favors unimodal problems, while the topology which inter-connecting with adjacent particles (Lbest) performs better on multimodal problems.

Different from the existing topologies in particle swarm optimization, this study incorporates the *Q*-learning algorithm (QL) [26] into PSO to dynamically adjust the neighborhood during the search process. The main contributions of this paper are summarized as follows:

- QLPSO is a problem-free algorithm, which is the most significant difference from current problem-related PSOs. In this paper, the neighborhood is dynamically changed under the control of a reinforcement learning method (*Q*-learning). During the iteration, the Q-table is updated according to the states of the particles and the rewards computed by fitness and diversity. Then, the *Q*-learning selects the optimal topology and returns the action.
- When embedding *Q*-learning, this paper investigates the effects of topology of different dimensions. The swarm is assigned to two topologies: a one-dimensional ring topology and a two-dimensional square topology.

Extensive computational studies are also carried out to evaluate the performance of the proposed algorithm QLPSO on 28 CEC 2013 benchmark functions. Experimental results demonstrate that the proposed algorithm outperforms the other static or dynamic methods. Moreover, the experiment also investigates the effects of QL topology with different dimensions by comparing the error and analyzing the call probability.

This paper controls the topology in PSO with *Q*-learning. The research on embedding RL into optimization algorithm to control the operation has become the hot spot gradually, with the reinforcement learning (RL) arising and developing rapid. For example, Rakshit et al. [27] introduced *Q*-learning to develop an adaptive memetic algorithm for local refinement. A reinforcement learning-based memetic particle swarm optimization (RLMPSO) was proposed in [24], and each particle selected the best-performing operator among five possible operations. Among them, four are global search operators and one is local search operator. Samma et al. [28] proposed a *Q*-learning-based simulated annealing algorithm (QLSA) to control its parameter adaptively at run time. The experimental results of the above studies verified the effectiveness of integrating the reinforcement learning method into optimization algorithm to control executions.

The rest of the paper is arranged as follows. A detailed description of the standard PSO and its variants with different topologies is presented in Sect. 2. This section further lists these topologies from a view of static topology and dynamic topology and sums up their strengths and weaknesses. Section 3 elaborates on the proposed algorithm QLPSO with different communication topologies and *Q*-learning. Experimental results and analysis are supplied in Sect. 4. This paper is ended with conclusion and forecast in Sect. 5.

2 Related works

2.1 Particle swarm optimization

Particle swarm optimization is a swarm intelligence metaheuristic algorithm, which is the representation of swarm behaviors in some ecological system, such as birds flying and bees foraging [29]. In a canonical PSO [10, 30, 31], the movement of a particle is influenced iteratively by its previous best solution and the previous best of its best neighbor. Assume that in a D-dimensional search space, the position and velocity of a particle i are denoted as x_i and v_i, respectively. As the search progresses, these two movement vectors are updated as (1) and (2).

$$v_i(t+1) = wv_i(t) + c_1u_1(t)(nBest_i(t) - x_i(t)) + c_2u_2(t)(pBest_i(t) - x_i(t)), \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (2)$$

where w is inertia weight. It is a convergence controlling parameter and equivalent to Clerc and Kennedy's constriction parameter [14]. Then, c_1 is the social acceleration coefficient and c_2 is the cognitive acceleration coefficient; u_1 and $u_2 \in [0,1]$ are random values drawn form a uniform distribution; $pBest_i$ is the personal best position for particle i, and $nbest_i$ is $pbest$ of the best neighbor.

2.2 Its variants with different topologies

2.2.1 Static topology

Topology is one of the essential components in the canonical PSO. The first two topologies were proposed in

[32] in 1995. The topology, where each particle is affected by its adjacent neighbors, is called Lbest. With the increase in the size of neighborhood, set to $N - 1$ (N is the number of the swarm) finally, each particle communicates with the rest of the particles in the swarm, and this topology is called Gbest. Experiments were carried out to compare the performance between them. The results showed that Gbest version requires less iterations to obtain a good solution, which implies better convergence, while the Lbest version is not easy to fall into local optimal. It is regarded as the beginning of the topology research.

In the next few years, many variants of PSO with static topologies would have been proposed. Four different population topologies, such as Lbest, wheel, Gbest, and random, were designed in [31]. The reported results implied that the population topologies have an important effect on solving different problems: The topology with fewer connections would be better at solving multimodal problems, while the fully interconnected population is more suitable for solving unimodal optimization problems. Inspired by the paramount importance of population topologies, Kennedy and Mendes investigated on the effects of various population topologies on PSO [25]. Figure 1 depicts several common static population topologies, such as Lbest, wheel, Gbest, von Neumann, four-cluster, and pyramid. Detailed descriptions are listed as follows [25, 31, 33].

- Lbest: It is also called ring, defined as adjacent members comprise the neighborhood.
- wheel: In this topology, one central node influences, and is influenced by all other members of the population. This topology is also named as star, because of its shape.
- Gbest: Members are all arranged in a ring. Different from Lbest topology, each member can communicate with all other ones. It is the most common topology in PSO.
- von Neumann: In this topology, members above, below, and on each side on a two-dimensional lattice comprise the neighborhoods.
- four clusters: It is defined as a graph with four cliques connected among themselves by gateways.
- pyramid: This topology is based on a three-dimensional wire-frame triangle.

2.2.2 Dynamic topology

The above population topologies, as illustrated in Fig. 1, are all static. The main feature of this kind of topologies is that the neighborhood of each particle is fixed at the beginning and stays the same during the iteration, while in dynamic topology the size or structure of neighborhood or the way of exchanging information would change during the iteration. The main benefit of the dynamic topology is to avoid getting in the local best solution when solving more complex problems.

Hence, the dynamic topology in PSOs has become a research hot spot to solve complex optimization problems due to their efficiency. For example, due to the discovery that Lbest topology seems better for exploring the search region while the Gbest converges faster, Suganthan [34] increased the size of neighborhood from Pbest to Gbest. The reported results showed the superiority of using a neighborhood operator. Inspired by Mendes et al. [34], a time-adaptive topology was proposed, in which the communication is updated from small sub-swarms to fully connected topology Gbest [35]. The experiments indicated that the disjoint feasible regions have been found at the beginning of iteration by nonoverlapping topology with few particles, and the best region would be concentrated later in the iteration by the topology with large number of particles. Besides, several variants of PSO based on the similar idea have also been introduced, i.e., PSO with increasing topology connectivity [36] and PSO with expanding neighborhood topology [37].

In addition to changing the size of neighborhood, fitness sharing and speciation are two classic representations of dynamic topology. The concept of fitness sharing was originated from [38] in genetic algorithm, and subsequently applied in PSO [39, 40]. In the fitness sharing, the fitness of each individual was degraded to "limit the number of individuals occupying the same niche" [41]. In speciation, the swarm was divided into multi-species according to species radius, and the particles in the same specie shared information to explore the search region [42]. The main

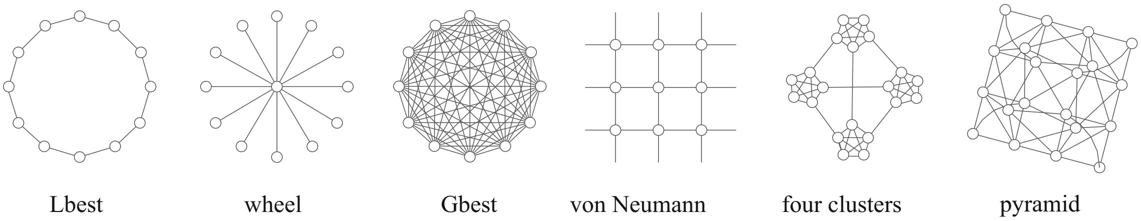


Fig. 1 Static population topologies

similarity of these dynamic topologies is that regardless of the sharing radius and scaling factor in fitness sharing or specie radius in speciation, it is not easy to set proper value for these niching parameters.

To tackle this problem, another dynamic topology identifies the neighborhood by fitness distance. A PSO with fitness distance ratio, named as PSO-FDR, was proposed in [43]. The *nbest* in each dimension was the particle that maximized the fitness distance ratio. On the basis of PSO-FDR with the basic concept of encouraging the survival of fitter and closer particles, Li introduced fitness Euclidean-distance ratio to calculate the best neighbor and called the new version as PSO-FER [44]. The results indicated that PSO-FER has provided excellent performance on dealing with multimodal problems. Furthermore, a distance-based locally informed particle swarm (LIPS) optimizer was presented, which fully utilized all neighborhood information and enhanced the fine search ability by Euclidean distance-based neighborhood selection [45]. The advantage of this type of methods is removing the requirement of specifying niching parameters. However, more calculation on fitness distance between particles is also needed.

To intuitively show characteristics about these variants, Table 1 is listed as a brief summary on the methods with their strengths and weaknesses, from the view of static topology and dynamic topology.

In conclusion, the dynamic topology seems more effective than the static one at the cost of updating neighborhood, niching parameter setting or fitness distance calculation. The static topology without niching parameters is simple and easy to be implemented.

In this paper, a novel communication topology based on reinforcement learning is proposed to strengthen the performance of PSO. The proposed algorithm QLPSO adjusts the neighbor structure for each particle dynamically through QL. The proposed algorithm is more effective with few parameters. Besides, it is a problem-free algorithm, which is the most significant difference from most current problem-related PSOs. Due to combining with QL, more calculation on reward function and updating Q-table are also needed.

Table 1 A brief summary on variants of PSO with different topologies

Type	Methods with their strengths and weaknesses	
Static	Methods	Lbest and Gbest [32]
		Lbest, wheel, Gbest, and random [8]
		Lbest, Gbest, pyramid, star, small, and von Neumann [25]
	Strength	Simple and easy to be implemented
		Without niching parameters
	Weakness	Only the same type of problem can be solved (e.g., Gbest favors unimodal problems, while Lbest performs better on multimodal problems)
Dynamic	Methods	Changing the size of neighborhood [34–37]
	Strength	More effective
		Few parameters
	Weakness	Need computation on updating neighborhood
		Premature for complex problems
	Methods	Fitness sharing [39, 40]
		Speciation [42]
	Strength	More effective
	Weakness	Difficult for parameter setting
		Need to recalculate the fitness or divide the swarm into subpopulations
	Methods	Fitness distance [43–45]
	Strength	More effective
		Removing the requirement of specifying niching parameters
	Weakness	Need for calculation on fitness distance between particles

2.2.3 Trend of topology research

The trend of Lbest and Gbest topology in PSO can be summed up as origin, turning point, and new point. The PSO with Lbest and Gbest topology was first published in 1995, and since then, more and more studies have recognized the effect of communication topology on the performance of the PSO, while in 2013 Engelbrecht deemed the opinion that the PSO with Lbest topology should be more preferred than that with Gbest as intuitive expectations [46]. As quoted in [46], "neither of the two algorithms can be considered the outright best for any of the function classes." These results led a rise of Gbest (turning point), which has gradually become the default topology in PSO. Here came a new point in 2018. Blackwell and Kennedy [10] compared the performance of Lbest and Gbest topology in PSO on a comprehensive benchmark suite: repeating 37 base functions in Engelbrecht's study, adding 25 functions in CEC2005 and 28 functions in CEC2013. They "confirm the early lore that Gbest is the overall better algorithm for unimodal and separable problems and that a ring neighborhood of connectivity two (Lbest) is the preferred choice for multimodal, non-separable and composition functions" [10].

3 The proposed algorithm QLPSO

3.1 Notations

x_i	The position of particle *i*
$x_{\max}$	The upper boundary
$x_{\min}$	The lower boundary

v_i	The velocity of particle i
w	Inertia weight
c_1	The social acceleration coefficient
c_2	The cognitive acceleration coefficient
$pBest_i$	The personal best position for particle i
$nbest_i$	*pbest* of the best neighbor
Max_FEs	The maximum number of function evaluations
N	The population size
D	The dimension of the solution
Lk	Ring topology
Sk	Square topology
S	A set of possible state of the learning agent in a given environment
A	A set of possible actions that agent can choose and execute
Q	The total cumulative reward
α	Learning rate
γ	Discount factor
R	The immediate reward
f	The fitness function
d	The diversity function

3.2 Topology of different dimensions

As mentioned above, a majority of studies have focused on the fully connected topology (Gbest), while a majority of the rest, using different topologies, aim to solve the multimodal optimization problems due to the superiority of their local neighborhoods. This paper proposes a problem-free Q-learning-based particle swarm optimization algorithm. Whether the objective function is multimodal or not, particles dynamically select the best topology under the control of Q-learning.

This section describes in detail the Q-learning algorithm integrated in PSO and two-dimensional topologies. Furthermore, the complete framework of QLPSO is presented. The detailed descriptions are shown as follows.

As noted before, the number of edges connecting each vertex (degree) has different effects on the performance of a regular population topology. In other words, the number of neighbors directly determines the quality of the topological communication. Take Lbest and Gbest as an example; they are all arranged in a ring structure, but the connections with other particles (exchanging information) are extremely different. The Lbest topology connects with adjacent members, and the number of neighbors is 2, while in Gbest topology the particles are fully interconnected and the number of neighbors is $N - 1$. As we know, searching around the global best particle Gbest speeds up the convergence rate while exploring around the current particle from its neighborhoods can increase the population diversity so as to be closer to the optimum. Hence, as the number of neighbors decreases, the ring topology performs better for multimodal problems.

As defined in [10], the topology constructed from a ring topology by adding links to the next k nearest neighbors is a degree k ring topology, denoted as Lk. Hence, the neighborhoods of particle i can be represented as (3).

$$\left\{\left(i-\frac{k}{2}\right) \bmod N, \ldots, (i-1) \bmod N, (i+1) \bmod N, \ldots, \left(i+\frac{k}{2}\right) \bmod N\right\}, \tag{3}$$

where N is the swarm size and k is an even integer. Assume the swarm size is 10, for example a sequence of local neighborhoods L2 (Lbest), L4, L8, and L10 (Gbest) is illustrated in Fig. 2.

Different lattice structures represent different topologies of swarm and could lead to different patterns of information interaction [47]. Hence, in order to better investigate the effects of different topology structures when embedding Q-learning, this paper also tests on a two-dimensional square lattice topology. In this case, the neighborhoods of particle are based on the von Neumann topology, Moore topology, their extensions and combinations. In these two-dimensional lattice topologies, this paper denotes Sk as the square topology by adding links to the next k nearest neighbors. The neighborhoods of particle (i, j) in von Neumann and its extensions, (ni, nj), are computed as (4).

$$\left\{ (ni,nj) \middle| \begin{array}{l} ni \in \left[\left(i-\frac{k}{4}\right) \bmod len, \left(i+\frac{k}{4}\right) \bmod len\right], ni \neq i, nj = j, \text{ or} \\ ni = i, nj \in \left[\left(j-\frac{k}{4}\right) \bmod wid, \left(j+\frac{k}{4}\right) \bmod wid\right], nj \neq j \end{array} \right\}, \tag{4}$$

where len and wid are length and width of the two-dimensional square, respectively. And, len*wid = N. k in the two-dimensional lattice structure is a multiple of four. Similarly, the Moore neighborhood is composed of eight members which surround the central particle. In this paper, the neighborhoods (ni, nj) in Moore topology and its extensions can be denoted as (5).

$$\left\{ (ni,nj) \middle| ni \in \left[\left(i-\frac{k}{4}\right) \bmod len, \left(i+\frac{k}{4}\right) \bmod len\right], nj \in \left[\left(j-\frac{k}{4}\right) \bmod wid, \left(j+\frac{k}{4}\right) \bmod wid\right], ni \neq i, nj \neq j \right\}. \tag{5}$$

Taking the topologies in Fig. 3 as an example, the particle swarm is assumed to be 36. Then, particles in the swarm are assigned into a two-dimensional 6 * 6 square grid according to their initial indexes. The particle, in row 4 and column 3, is connected to k adjacent neighbors in Fig. 3.

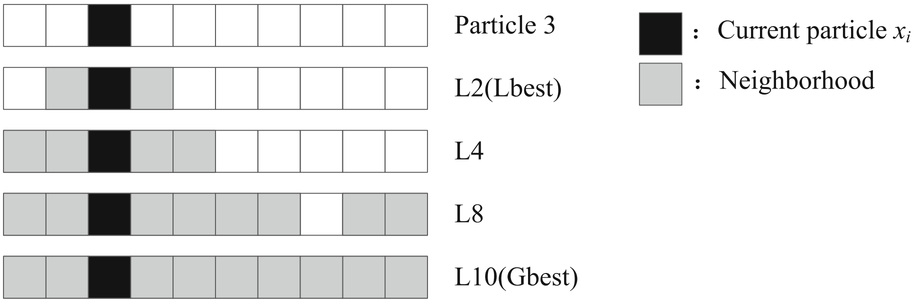


Fig. 2 Several topologies constructed from a ring

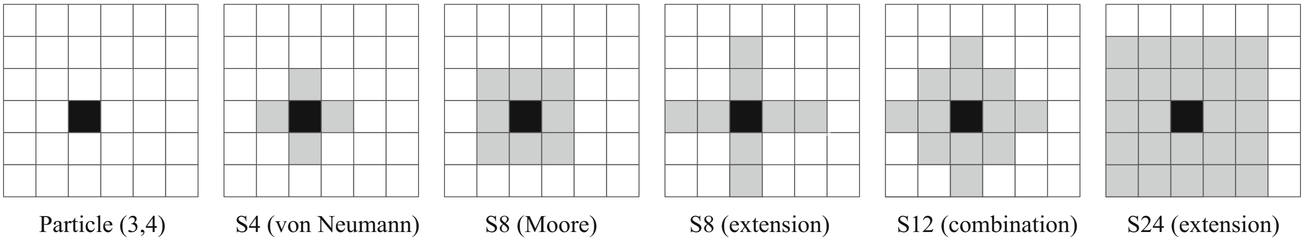


Fig. 3 Several topologies constructed from a two-dimensional square

Topologies S4 and S8 are von Neumann and Moore neighborhood of the particle, separately. S12 is the combination, and the rest are extensions of these two topologies.

One important thing to note is that, for both the ring topology and the square topology, the particle is assigned to one unduplicated grid one by one according to its index in the swarm. The assignment is set in the initialization stage of the algorithm and remains unchanged during all iterations. The purpose of this is to make the algorithm simple, stable, and easy to be implemented.

3.3 *Q*-learning

Q-learning (QL) is a model-free reinforcement learning algorithm for agents to learn how to act optimally in controlled Markovian domains [26]. In QL, agent usually performs an action through state transition in the environment and receives a reward or a penalty by executing the action to reach the goal state [48], as shown in Fig. 4.

As shown in Fig. 4, QL includes five basic components, i.e., agent, environment, action, state, and reward. Considering that S is a set of possible state of the learning agent in a given environment, and A is a set of possible actions that agent can choose and execute, these components of agent i at time t can be represented as (6).

$$s_t^i \in S, a_t^i \in A, r_t^i \in S \times A \rightarrow \mathbb{R}. \tag{6}$$

Hence, the total cumulative reward, referred to as *Q*-value, can be updated as (7).

$$Q_{t+1}(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a_t)\right], \tag{7}$$

where α and γ are learning rate and discount factor, respectively, and both within [0,1]. As can be seen from (7), the learning rate is a balance between exploration and exploitation. That is, low value of learning rate makes the algorithm learn more about the existing information, while the larger the learning rate α, the lesser the retention of the previous *Q*-value. Hence, α normally is set to a high value at the beginning of the search process, and is decreased during the iterations [24, 27].

$$a(t) = 1 - \left(0.9 * \frac{t}{Max\_FEs}\right), \tag{8}$$

where *Max_FEs* is the maximum number of function evaluations. Besides, the discount factor is a trade-off between previous experience and immediate reward. The greater the value of γ, the algorithm attaches more importance to the previous information; and the smaller the value of γ, the algorithm is more concerned about the reward. Usually, it is set to 0.8, as also suggested in [24, 27]. Calibration of parameters in Sect. 4.2 verified the rationality of these parameter values. According to the above definitions, the pseudo-code of QL is described in Algorithm 1.

Algorithm 1. The pseudo code of QL

```
Initialize Q($s$,$a$)=$zero$ in Q-table;
Randomly select an initial state $s$;
Repeat
    Select the best action $a_t$ for $s_t$ from the Q-table;
    Execute action $a_t$ and get a new solution;
    Get the maximum Q value for the next state $s_{t+1}$;
    Obtain the immediate reward $r_t$;
    Update Q-table entry using (7);
    $s_t = s_{t+1}$;
Until the $Max\_FEs$ is met.
```

In this paper, QL is integrated with PSO to select the optimum size of neighborhood according to the reward from the search region. Let each particle act as an agent, and the environment be the search space. One thing to notice is that each particle has its own Q-table. The states of each particle are the possible neighborhoods, which can be chosen in a regular topology structure. Changing from one neighborhood to another neighborhood is its action. The improvements in the fitness and diversity are two measurements of the reward. This paper computes the immediate reward as (9).

$$r = \begin{cases} 2, & f(x_{\text{new}}) < f(x_{\text{old}}) \text{ and } d(x_{\text{new}}) > f(x_{\text{old}}). \\ 1, & f(x_{\text{new}}) < f(x_{\text{old}}) \text{ and } d(x_{\text{new}}) \leq f(x_{\text{old}}). \\ 0, & f(x_{\text{new}}) \geq f(x_{\text{old}}) \text{ and } d(x_{\text{new}}) > f(x_{\text{old}}). \\ -2, & f(x_{\text{new}}) \geq f(x_{\text{old}}) \text{ and } d(x_{\text{new}}) \leq f(x_{\text{old}}). \end{cases} \tag{9}$$

where *f* and *d* are fitness and diversity function, separately. And the diversity of a swarm can be calculated as (10), which was also used in [49].

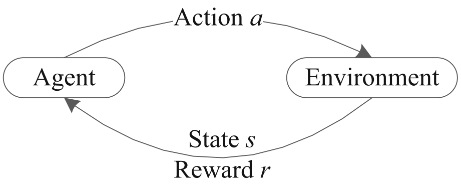


Fig. 4 Reinforcement learning

$$\begin{aligned} d(t) &= \frac{1}{N}\sum_{i=1}^{N}\sqrt{\sum_{j=1}^{D}\left(x_{ij}(t) - \overline{x_j(t)}\right)^2}, \\ \overline{x_j(t)} &= \frac{\sum_{i=1}^{N} x_{ij}(t)}{N}. \end{aligned} \tag{10}$$

Take a ring topology, illustrated in Fig. 2, as an example. L2, L4, L8, and L10 are four different neighborhoods, which are the possible states in QL. Assume that the *Q*-table of particle *i* at time *t* is shown in Fig. 5, with current state L4. Changing from L4 to L8 is the best action, which can make the particle explore more regions of the feasible search space. Assuming the reward $r = 1$, learning rate $\alpha = 0.9$, and discount factor $\gamma = 0.8$, the *Q*-table can be updated as shown in Fig. 5.

3.4 Complete framework of QLPSO

A novel particle swarm optimization based on *Q*-learning (QLPSO) is proposed in this paper. During the iterations, each particle, acting as a learning agent, selects the best neighborhood in a regular structure. The swarm is initialized randomly with the population size *N* and dimension *D*. For particle *i*, its initial position in *j*-dimension can be calculated as (11).

$$x_{ij} = x_{\min} + rand(x_{\max} - x_{\min}), \tag{11}$$

where *i* and *j* are the indexes of population size and dimension, respectively ($i = 1, 2,\ldots, N$ and $j = 1,2,\ldots D$). $x_{\max}$ and $x_{\min}$ are upper and lower boundaries, separately. *rand* is a random value drawn from a uniform distribution between 0 and 1.

To better investigate the effects of different topologies when embedding *Q*-learning, the swarm is assigned to two topologies: a one-dimensional ring topology and a two-dimensional square topology. Particles are constructed in the same topology, but different in the connection of others. The *Q*-learning algorithm selects the best neighborhood topology (action) according to the state of current particle and immediate reward. Then, each particle updates the velocity and position using (1) and (2), separately. The QLPSO repeats the above steps until the maximum number of function evaluations is met.

$$\begin{array}{c|cccc} & L2 & L4 & L8 & L10 \\ \hline L2 & -1.9989 & -1.9939 & -3.5865 & -3.5779 \\ L4 & -1.9993 & -3.5836 & -1.9935 & -1.9960 \\ L8 & -1.9941 & -2.0000 & -1.9987 & -1.9957 \\ L10 & -3.5870 & -1.9996 & -1.9920 & -1.9971 \end{array} \xrightarrow[\substack{r=1 \\ \alpha=0.9 \\ \gamma=0.8}]{t=t+1} \begin{array}{c|cccc} & L2 & L4 & L8 & L10 \\ \hline L2 & -1.9989 & -1.9939 & -3.5865 & -3.5779 \\ L4 & -1.9993 & -3.5836 & -0.7351 & -1.9960 \\ L8 & -1.9941 & -2.0000 & -1.9987 & -1.9957 \\ L10 & -3.5870 & -1.9996 & -1.9920 & -1.9971 \end{array}$$

Fig. 5 An example of QL

According to the above description, the simple framework of QLPSO is summarized in Fig. 6. It is clear that the new algorithm retains the main framework of PSO with the advantage of being simple and easy to be implemented. On this basis, QL adjusts the velocity and position of particles adaptively. Different sizes of neighborhoods make a better trade-off between exploration and exploitation.

Algorithm 2 provides the pseudo-code of QLPSO. The line 3, line 6, and line 10 to line 13 are the steps of QL, and the rest are that of PSO.

The computational complexity of PSO is $O(Max\_FEs \times N \times T_f)$, where Max_FEs is the maximum number of generations, N is the population size, and $O(T_f)$ is the computational complexity of its fitness evaluation. Compared with the basic PSO, the main difference of QLPSO is that QL is incorporated into PSO to enhance its performance by optimizing the neighborhood structure adaptively during the search process. In QL, the improvements in the fitness and diversity are two measurements of the reward, which are applied to keep track of the optimal topology. Suppose that $O(T_d)$ is the computational

Algorithm 2. The pseudo code of QLPSO

```
01. Uniformly randomly initialize each particle in the swarm using (11);
02. Initialize $pbest$ and $gbest$;
03. Initialize Q-table;
04. For $FEs$ =1 to $Max\_FEs$ do
05.     For $i$ =1 to $N$ do
06.         Select the best action $a_t$ for $s_t$ from the Q-table;
07.         According $a_t$, calculate the best neighbor $nbest_i$;
08.         Update the velocity and position using (1) and (2) for particle $x_i$;
09.         Compute the fitness and diversity of the new solution $x_{new}$;
10.         Get immediate reward $r_t$ using (9);
11.         Get the maximum Q value for the next state $s_{t+1}$;
12.         Update Q-table entry using (7);
13.         Update the current state $s_t = s_{t+1}$;
14.         Update $pBest_i$ and $gbest$;
15.     End
16. End
```

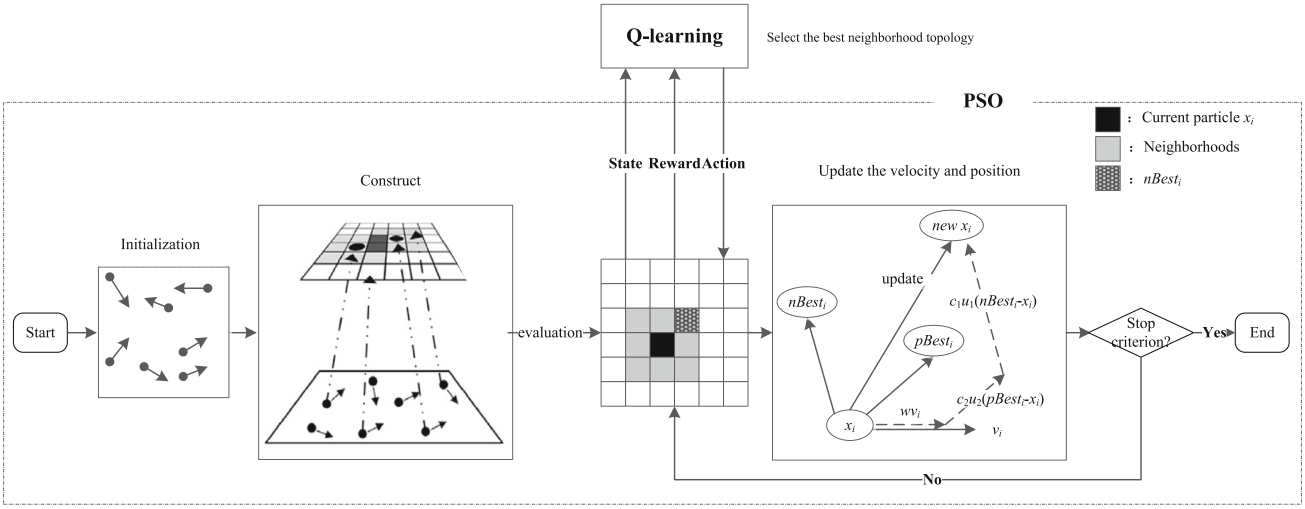


Fig. 6 The framework of QLPSO

complexity of its diversity evaluation, and as a result, the entire computational complexity of QLPSO is $O(Max\_FEs \times N \times (T_f + T_d))$.

4 Experimental study

4.1 Experiment setup

In order to verify the performance of QLPSO, 28 CEC 2013 benchmark functions [50] are listed in Table 2. This benchmark suite is a real-parameter single-objective optimization problem, including 5 unimodal functions (F01–F05), 15 basic multimodal functions (F06–F20), and 8 composition functions (F21–F28), which is available at https://www.ntu.edu.sg/home/EPNSugan/. For these functions, the search range of every dimension is the same, from − 100 to 100. The aim of function optimization is to find the minimum of this objective function. The measurement of the algorithm performance is function error $(f(x) - f(x^*))$. The errors are the differences between the best solutions obtained by algorithms and true global optima recorded for CEC13 (as shown in Table 2).

To make a fair comparison, each algorithm is run 51 times for each instance and the mean error of results will be displayed as the comparative criterion. In the following Experiment 1–3, the dimension of all benchmark D and the population size N are all set to 30, which were recommend from [10]. All runs are terminated at 150,000 function evaluations. Furthermore, the velocities are initialized to zero for all particles.

4.2 Calibration of parameters

To calibrate the parameters of QLPSO, the Design of Experiments (DOE) methodology [51] is employed in this section. The proposed DLPSO has the following key controlled factors: inertia weight w, social acceleration coefficient c_1 and cognitive acceleration coefficient c_2 ($c = c_1 = c_2$), learning rate α, and discount factor γ.

Table 2 Summary of CEC 2013 benchmark functions

No.	Property	Function	Optimum
F01	Unimodal	Sphere function	− 1400
F02		Rotated high-conditioned elliptic function	− 1300
F03		Rotated bent cigar function	− 1200
F04		Rotated discus function	− 1100
F05		Different powers function	− 1000
F06	Multimodal	Rotated Rosenbrock's function	− 900
F07		Rotated Schaffers F7 function	− 800
F08		Rotated Ackley's function	− 700
F09		Rotated Weierstrass function	− 600
F10		Rotated Griewank's function	− 500
F11		Rastrigin's function	− 400
F12		Rotated Rastrigin's function	− 300
F13		Non-continuous rotated Rastrigin's function	− 200
F14		Schwefel's function	− 100
F15		Rotated Schwefel's function	100
F16		Rotated Katsuura function	200
F17		Lunacek Bi_Rastrigin function	300
F18		Rotated Lunacek Bi_Rastrigin function	400
F19		Expanded Griewank's plus Rosenbrock's function	500
F20		Expanded Schaffer's F6 function	600
F21	Composition	Composition function 1 (n = 5, Rotated)	700
F22		Composition function 2 (n = 3, unrotated)	800
F23		Composition function 3 (n = 3, rotated)	900
F24		Composition function 4 (n = 3, rotated)	1000
F25		Composition function 5 (n = 3, rotated)	1100
F26		Composition function 6 (n = 5, rotated)	1200
F27		Composition function 7 (n = 5, rotated)	1300
F28		Composition function 8 (n = 5, rotated)	1400

According to some preliminary experiments and related studies [20, 24, 27], the potential values (levels) chosen for these parameters are $(w,c) \in \{(0.9 \sim 0.4, 2.0), (0.729844, 1.49618)\}$, $\alpha \in \{1 \sim 0.1, 0.25\}$, $\gamma \in \{0.6, 0.8, 1.0\}$, where $w = 0.9 \sim 0.4$ means inertia weight w linearly decreases from 0.9 to 0.4. Each parameter configuration is executed 5 times on 28 CEC functions. In summary, there are a total of 2 * 2 * 3 * 5 * 28 = 1680 independent runs needed for tuning parameters of QLPSO. In addition, the normalized function error $(f(x) - f(x^*))$ is chosen as the response value in this paper.

The results of ANOVA of parameter calibration for QLPSOs are listed in Tables 3 and 4. Note that [52], "for the results of ANOVA, the *F*-ratio is a clear indicator of significance when *p* value is close to zero. A large *F*-ratio indicates that the analyzed factor has a considerable effect on the response variable. Meanwhile, it is also remarkable that calibration is a fine-tuning process and algorithms are not expected to behave entirely different after calibration." As shown in Tables 3 and 4, the *p*-values of (w,c) and γ are less than 0.05 level for both QLPSO_{1D} and QLPSO_{2D}, which indicates that (w,c) and γ significantly affect the performance of the proposed algorithm. To be specific, (w,c) has the greatest *F*-ratio and *F*-ratio obtained by γ is slightly lower than (w,c) for QLPSO_{1D}, while for QLPSO_{2D} γ has the greatest *F*-ratio, and *F*-ratio obtained by (w,c) is slightly lower than γ. Therefore, we can conclude that (w,c) and γ all have the greatest impact on the proposed algorithm QLPSOs.

Moreover, the main effects plots of all parameters for QLPSO_{1D} and QLPSO_{2D} are depicted in Figs. 7 and 8, separately. It is obvious that QLPSO_{1D} and QLPSO_{2D} obtain similar main effect plots. Generally, the lower the average response, the better the algorithm. As shown in Figs. 7 and 8, QLPSO_{1D} and QLPSO_{2D} all achieve the best performance when $w = 0.729844$, $c = 1.49618$, α linearly decreases from 1 to 0.1, $\gamma = 0.8$.

However, if there are significant interactions between parameters, the main effect plot is not meaningful [52]. Hence, it is necessary to check interactions between parameters. As shown in Table 3, $(w,c)*\gamma$ interaction is significant since its *p*-value is less than 0.05 level. The interaction plot for (w,c) and γ is given in Fig. 9. From this figure, it can be seen that the results of Fig. 8 are in accordance with the conclusions above.

Therefore, based on the analysis above, the parameters of QLPSOs can be set as follows: $w = 0.729844$, $c = 1.49618$, α linearly decreases from 1 to 0.1, $\gamma = 0.8$.

4.3 Experiment 1: comparing to static topologies

4.3.1 In one-dimensional ring topology

To confirm the contribution of QLPSO with one-dimensional ring topology, the following experiments are carried out. In the first trail, PSO with QL is executed, denoted as QLPSO_{1D}. In the following trails, PSO is executed with L2 (Lbest), L4, L8, L16, and L30 (Gbest), separately. The results, mean error which is considered as the main statistical feature, obtained by these trails are shown in Table 5 (the best results are in bold).

As shown in Table 5, it is obvious that PSOs with different L*k* ring topologies perform differently on CEC13 functions. PSO with L8 ring topology obtains the best results on unimodal functions (F01, F02, and F05), while PSO with Lbest or Gbest performs worse than L8 on all unimodal functions (F01–F05). It implies that PSO-L8 is the preferred choice for unimodal problems. A reasonable inference for this is that PSO-L8 converges faster than PSO-L2 and searches more widely than PSO-L30. Furthermore, QLPSO_{1D} produces the best solution in F03, which means the proposed algorithm is excellent when solving some simple unimodal problems.

In multimodal and composition problems, the QLPSO_{1D} achieves better results on 10 functions (F07, F09, F11, F14, F17, F19, F22, F24, F25, F27), 10 out of 23 functions. PSO-L2 obtains the best results on 4 functions (F06, F21, F26, F28), PSO-L4 on 3 functions (F12, F13, F23), PSO-L8 on 3 functions (F15, F18, F20), PSO-L16 on F10, and PSO-L30 on 2 functions (F08 and F16). The proposed

Table 3 Results of ANOVA of parameter calibration for QLPSO_{1D}

Source	Sum of squares	Degrees of freedom	Adjusted mean square	*F*-statistic	*p*-value
A: (w,c)	3.195	1	3.19509	**27.66**	**0.0000**
B: α	0.028	1	0.02849	0.25	0.6195
C: γ	6.274	2	3.13681	**27.16**	**0.0000**
Interaction					
AB	0.078	1	0.07835	0.68	0.4103
AC	0.820	2	0.40988	3.55	**0.0290**
BC	0.059	2	0.02956	0.26	0.7742
Error	192.909	1670	0.11551		
Total	203.363	1679			

Table 4 Results of ANOVA of parameter calibration for QLPSO$_{2D}$

Source	Sum of squares	Degrees of freedom	Adjusted mean square	*F*-statistic	*p*-value
A: (w,c)	1.024	1	1.02362	**8.33**	**0.0040**
B: α	0.042	1	0.04198	0.34	0.5590
C: γ	2.050	2	1.02483	**8.34**	**0.0002**
Interaction					
AB	0.142	1	0.14248	1.16	0.2818
AC	0.374	2	0.18680	1.52	0.2190
BC	0.539	2	0.26974	2.19	0.1117
Error	205.24	1670	0.1229		
Total	209.41	1679			

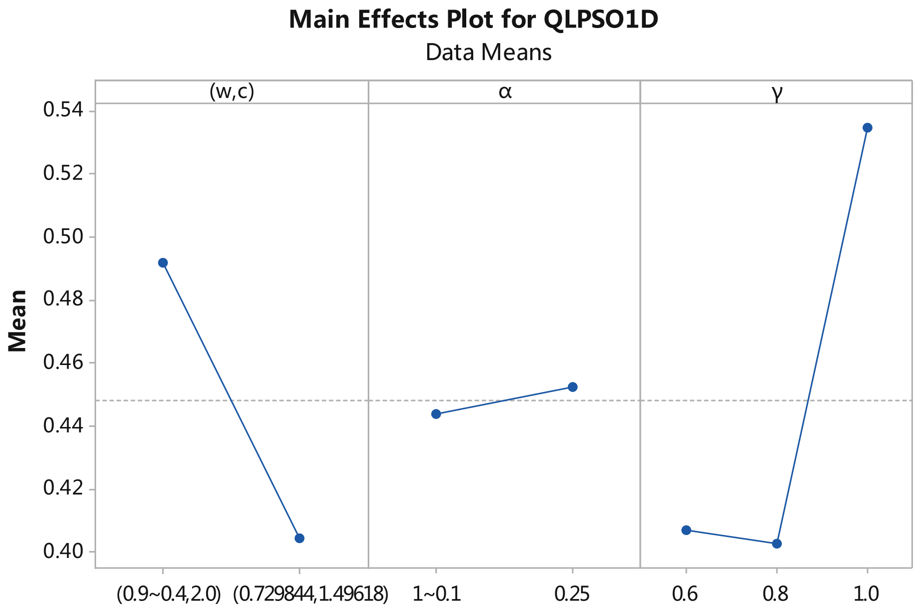


Fig. 7 The main effect plot of parameters for QLPSO$_{1D}$

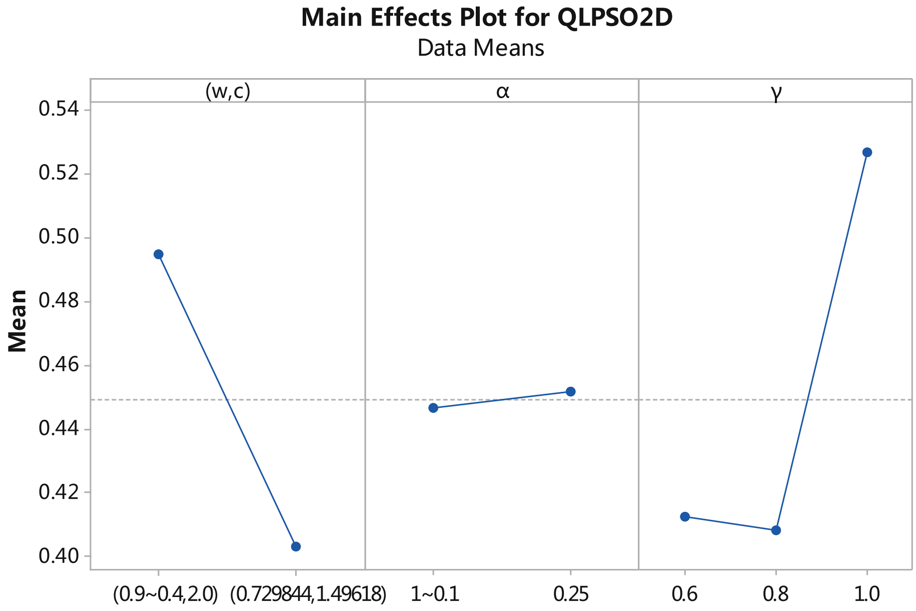


Fig. 8 The main effect plot of parameters for QLPSO$_{2D}$

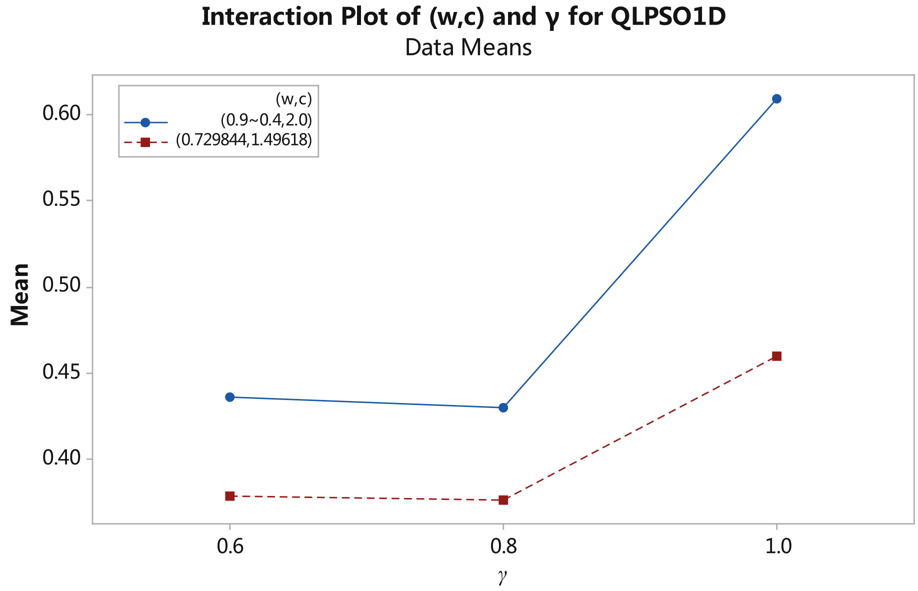


Fig. 9 The interaction plot for (w,c) and γ for QLPSO$_{1D}$

Table 5 Convergence accuracy comparison between QLPSO$_{1D}$ and other static topologies

No.	QLPSO$_{1D}$	PSO-L2	PSO-L4	PSO-L8	PSO-L16	PSO-L30
F01	2.22E−13	1.14E−14	3.41E−14	**5.68E−15**	1.71E−14	1.06E−12
F02	1.36E+07	1.42E+07	1.32E+07	**1.31E+07**	1.57E+07	1.73E+07
F03	**5.67E+07**	9.31E+07	1.08E+08	1.01E+08	2.42E+08	9.96E+08
F04	7.51E+03	2.19E+04	1.06E+04	5.39E+03	**4.31E+03**	5.55E+03
F05	1.56E−13	1.14E−13	**1.08E−13**	**1.08E−13**	**1.08E−13**	3.22E−12
F06	8.96E+01	**7.89E+01**	9.05E+01	8.17E+01	9.53E+01	1.01E+02
F07	**3.33E+01**	6.09E+01	4.95E+01	4.41E+01	7.63E+01	1.10E+02
F08	2.10E+01	2.10E+01	2.10E+01	2.10E+01	2.10E+01	**2.09E+01**
F09	**2.35E+01**	2.79E+01	2.60E+01	2.51E+01	2.37E+01	2.72E+01
F10	1.52E−01	1.49E−01	1.37E−01	1.73E−01	**1.35E−01**	1.45E−01
F11	**4.38E+01**	5.27E+01	5.01E+01	4.58E+01	5.26E+01	8.99E+01
F12	6.58E+01	6.50E+01	**5.42E+01**	5.52E+01	6.66E+01	1.22E+02
F13	1.27E+02	1.32E+02	**1.13E+02**	1.15E+02	1.46E+02	2.01E+02
F14	**1.82E+03**	2.43E+03	2.23E+03	2.01E+03	1.87E+03	2.32E+03
F15	6.95E+03	4.30E+03	3.96E+03	**3.83E+03**	3.87E+03	4.21E+03
F16	2.58E+00	2.27E+00	2.60E+00	2.61E+00	2.63E+00	**2.20E+00**
F17	**8.37E+01**	1.05E+02	9.30E+01	8.74E+01	9.43E+01	1.24E+02
F18	2.08E+02	1.56E+02	1.38E+02	**1.30E+02**	1.36E+02	1.53E+02
F19	**4.11E+00**	5.67E+00	4.64E+00	4.31E+00	4.98E+00	6.86E+00
F20	1.19E+01	1.21E+01	1.19E+01	**1.18E+01**	1.22E+01	1.48E+01
F21	2.92E+02	**2.61E+02**	2.70E+02	3.07E+02	3.11E+02	2.98E+02
F22	**1.98E+03**	2.81E+03	2.50E+03	2.19E+03	2.00E+03	2.66E+03
F23	6.27E+03	4.59E+03	**3.90E+03**	4.02E+03	4.40E+03	4.75E+03
F24	**2.62E+02**	2.75E+02	2.72E+02	2.69E+02	2.71E+02	2.85E+02
F25	**2.86E+02**	2.95E+02	2.93E+02	2.92E+02	2.92E+02	3.04E+02
F26	3.01E+02	**2.47E+02**	2.58E+02	3.09E+02	3.20E+02	3.44E+02
F27	**8.83E+02**	1.03E+03	9.84E+02	9.50E+02	9.53E+02	1.01E+03
F28	3.88E+02	**2.95E+02**	3.00E+02	3.30E+02	3.89E+02	4.13E+02

Table 6 Pairwise comparison between QLPSO$_{1D}$ and other static topologies

QLPSO$_{1D}$ vs	PSO-L2	PSO-L4	PSO-L8	PSO-L16	PSO-L30
Wins	15	14	15	20	21
Draws	1	2	1	1	0
Losses	12	12	12	7	7
Wilcoxon *p*-value	**1.03E−02**	3.25E−01	**3.00E−02**	**8.63E−16**	**1.30E−24**

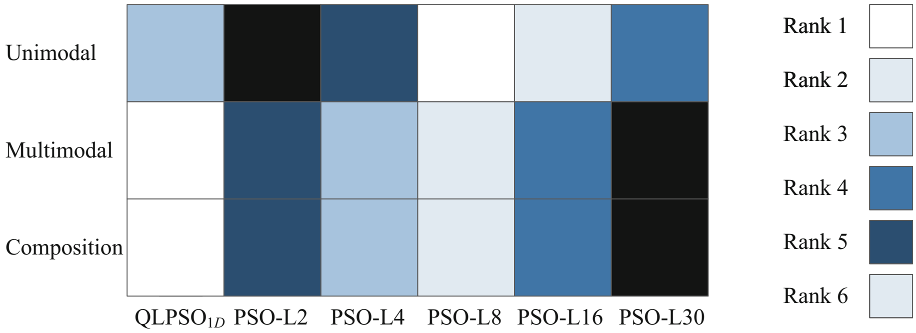


Fig. 10 Multiple comparison between QLPSO$_{1D}$ and other static topologies

Table 7 Convergence accuracy comparison between QLPSO$_{2D}$ and other static topologies

No.	QLPSO$_{2D}$	PSO-S4	PSO-S8	PSO-S12	PSO-S24
F01	2.18E−13	**1.18E−13**	1.96E−13	2.18E−13	1.46E−13
F02	**1.16E+07**	1.24E+07	1.20E+07	1.46E+07	1.40E+07
F03	5.40E+07	**3.48E+07**	9.13E+07	1.46E+08	3.79E+08
F04	6.37E+03	7.50E+03	5.01E+03	5.59E+03	**3.96E+03**
F05	1.32E−13	**1.14E−13**	**1.14E−13**	1.27E−13	**1.14E−13**
F06	8.35E+01	**8.07E+01**	8.41E+01	8.28E+01	8.43E+01
F07	**3.27E+01**	4.19E+01	4.86E+01	5.09E+01	6.78E+01
F08	**2.10E+01**	**2.10E+01**	**2.10E+01**	**2.10E+01**	**2.10E+01**
F09	**2.20E+01**	2.59E+01	2.39E+01	2.63E+01	2.29E+01
F10	**1.28E−01**	1.48E−01	1.44E−01	1.47E−01	1.32E−01
F11	4.44E+01	**4.27E+01**	4.90E+01	5.33E+01	6.06E+01
F12	5.50E+01	**5.34E+01**	6.18E+01	6.42E+01	7.31E+01
F13	1.30E+02	**1.10E+02**	1.29E+02	1.28E+02	1.50E+02
F14	**1.84E+03**	1.96E+03	1.88E+03	2.06E+03	2.11E+03
F15	5.83E+03	4.07E+03	3.85E+03	4.10E+03	**3.82E+03**
F16	2.54E+00	2.57E+00	2.67E+00	**2.23E+00**	2.65E+00
F17	**8.66E+01**	8.69E+01	8.95E+01	9.50E+01	8.84E+01
F18	1.73E+02	1.55E+02	1.40E+02	1.46E+02	**1.23E+02**
F19	**4.08E+00**	4.25E+00	4.32E+00	4.62E+00	4.62E+00
F20	**1.16E+01**	**1.16E+01**	**1.16E+01**	1.20E+01	1.22E+01
F21	3.10E+02	3.20E+02	3.00E+02	2.98E+02	**2.87E+02**
F22	**1.89E+03**	2.29E+03	2.10E+03	2.35E+03	2.22E+03
F23	5.74E+03	**3.95E+03**	4.03E+03	4.19E+03	4.55E+03
F24	**2.63E+02**	2.66E+02	2.69E+02	2.71E+02	2.76E+02
F25	**2.85E+02**	2.87E+02	**2.85E+02**	2.87E+02	2.94E+02
F26	**2.95E+02**	2.98E+02	3.21E+02	3.18E+02	3.35E+02
F27	**9.03E+02**	9.38E+02	9.23E+02	9.46E+02	9.39E+02
F28	**3.23E+02**	3.25E+02	3.50E+02	**3.23E+02**	4.72E+02

QLPSO$_{1D}$ achieves significantly better performance than PSOs with static ring topologies.

In addition, for a more comprehensive statistical analysis, this paper carries out nonparametric statistical tests, including the pairwise comparison and multiple comparison. Table 6 presents the results of the pairwise comparison including the number of wins, draws, and losses and *p*-value of Wilcoxon test. According to the Sign test in [53], if the number of wins is at least $n/2 + 1.96 \cdot \sqrt{n}/2 = 19.18$ (n is the number of problems), then the algorithm is significantly better with $p < 0.05$. Hence, QLPSO$_{1D}$ is considered better than PSO-L16 and PSO-L30 with a level of significance $\alpha = 0.05$. According to the results of Wilcoxon test, QLPSO$_{1D}$ is considered better than all algorithms with a level of significance $\alpha = 0.05$, except PSO-L4. For a multiple comparison, Friedman test is performed and the average ranking of the six algorithms is summarized in Fig. 10. As shown in Fig. 10, QLPSO$_{1D}$ has the best average rankings on multimodal and composition problems, while PSO-L8 outperforms unimodal functions. These results verify the efficiency of QLPSO$_{1D}$ when dealing with complex test functions.

Table 8 Pairwise comparison between QLPSO$_{2D}$ and other static topologies

QLPSO$_{2D}$ vs	PSO-S4	PSO-S8	PSO-S12	PSO-S24
Wins	16	17	16	20
Draws	2	3	3	1
Losses	10	8	9	7
Wilcoxon *p*-value	1.11E−01	5.37E−01	1.09E−01	**2.93E−02**

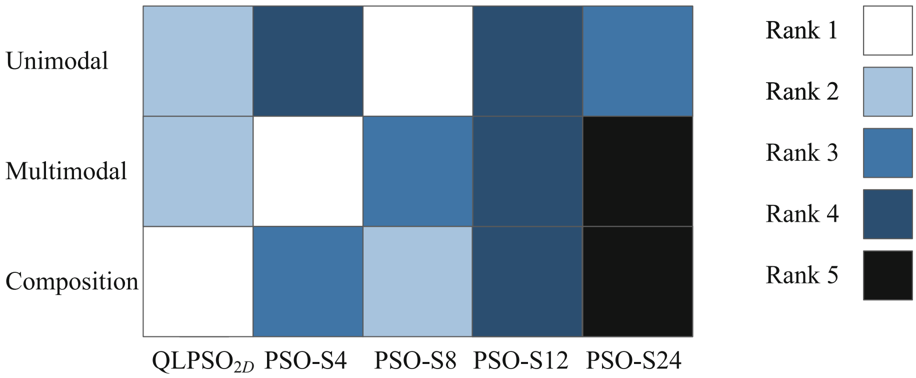


Fig. 11 Multiple comparison between QLPSO$_{2D}$ and other static topologies

Table 9 Convergence accuracy comparison between QLPSOs and other dynamic topologies

No.	QLPSO$_{1D}$	QLPSO$_{2D}$	PSO-ISN	PSO-FS	PSO-S	PSO-DBN	PSO-DC
F01	2.22E−13	2.18E−13	**4.09E−14**	2.59E+00	1.28E+04	2.46E−13	1.11E+04
F02	1.36E+07	**1.16E+07**	1.42E+07	4.54E+07	1.54E+08	1.23E+07	1.39E+08
F03	5.67E+07	**5.40E+07**	1.08E+08	7.16E+09	4.75E+10	3.31E+08	3.17E+10
F04	7.51E+03	6.37E+03	**3.47E+03**	2.51E+04	5.38E+04	1.98E+04	3.84E+04
F05	1.56E−13	1.32E−13	**1.11E−13**	1.02E+00	3.72E+03	2.22E−04	2.50E+03
F06	8.96E+01	8.35E+01	**7.16E+01**	1.13E+02	7.25E+02	1.02E+02	6.18E+02
F07	3.33E+01	**3.27E+01**	5.37E+01	1.06E+02	1.87E+02	5.19E+01	1.58E+02
F08	**2.10E+01**	**2.10E+01**	**2.10E+01**	**2.10E+01**	**2.10E+01**	**2.10E+01**	**2.10E+01**
F09	2.35E+01	**2.20E+01**	2.79E+01	2.70E+01	3.94E+01	2.26E+01	4.02E+01
F10	1.52E−01	**1.28E−01**	1.55E−01	3.15E+00	1.46E+03	1.53E−01	1.24E+03
F11	**4.38E+01**	4.44E+01	5.59E+01	6.97E+01	3.76E+02	9.15E+01	3.63E+02
F12	6.58E+01	**5.50E+01**	7.14E+01	1.10E+02	3.69E+02	2.13E+02	3.57E+02
F13	**1.27E+02**	1.30E+02	1.35E+02	2.02E+02	3.70E+02	2.30E+02	3.58E+02
F14	**1.82E+03**	1.84E+03	2.35E+03	3.09E+03	7.47E+03	1.88E+03	7.42E+03
F15	6.95E+03	5.83E+03	**4.05E+03**	5.08E+03	7.54E+03	4.81E+03	7.43E+03
F16	2.58E+00	**2.54E+00**	2.65E+00	2.67E+00	2.66E+00	2.61E+00	2.73E+00
F17	**8.37E+01**	8.66E+01	9.76E+01	1.37E+02	8.39E+02	2.92E+02	7.41E+02
F18	2.08E+02	1.73E+02	**1.51E+02**	1.59E+02	8.39E+02	3.16E+02	7.73E+02
F19	4.11E+00	**4.08E+00**	5.30E+00	1.11E+01	2.36E+03	1.83E+01	1.81E+03
F20	1.19E+01	**1.16E+01**	1.18E+01	1.49E+01	1.50E+01	1.42E+01	1.46E+01
F21	2.92E+02	3.10E+02	2.66E+02	3.30E+02	3.33E+03	**2.40E+02**	2.70E+03
F22	1.98E+03	**1.89E+03**	2.77E+03	3.46E+03	7.68E+03	2.05E+03	7.59E+03
F23	6.27E+03	5.74E+03	**4.57E+03**	5.92E+03	7.76E+03	5.22E+03	7.48E+03
F24	**2.62E+02**	2.63E+02	2.73E+02	3.08E+02	3.04E+02	2.68E+02	3.02E+02
F25	2.86E+02	**2.85E+02**	2.98E+02	3.31E+02	3.15E+02	2.98E+02	3.11E+02
F26	3.01E+02	2.95E+02	2.54E+02	3.57E+02	**2.21E+02**	2.30E+02	2.68E+02
F27	**8.83E+02**	9.03E+02	1.03E+03	1.19E+03	1.29E+03	9.55E+02	1.30E+03
F28	3.88E+02	3.23E+02	**3.00E+02**	6.08E+02	3.73E+03	3.07E+02	2.94E+03

Table 10 Pairwise comparison between QLPSO$_{1D}$ and other dynamic topologies

QLPSO$_{1D}$ vs	PSO-ISN	PSO-FS	PSO-S	PSO-DBN	PSO-DC
Wins	16	24	26	20	26
Draws	1	1	1	1	1
Losses	11	3	1	7	1
Wilcoxon p-value	9.30E−01	**4.62E−25**	**4.65E−270**	**2.75E−06**	**9.14E−247**

Table 11 Pairwise comparison between QLPSO$_{2D}$ and other dynamic topologies

QLPSO$_{2D}$ vs	PSO-ISN	PSO-FS	PSO-S	PSO-DBN	PSO-DC
Wins	17	25	26	22	26
Draws	1	1	1	1	1
Losses	10	2	1	5	1
Wilcoxon p-value	2.14E−01	**9.76E−31**	**6.19E−278**	**1.84E−09**	**1.96E−257**

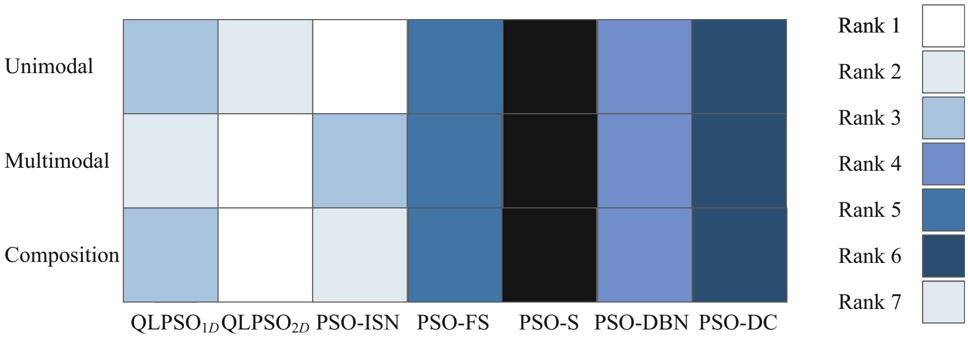


Fig. 12 Multiple comparison between QLPSO$_{1D}$, QLPSO$_{2D}$, and other static topologies

4.3.2 In two-dimensional square topology

This paper also conducts experiments to evaluate the performance of QLPSO with a two-dimensional square topology, named as QLPSO$_{2D}$. In the first trail, QLPSO$_{2D}$ is executed. In the following trails, PSO is conducted with S4 (von Neumann), S8 (Moore), S12 (combination), and S24 (extension), separately. Table 7 lists the mean errors of these trails (the best results are in bold).

It is clear from Table 7 that PSOs with different Sk grid topologies obtain different results on CEC13 functions. Von Neumann seems a better topology for unimodal problems because PSO-S4 obtains the best results on F01, F03, and F05. QLPSO$_{2D}$ achieves the best results on one unimodal function F02. When encountering multimodal and composition problems, 14 out of 23 mean values obtained by QLPSO$_{2D}$ are the best, i.e., F07, F08, F09, F10, F14, F17, F19, F20, F22, F24, F25, F26, F27, and F28. In these problems, the result of F08 is the same as that obtained by all other algorithms, the result of F20 is the same as that obtained by PSO-S4 and PSO-S8, the result of F25 is the same as that obtained by PSO-S8, and the result of F28 is the same as that obtained by PSO-S12. Hence, the comparison with other static PSOs using different grid topologies implies the superiority of QLPSO$_{2D}$ on CEC13 functions.

Table 8 lists the results of pairwise comparison between QLPSO$_{2D}$ and other static topologies. As shown in Table 8, QLPSO$_{2D}$ is better than PSO-S24 with a level of significance $\alpha = 0.05$. Besides, Fig. 11 describes the average ranking of these algorithms for a multiple comparison. QLPSO$_{2D}$ obtains the second average rankings on unimodal and multimodal functions and gets the best average ranking on composition functions. Overall, QLPSO$_{2D}$ has an excellent performance on all test functions.

4.4 Experiment 2: comparing to dynamic topologies

To further verify the performance of the new QLPSO, the second experiment compares it with several state-of-the-art variants of PSO with dynamic topology strategies. For a fair comparison, the velocity and position vectors are all updated using (1) and (2), separately. No addition operators are implemented in the comparing algorithms, such as local search. In addition, each algorithm is run 50 times for each instance. The common parameters of these algorithms are the same as that of QLPSO, as shown in Sect. 4.2. The brief description and other parameter settings are as follows.

- PSO with increasing the size of neighborhood (PSO-ISN) [34]: It is the most common way to changing the size of neighborhood. The neighborhood of the particle starts from L2, then experiences L4, L8,..., L28, and ends with L30.
- PSO with fitness sharing (PSO-FS) [39, 40]: The parameters are set as follows: the niching radius $\sigma_{\text{share}} = 2$, scaling factor $\alpha = 2$, which are also suggested in [39].
- PSO with speciation (PSO-S) [42]: In the PSO-MS, the swarm is divided into multi-species according to species radius, and the particles in the same specie share information to explore the search region. The specie radius $\sigma = 2.0$.
- PSO with distance-based neighborhood (PSO-DBN) [45]: The dynamic neighborhood size is linearly increased from 2 to 5 over the function evaluations.

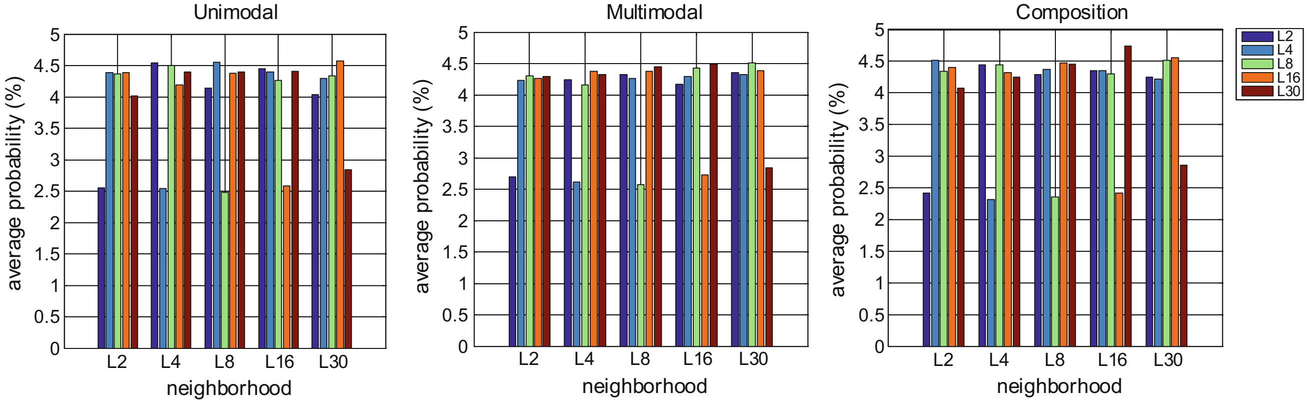


Fig. 13 The average probability of calls obtained by QLPSO$_{1D}$

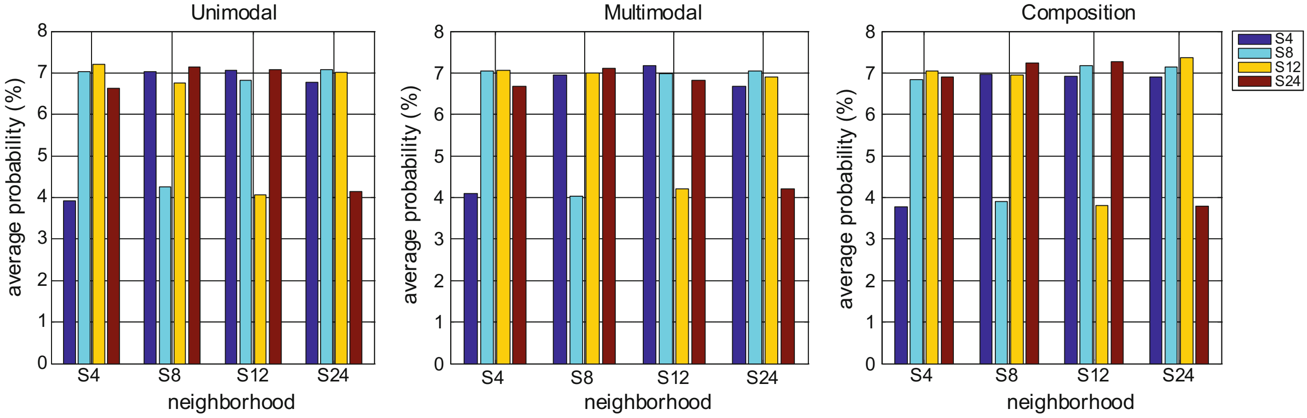
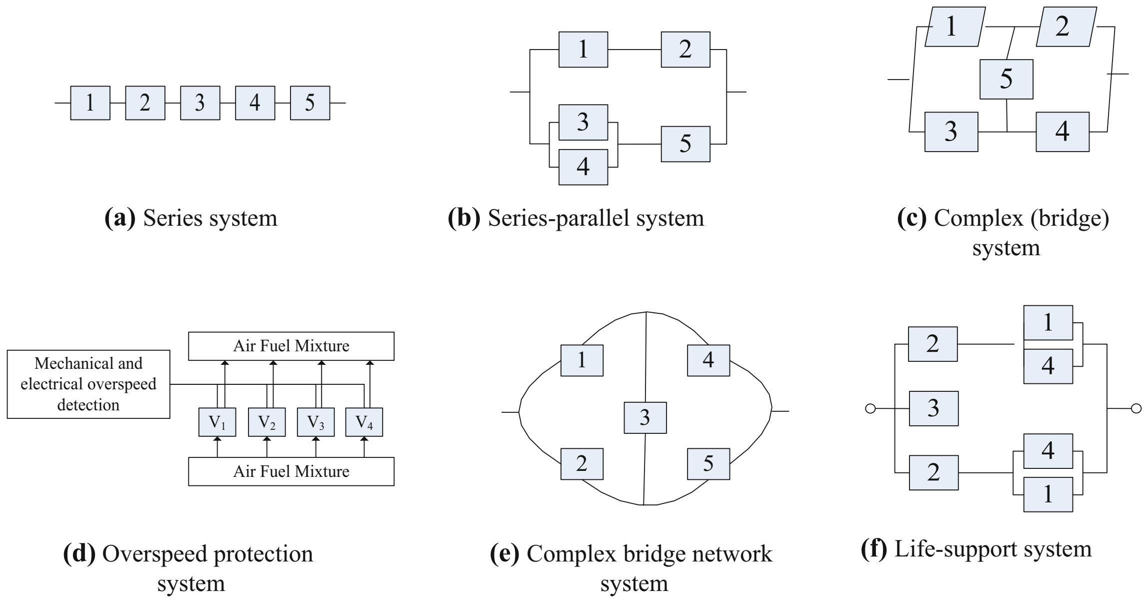


Fig. 14 The average probability of calls obtained by QLPSO$_{2D}$



(a) Series system

(b) Series-parallel system

(c) Complex (bridge) system

(d) Overspeed protection system

(e) Complex bridge network system

(f) Life-support system

Fig. 15 Different kinds of systems

Table 12 The comparative results for the series system (maximize R_S)

Method	(n_1,n_2,n_3,n_4,n_5)	r_1	r_2	r_3	r_4	r_5
ABC [56]	(3,2,2,3,3)	0.779403565208	0.87183320141	0.902886411643	0.711398061305	0.787808548579
IA [57]	(3,2,2,3,3)	0.779462304	0.871883456	0.902800879	0.711350168	0.787861587
INGHS [58]	(3,2,2,3,3)	0.7793988710	0.8718370210	0.9028853550	0.7114025151	0.787799488032
PSO [59]	(2,3,2,4,2)	0.80059281	0.74049316	0.82914384	0.63686144	0.88704276
SSO [59]	(3,2,2,3,3)	0.78271484	0.8735199	0.90264893	0.71313477	0.77729797
PSSO [59]	(3,2,2,3,3)	0.77946645	0.87173278	0.90284951	0.71148780	0.78781644
QLPSO$_{1D}$	(3,2,2,3,3)	0.7791424524	0.8720497875	0.9028678356	0.7113074160	0.7878724978
QLPSO$_{2D}$	(3,2,2,3,3)	0.7794385446	0.8718371372	0.9028738661	0.7114083570	0.7877618267
Method	R_S	Slack(g_1)	Slack(g_2)	Slack(g_3)	SD	NFE
ABC [56]	**0.931682387672**	27	2.258957E−10	7.518918241	2.37E−08	20,000
IA [57]	0.931682340	27	5.284E−07	7.518918	1.3E−14	20,000
INGHS [58]	0.931682387723	27	3.118E−07	7.5189182411	4.56E−03	150,000
PSO [59]	0.8885037	4	1.6775727E+01	4.8146147	–	100,000
SSO [59]	0.93150199	27	1.82140E−03	7.51891824	–	100,000
PSSO [59]	0.93168230	27	4.9081966E−05	7.51891824	2.67E−02	100,000
QLPSO$_{1D}$	0.9316820871	27	4.753702427e−05	7.518918241	8.02e−07	50,000
QLPSO$_{2D}$	0.9316823778	27	9.152836356e−06	7.5189182411	1.96e−07	50,000

- PSO with dynamic cluster (PSO-DC) [54]: It is a dynamic topology, based on a combination of two topologies: four-cluster and fitness.

The means errors obtained by QLPSOs and PSOs with other dynamic topologies are listed in Table 9, and the best results are also shown in bold. As is obvious, all algorithms obtain the same result on the function F08. QLPSO$_{1D}$ obtains the best results on 6 functions (F11, F13, F14, F17, F24, and F27), QLPSO$_{2D}$ performs better on 11 functions (F02, F03, F07, F09, F10, F12, F16, F19, F20, F22, and F25). In brief, 17 out of 28 mean values obtained by QLPSOs are the best. The second best algorithm is PSO-ISN, which performs better on 8 out of 28 mean values, and then are PSO-S and PSO-DBN, getting the best result on 1 out of 28 values, respectively.

Table 13 The comparative results for the series–parallel system (maximize R_S)

Method	(n_1,n_2,n_3,n_4,n_5)	r_1	r_2	r_3	r_4	r_5
ABC [56]	(2,2,2,2,4)	0.819737753469	0.844991099776	0.895529543820	0.895433687206	0.868434824469
IA [57]	(2,2,2,2,4)	0.819591561	0.844951068	0.895428548	0.895522339	0.868490229
INGHS [58]	(2,2,2,2,4)	0.8198118626	0.8449506842	0.8956701585	0.8952327069	0.868438057445
PSO [59]	(4,3,2,1,2)	0.84025282	0.88865099	0.62375055	0.93984950	0.75158691
SSO [59]	(2,2,2,2,4)	0.81385803	0.83912659	0.89366150	0.89845276	0.87106323
PSSO [59]	(2,2,2,2,4)	0.81958939	0.84458412	0.89534134	0.89581626	0.86852902
QLPSO$_{1D}$	(2,2,2,2,4)	0.81980097	0.84490165	0.89583336	0.89530991	0.86839295
QLPSO$_{2D}$	(2,2,2,2,4)	0.8196660524	0.8449988154	0.8954862237	0.8955262459	0.8684414181
Method	R_S	Slack(g_1)	Slack(g_2)	Slack(g_3)	SD	NFE
ABC [56]	0.9999766491	40	1.39152E−10	1.609288966	3.18E−11	20,000
IA [57]	0.9999766490	40	5.984537E−08	1.6092889667	3.0E−21	20,000
INGHS [58]	0.9999766489	40	5.305414E−5	1.6092889667	1.41E−05	75,000
PSO [59]	0.99985845	68	0.916915088	4.017703641	–	100,000
SSO [59]	0.99997657	40	2.4 E-03	1.609288966	–	100,000
PSSO [59]	0.9999766487	40	8.117946E−05	1.609288966	2.32E−04	100,000
QLPSO$_{1D}$	0.9999766488	40	4.2683221e−05	1.609288967	1.28-09	50,000
QLPSO$_{2D}$	**0.9999766491**	40	3.20E−09	1.609288967	6.62E−10	50,000

Table 14 The comparative results for the complex (bridge) system (maximize R_S)

Method	(n_1,n_2,n_3,n_4,n_5)	r_1	r_2	r_3	r_4	r_5
ABC [56]	(3,3,2,4,1)	0.8279702762	0.85787475859	0.91418640423	0.64835538681	0.70357531105
IA [57]	(3,3,3,3,1)	0.816624176	0.868767396	0.858748781	0.710279379	0.753429200
INGHS [58]	(3,3,2,4,1)	0.8279847911	0.8576796813	0.9141564522	0.6484814055	0.7048654988
PSO [59]	(3,3,2,2,3)	0.77061588	0.90109253	0.89278651	0.60083008	0.73451002
SSO [59]	(3,3,2,4,1)	0.82008362	0.85119629	0.91854858	0.66072083	0.70275879
PSSO [59]	(3,3,2,4,1)	0.82783292	0.85771241	0.91437458	0.64861002	0.70287554
QLPSO$_{1D}$	(3,3,2,4,1)	0.8282158643	0.8577921230	0.9142554735	0.6478566811	0.7049914734
QLPSO$_{2D}$	(3,3,2,4,1)	0.8280914450	0.8578525657	0.9142393070	0.6479694262	0.7049731448

Method	R_S	Slack(g_1)	Slack(g_2)	Slack(g_3)	SD	NFE
ABC [56]	0.99988963581	5	3.74636763E−04	1.560466288	8.67E−09	20,000
IA [57]	0.9998893505	18	4.0420871E−08	4.264770	4.0E−20	20,000
INGHS [58]	0.9998896364	5	1.8985E−06	1.560466288	5.19E−05	75,000
PSO [59]	0.99967140	37	1.654571312E+01	1.411803224	–	100,000
SSO [59]	0.99988862	5	4.37599E−03	1.560466288	–	100,000
PSSO [59]	0.999889635738	5	2.50290205E−05	1.560466288	5.02E−04	100,000
QLPSO$_{1D}$	0.9998896361	5	2.68371801E−04	1.5604662880	3.32e−09	50,000
QLPSO$_{2D}$	**0.9998896371**	5	4.23172141E−05	1.5604662880	5.16e−09	50,000

Furthermore, nonparametric statistical tests are also carried out to investigate the performance of QLPSO, comparing with several state-of-the-art PSOs with dynamic topologies. Tables 10 and 11 are pairwise comparisons between PSOs with dynamic topologies. It is clear that QLPSO$_{1D}$ and QLPSO$_{2D}$ are significantly better than PSO-FS, PSO-S, PSO-DBN, and PSO-DC with a level of significance $\alpha = 0.05$. In addition, Fig. 12 depicts the results (average ranking) of the multiple comparison. It is obvious that QLPSO$_{2D}$ obtains the second ranking on unimodal functions and the best rankings on multimodal and composition problems. From these results, it can be concluded that QLPSO$_{2D}$ behaves better than PSOs with other dynamic topologies at most cases.

Table 15 The comparative results for the overspeed protection system (maximize R_S)

Method	(n_1,n_2,n_3,n_4)	r_1	r_2	r_3	r_4
ABC [56]	(5,5,4,6)	0.901626809561	0.888208355883	0.948134377884	0.849942135673
IA [57]	(5,5,4,6)	0.901588628	0.888192380	0.948166022	0.849969792
INGHS [58]	(5,5,4,6)	0.9015565830	0.8882438856	0.9481110971	0.8499817375
PSO [59]	(4,6,5,5)	0.92952331	0.81370356	0.88663747	0.89987183
SSO [59]	(5,6,4,5)	0.90208435	0.85472107	0.94606018	0.88633728
PSSO [59]	(5,5,4,6)	0.90166461	0.88817296	0.94821033	0.84987084
QLPSO$_{1D}$	(5,6,4,5)	0.901584353	0.849956118	0.94813963	0.88822184
QLPSO$_{2D}$	(5,6,4,5)	0.901608249	0.849919449	0.948135432	0.88823203

Method	R_S	Slack(g_1)	Slack(g_2)	Slack(g_3)	SD	NFE
ABC [56]	0.999954674663	55	5.57304247E−9	15.3634630874	3.39E−11	20,000
IA [57]	0.999954674554580	55	1.24953721E−04	15.363463087	4.14E−18	18,000
INGHS [58]	0.9999546745	55	5.05411E−5	24.8018827221	1.52E−05	75,000
PSO [59]	0.99990474	37	1.15265677E+01	1.16447077E+01	–	100,000
SSO [59]	0.99995416	55	0.109233104	2.48018827E+01	–	100,000
PSSO [59]	0.99995467	55	3.2872253E−05	15.36346308	–	100,000
QLPSO$_{1D}$	0.9999546746	55	1.0208235E−04	24.801882722	2.88E−10	50,000
QLPSO$_{2D}$	**0.9999546747**	55	9.0050492E−08	24.801882722	1.03E−10	50,000

Table 16 The comparative results for the complex bridge network system (minimize C_S)

Method	r_1	r_2	r_3	r_4	r_5	C_S	R_S	SD	NFE
ABC [56]	0.93545253	0.93438965	0.79041103	0.93541044	0.93448154	5.01991875	0.99	7.51E−06	20,000
FGO [60]	0.93635	0.93869	0.80615	0.93512	0.93476	5.02042	0.9905	–	–
I-NESA [61]	0.93747	0.93291	0.78485	0.93641	0.93342	5.01993	0.99	–	100,000
$QLPSO_{1D}$	0.934972115139146	0.934805471249050	0.792291343103702	0.934943510327322	0.934837855908337	5.019918194790 92	0.990000058300484	2.06E−07	50,000
$QLPSO_{2D}$	0.934851305422408	0.934884720019623	0.791820866715045	0.934938126584908	0.934930205684251	**5.01991813860112**	0.990000002971920	8.34E−08	50,000

4.5 Experiment 3: Analyzing QLPSO of different dimensions

4.5.1 Comparison between $QLPSO_{1D}$ and $QLPSO_{2D}$

The above results show the excellent convergence accuracy of integrating QL to PSO. The QL helps each particle in the swarm independently learn from its own Q-table and select the best number of neighborhood. It makes the PSO a problem-free algorithm.

As we know, the swarm can be constructed and assigned into two topology structures: a one-dimensional ring and a two-dimensional square. Although both topologies are conducive to the performance of PSOs, this paper tries to seek which is the optimal topology when embedding QL. To tackle this problem, pairwise comparison [53], a non-parametric statistical test, is carried out in this paper.

As shown in Table 9, $QLPSO_{2D}$, compared with $QLPSO_{1D}$, is carried out on 28 CEC13 functions, won 20, drawn 1, and lost 7. According to the Sign test, $QLPSO_{2D}$ is considered better than $QLPSO_{1D}$ with a level of significance $\alpha = 0.05$. This finding is consistent with the earlier recommendation of von Neumann configuration in [25]. According to our analysis, a possible reason may be that the two-dimensional square topology contains more effective information than the one-dimensional ring topology.

4.5.2 The average probability of calls

For further analysis of the different dimensions, the average probability of calls pertaining to each neighborhood is computed in this paper, as illustrated in Figs. 13 and 14. This paper simplifies 28 basic functions to three types by averaging.

As shown in Figs. 13 and 14, the following results can be found: (1) The average probability of calling the current neighborhood by QL is the lowest no matter the property of the problem. It is mainly because using the current neighborhood is not conducive to searching other regions more broadly, which may trap the algorithm into local optimum. (2) Furthermore, suppose an extreme neighborhood as the current neighborhood, the average probability of calling another extreme neighborhood is lower than that of calling other neighborhoods (except the current one). For example, when using $QLPSO_{2D}$ and the current neighborhood is S24, there is no possibility of selecting the neighborhood S4. The situation is similar in $QLPSO_{1D}$. A possible inference is that if an optimal region has been found by the current extreme neighborhood, the selection of another extreme neighborhood would obtain a bad solution.

Table 17 The comparative results for the life-support system in a space capsule (minimize C_S)

Method	r_1	r_2	r_3	r_4	C_S	R_S	SD	NFE
FGO [60]	0.82569	0.89022	0.62732	0.72865	390.570	0.989999013262[a]	–	–
SA [61]	0.82529	0.89169	0.62161	0.72791	390.6327	0.990003	–	–
I-NESA [61]	0.825160	0.890130	0.62825	0.72917	390.572	0.99	–	100,000
QLPSO$_{\mathrm{1D}}$	0.825673012	0.890103176	0.628361983	0.728384377	390.5711428	0.990000002	2.42E−03	50,000
QLPSO$_{\mathrm{2D}}$	0.82555814	0.890157895	0.627554653	0.728955165	**390.5707588**	0.99	1.30E−03	50,000

[a] Constraint violation

4.6 Experiment 4: Real-world reliability optimization problems

Experiment 4 evaluates the performance of the proposed algorithm on real-world reliability optimization problems. In order to optimize system reliability, three main ways may be followed: increasing the component reliability (reliability allocation), using redundant components in parallel (redundancy allocation), and both of them (reliability–redundancy allocation) [55]. These ways aim to maximize the system reliability or minimize the system cost under some constraints. To evaluate the performance of the proposed SBA on reliability optimization problems, this paper presents six case studies with different kinds of systems, as shown in Fig. 15.

In constrained problems, adding penalty terms to the objective function is an effective approach to avoid constraint violation. Here, it is applied as follows.

$$Fitness = objective + \sum_{i=1}^{M} 1000 \cdot \max(0, g_i) \qquad (12)$$

The results in Tables 12, 13, 14, 15, 16, and 17 show that most studies have received accepted solutions without exceeding the boundaries. The results obtained by QLPSO$_{\mathrm{2D}}$ are better than those obtained by the QLPSO$_{\mathrm{1D}}$ at all cases. It implies that QL topology constructed from a two-dimensional square is more effective to advance the performance of PSO on real-world reliability optimization problems. Besides, the new proposed algorithm, compared with other meta-heuristic methods, can find the maximum reliability or the minimum cost without abandoning any rules at most cases. Hence, numerical results of six reliability optimization problems show the superiority the proposed approach compared with other advanced studies.

5 Conclusions

In this paper, a one-dimensional ring QLPSO and a two-dimensional square QLPSO are proposed, by embedding QL to PSO. Through each generation, the particle acts independently selecting the optimal topology returned by the Q-table. Also, the Q-table is updated by obtaining a balance between the current and previous information. The main challenging issue of this advancement is the measurement of the reward. In this paper, a combination of fitness and diversity is seen as the calculation of the immediate reward. Furthermore, this paper carries out comprehensive experiments to compare the new algorithm with other topologies and investigate the effects of QL topology with different dimensions.

In the experiment section, the proposed algorithm is compared to PSOs with static and dynamic topologies. The results exhibit the superiority of QL topology than all other topologies. QL controls the size of neighborhood different from just increasing it, e.g., PSO-ISN. Hence, when an optimal region is found, the QLPSO explores this search space more carefully with less connection topology. Once trapping into a local optimum, the QLPSO with more connection topology searches wider area so as to speed up the convergence. Massive experiments also imply that QLPSO is a problem-free algorithm. Regardless of the problem property (unimodal, multimodal, or composition), the QLPSO can outperform other algorithms. Moreover, the comparison between QL topology with different dimensions exhibits that the two-dimensional square structure achieves significantly better performance than one-dimensional ring structure. The analysis of the call probability implies that QL has few possibilities to operate as the following: (1) selecting the current neighborhood; (2) choosing another extreme neighborhood.

The future work can be summarized as follows. On the basis of QLPSO, particles based on multi-agent can be more meaningful. Applications to multi-objective real-world optimization problems would have both practical and theoretical benefits.

Acknowledgements This work was partially supported by National Natural Science Foundation of China (U1433116), the Fundamental Research Funds for the Central Universities (NP2017208), and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX19_0202).

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest with any person(s) or organization(s).

Appendix

To show the stability of the proposed algorithm, convergence curves are depicted in Figs. 16 and 17.

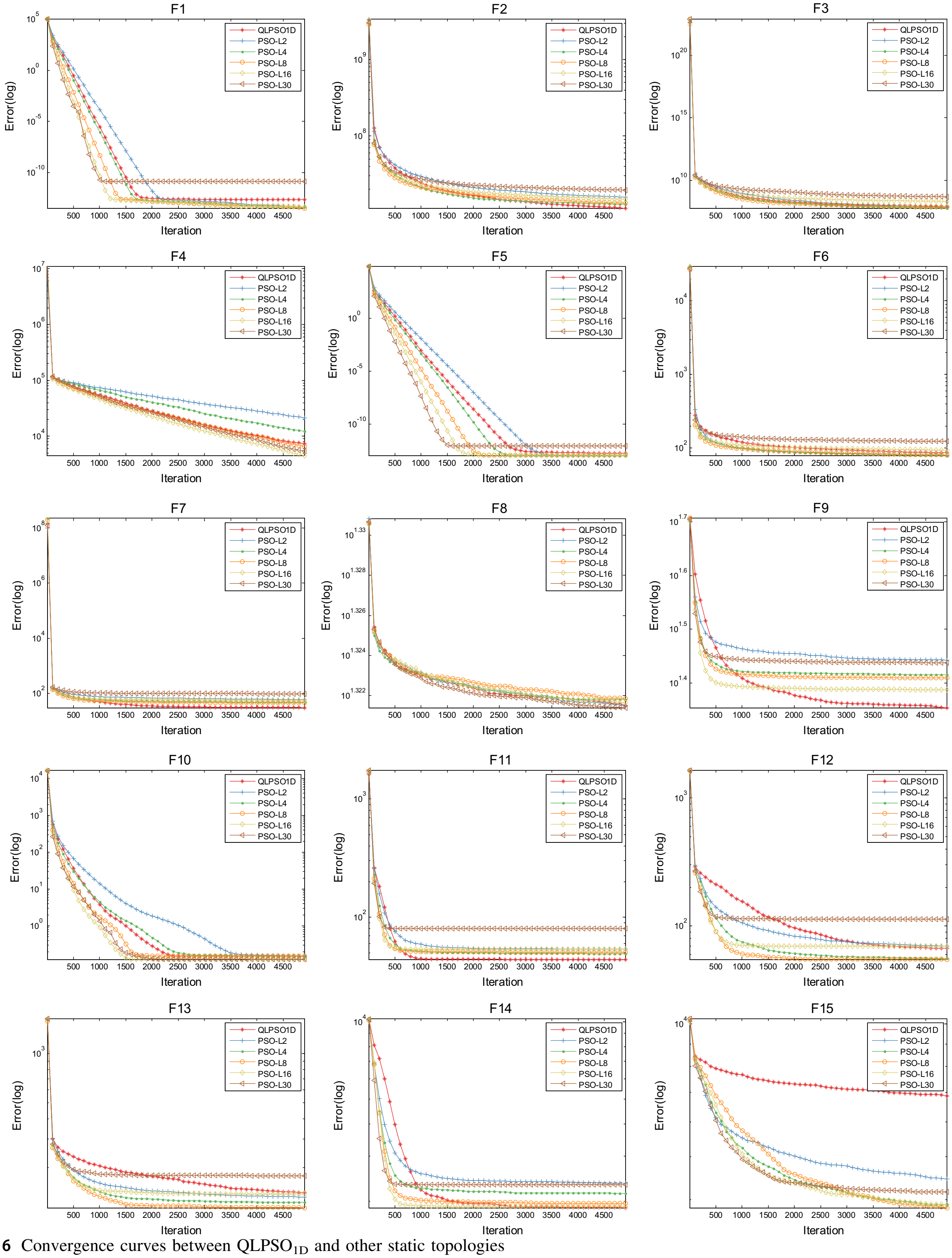


Fig. 16 Convergence curves between QLPSO$_{1D}$ and other static topologies

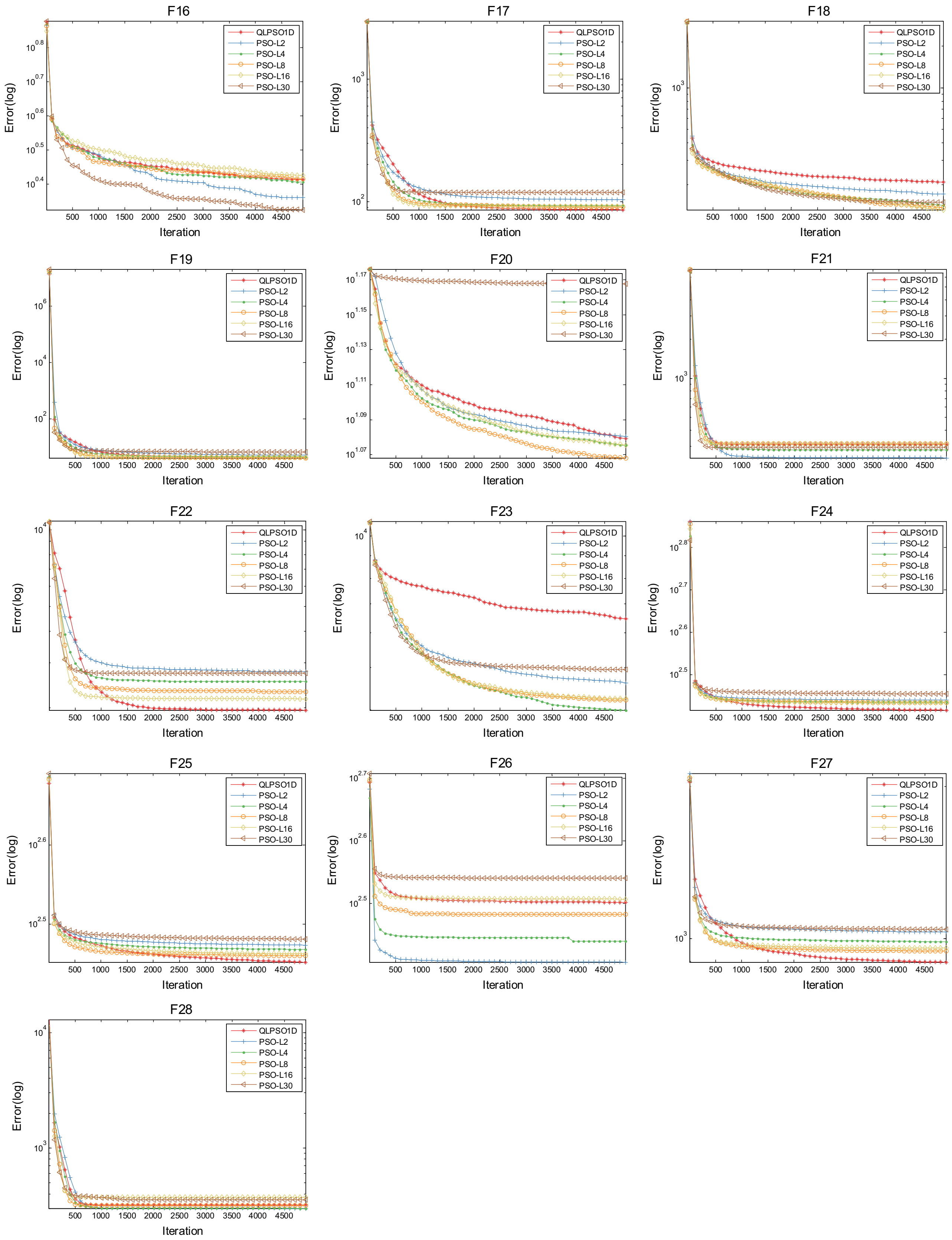


Fig. 16 continued

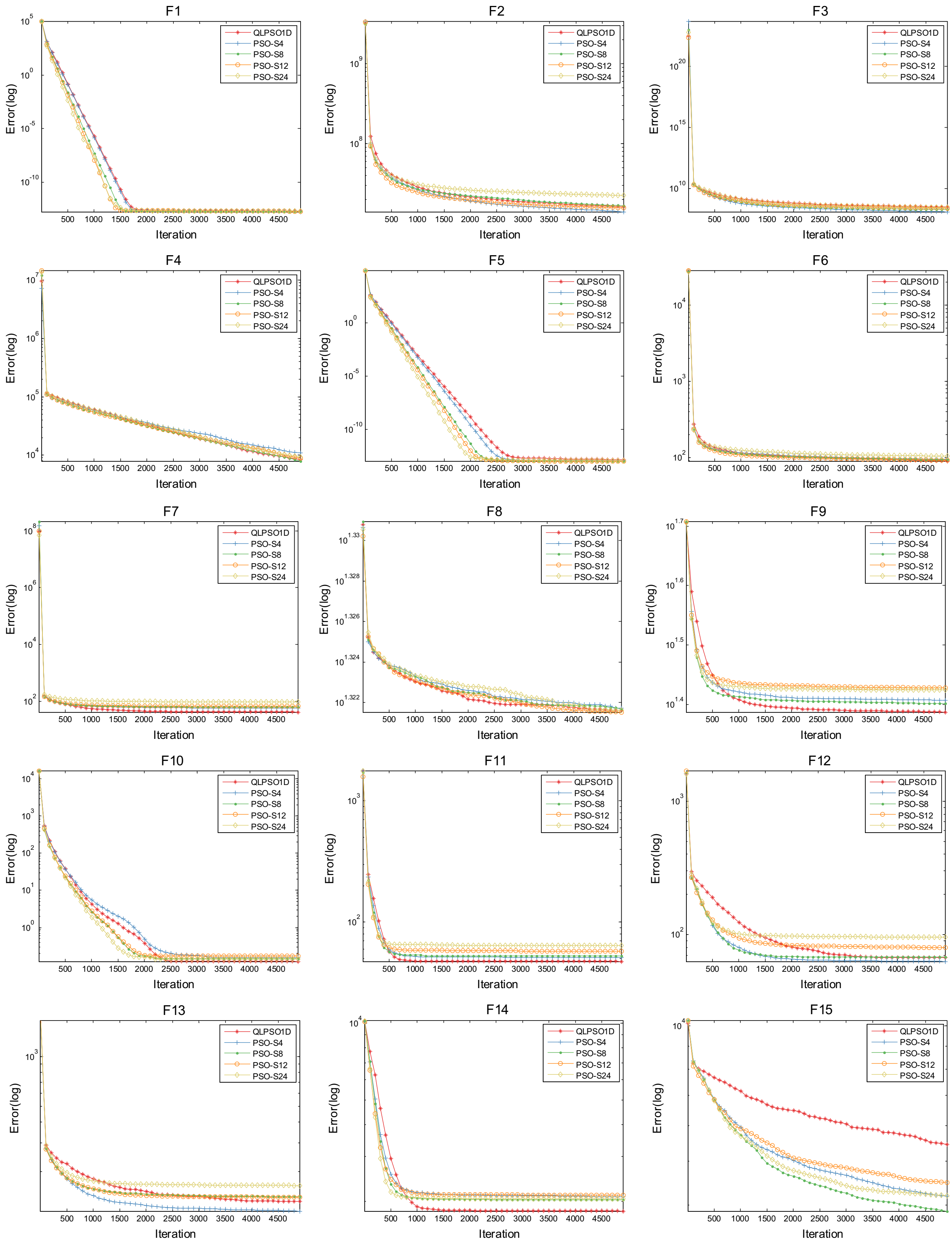


Fig. 17 Convergence curves between QLPSO$_{2D}$ and other static topologies

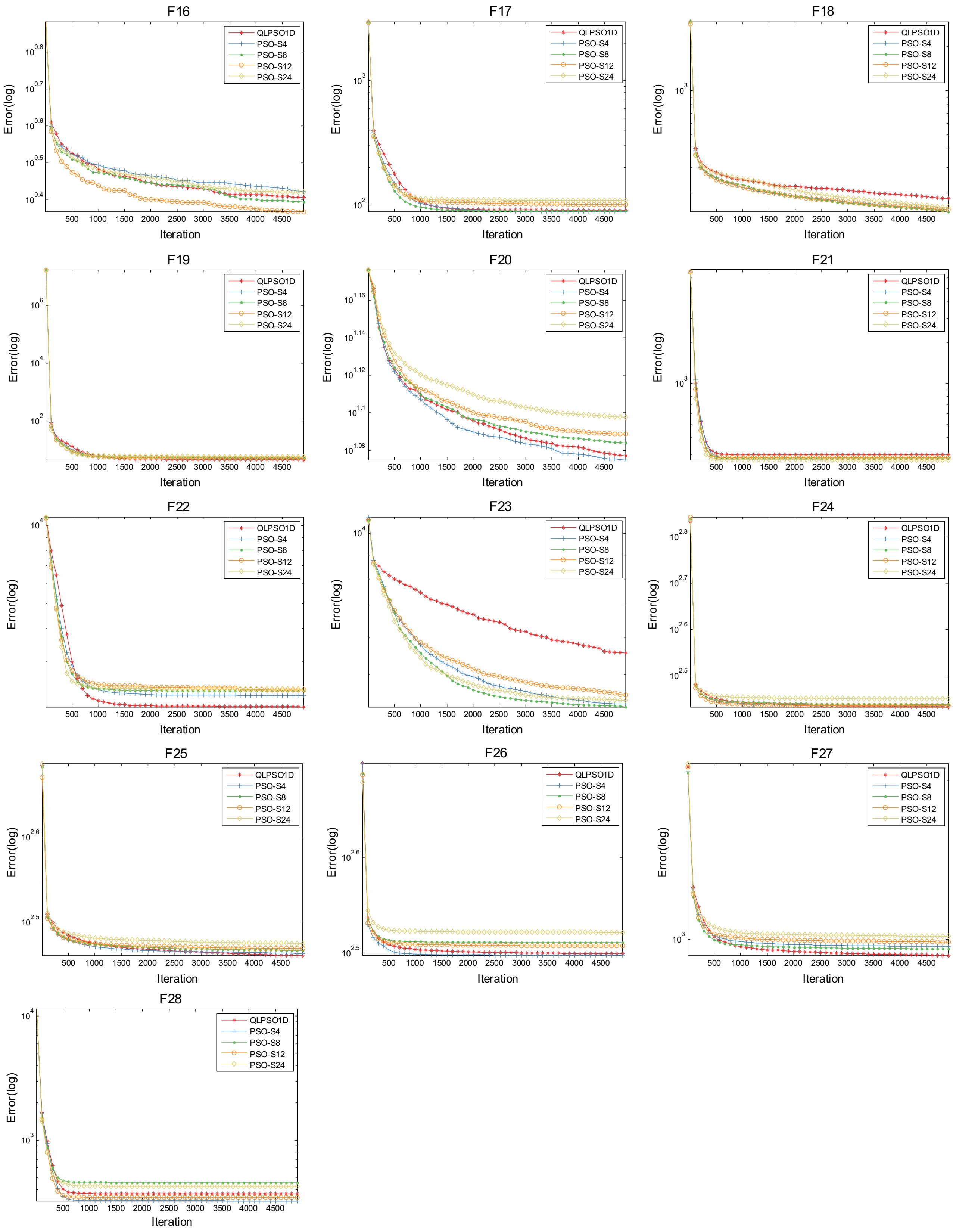


Fig. 17 continued

References

1. Del Ser J, Osaba E, Molina D, Yang X-S, Salcedo-Sanz S, Camacho D, Das S, Suganthan PN, Coello Coello CA, Herrera F (2019) Bio-inspired computation: where we stand and what's next. Swarm Evolut Comput 48:220–250. https://doi.org/10.1016/j.swevo.2019.04.008
2. Zhu Z, Zhou J, Zhen J, Shi YH (2011) DNA sequence compression using adaptive particle swarm optimization-based memetic algorithm. IEEE Trans Evolut Comput 15(5):643–658
3. Arqub OA, Abo-Hammour Z (2014) Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. Inf Sci 279:396–415
4. Arqub OA (2015) Adaptation of reproducing kernel algorithm for solving fuzzy Fredholm–Volterra integrodifferential equations. Neural Comput Appl 28(7):1–20
5. Arqub OA, Maayah B (2018) Solutions of Bagley–Torvik and Painlevé equations of fractional order using iterative reproducing kernel algorithm with error estimates. Neural Comput Appl 29(5):1465–1479
6. Zhang H, Llorca J, Davis CC, Milner SD (2012) Nature-inspired self-organization, control, and optimization in heterogeneous wireless networks. IEEE Trans Mob Comput 11(7):1207–1222
7. Zhang H, Xiong C, Ho JKL, Chow TWS (2017) Object-level video advertising: an optimization framework. IEEE Trans Ind Inf 13(2):520–531
8. Xu Y, Pi D (2019) A hybrid enhanced bat algorithm for the generalized redundancy allocation problem. Swarm Evolut Comput. https://doi.org/10.1016/j.swevo.2019.100562
9. Kennedy J, Eberhart R (2002) Particle swarm optimization. In: Icnn95-international conference on neural networks
10. Blackwell T, Kennedy J (2018) Impact of communication topology in particle swarm optimization. IEEE Trans Evolut Comput 23:689–702
11. Abido MA (2002) Optimal power flow using particle swarm optimization. Int J Electr Power Energy Syst 24(7):563–571
12. Abido AA (2001) Particle swarm optimization for multimachine power system stabilizer design. In: Power Engineering Society Summer Meeting
13. Ozcan E, Cad S, No TS, Mohan CK (2002) Particle swarm optimization: surfing the waves. In: Congress on evolutionary computation
14. Clerc M, Kennedy J (2002) The particle swarm: explosion, stability and convergence in multi-dimensional complex space. IEEE Trans Evolut Comput 20(1):1671–1676
15. Liao W, Wang J, Wang J (2006) Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. Comput Oper Res 33(3):859–871
16. Nickabadi A, Ebadzadeh MM, Safabakhsh R (2011) A novel particle swarm optimization algorithm with adaptive inertia weight. Appl Soft Comput J 11(4):3658–3670
17. Valle YD, Venayagamoorthy GK, Mohagheghi S, Hernandez JC, Harley RG (2008) Particle swarm optimization: basic concepts, variants and applications in power systems. IEEE Trans Evolut Comput 12(2):171–195
18. Banks A, Vincent J, Anyakoha C (2008) A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. Nat Comput 7(1):109–124
19. Kao YT, Zahara E (2008) A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Appl Soft Comput 8(2):849–857
20. Gong YJ, Li JJ, Zhou Y, Li Y, Chung HS, Shi YH, Zhang J (2017) Genetic learning particle swarm optimization. IEEE Trans Cybern 46(10):2277–2290
21. Deng L, Lu G, Shao Y, Fei M, Hu H (2016) A novel camera calibration technique based on differential evolution particle swarm optimization algorithm. Neurocomputing 174:456–465
22. Li Z, Wang W, Yan Y, Li Z (2015) PS–ABC: a hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems. Expert Syst Appl 42(22):8881–8895
23. Geng J, Li MW, Dong ZH, Liao YS (2015) Port throughput forecasting by MARS- R SVR with chaotic simulated annealing particle swarm optimization algorithm. Neurocomputing 147(1):239–250
24. Samma H, Lim CP, Saleh JM (2016) A new reinforcement learning-based memetic particle swarm optimizer. Appl Soft Comput 43(C):276–297
25. Kennedy J, Mendes R (2002) Population structure and particle swarm performance. In: Congress on evolutionary computation
26. Watkins CJCH, Dayan P (1992) Q-learning. Mach Learn 8(3–4):279–292
27. Rakshit P, Konar A, Bhowmik P, Goswami I, Das S, Jain LC, Nagar AK (2013) Realization of an adaptive memetic algorithm using differential evolution and Q-Learning: a case study in multirobot path planning. IEEE Trans Syst Man Cybern Syst 43(4):814–831
28. Samma H, Mohamad-Saleh J, Suandi SA, Lahasan B (2019) Q-learning-based simulated annealing algorithm for constrained engineering design problems. Neural Comput Appl 1:1–15
29. Feng W, Zhang H, Li K, Lin Z, Yang J, Shen X (2018) A hybrid particle swarm optimization algorithm using adaptive learning strategy. Inf Sci 436:162–177
30. Shi YH, Eberhart RC (1998) A modified particle swarm optimizer. In: The 1998 IEEE international conference on evolutionary computation proceedings, 1998. IEEE world congress on computational intelligence
31. Kennedy J (1999) Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In: Congress on evolutionary computation
32. Eberhart R, Kennedy J (2002) A new optimizer using particle swarm theory. In: Mhs95 sixth international symposium on micro machine & human science
33. Mendes R, Kennedy J, Neves J (2003) Watch thy neighbor or how the swarm can learn from its environment. In: Swarm intelligence symposium
34. Suganthan PN (1999) Particle swarm optimiser with neighbourhood operator. In: Congress on evolutionary computation
35. Bonyadi MR, Li X, Michalewicz Z (2014) A hybrid particle swarm with a time-adaptive topology for constrained optimization. Swarm Evolut Comput 18(1):22–37
36. Wei HL, Isa NAM (2014) Particle swarm optimization with increasing topology connectivity. Eng Appl Artif Intell 27(27):80–102
37. Marinakis Y, Marinaki M (2013) A hybridized particle swarm optimization with expanding neighborhood topology for the feature selection problem. In: Hybrid metaheuristics. 8th international workshop, HM 2013, pp 37–51
38. Goldberg DE, Richardson J (1987) Genetic algorithms with sharing for multimodal function optimization. In: International conference on genetic algorithms on genetic algorithms & their application
39. Goudos SK, Zaharis ZD, Kampitaki DG, Rekanos IT, Hilas CS (2009) Pareto optimal design of dual-band base station antenna arrays using multi-objective particle swarm optimization with fitness sharing. IEEE Trans Magn 45(3):1522–1525
40. Tao L, Wei C, Pei W (2004) PSO with sharing for multimodal function optimization. In: International conference on neural networks & signal processing

41. Li X (2010) Niching without niching parameters: particle swarm optimization using a ring topology. IEEE Trans Evolut Comput 14(1):150–169
42. Parrott D, Li X (2004) A particle swarm model for tracking multiple peaks in a dynamic environment using speciation. In: Congress on evolutionary computation
43. Peram T, Veeramachaneni K, Mohan CK (2012) Fitness distance, ratio based particle swarm optimization. In: Swarm intelligence symposium
44. Li X (2007) A multimodal particle swarm optimizer based on fitness Euclidean-distance ratio. In: Conference on genetic & evolutionary computation
45. Qu BY, Suganthan PN, Das S (2013) A distance-based locally informed particle swarm model for multimodal optimization. IEEE Trans Evolut Comput 17(3):387–402
46. Engelbrecht AP (2013) Particle swarm optimization: global best or local best? In: BRICS congress on computational intelligence & Brazilian congress on computational intelligence
47. Shi Y, Liu H, Gao L, Zhang G (2011) Cellular particle swarm optimization. Inf Sci Int J 181(20):4460–4493
48. Das PK, Behera HS, Panigrahi BK (2016) Intelligent-based multi-robot path planning inspired by improved classical Q-learning and improved particle swarm optimization with perturbed velocity. Eng Sci Technol Int J 19(1):651–669
49. Wang H, Sun H, Li C, Rahnamayan S (2013) Diversity enhanced particle swarm optimization with neighborhood search. Inf Sci 223(2):119–135
50. Liang JJ, Qu B-Y, Suganthan PN, Hernández-Díaz AG (2013) Problem definitions and evaluation criteria for the CEC 2013 special session and competition on real-parameter optimization. Technical Report 201212. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, January 2013
51. Montgomery DC (2009) Design and analysis of experiment, 7th edn. Wiley, New York
52. Shao Z, Pi D, Shao W (2018) A novel discrete water wave optimization algorithm for blocking flow-shop scheduling problem with sequence-dependent setup times. Swarm Evolut Comput 40:53–75. https://doi.org/10.1016/j.swevo.2017.12.005
53. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evolut Comput 1(1):3–18
54. Dor AE, Lemoine D, Clerc M, Siarry P, Deroussi L, Gourgand M (2015) Dynamic cluster in particle swarm optimization algorithm. Nat Comput 14(4):655–672
55. Lad BK, Kulkarni MS, Misra KB (2009) Optimal reliability design of a system. IEEE Trans Reliab R-22(5):255–258
56. Garg H, Rani M, Sharma SP (2013) An efficient two phase approach for solving reliability–redundancy allocation problem using artificial bee colony technique. Comput Oper Res 40(12):2961–2969
57. Hsieh YC, You PS (2011) An effective immune based two-phase approach for the optimal reliability–redundancy allocation problem. Appl Math Comput 218(4):1297–1307
58. Ouyang HB, Gao LQ, Li S, Kong XY (2015) Improved novel global harmony search with a new relaxation method for reliability optimization problems. Inf Sci 305(C):14–55
59. Huang CL (2015) A particle-based simplified swarm optimization algorithm for reliability redundancy allocation problems. Reliability Engineering & System Safety 142:221–230
60. Ravi VRPJ, Zimmermann HJ (2000) Fuzzy global optimization of complex system reliability. IEEE Trans Fuzzy Syst 8(3):241–248
61. Ravi V, Murty BSN, Reddy PJ (1997) Nonequilibrium simulated-annealing algorithm applied to reliability optimization of complex systems. IEEE Trans Reliab 46(2):233–239

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.