Received November 18, 2018, accepted November 27, 2018, date of publication December 4, 2018,
date of current version December 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2885036

An Elitist Learning Particle Swarm Optimization With Scaling Mutation and Ring Topology

GUANGZHI XU[1,2], XINCHAO ZHAO[2], TONG WU[3], RUI LI[2], AND XINGMEI LI[4]

[1]Automation School, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China
[3]School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China
[4]School of Economics and Management, North China Electric Power University, Beijing 102206, China

Corresponding author: Xinchao Zhao (xcbupt@126.com)

This work was supported by the National Natural Science Foundation of China under Grant 61375066 and Grant 71772060.

ABSTRACT Particle swarm optimization (PSO) is an evolutionary algorithm that is well known for its simplicity and effectiveness. It usually has strong global search capability but has the drawback of being easily trapped by local optima. A scaling mutation strategy and an elitist learning strategy are presented in this paper. Based on these strategies, an improved PSO variant (LSERPSO) is developed through a local search and ring topology strategy. The new scaling mutation strategy involved an exploration and exploitation balance focusing on mutation operation. A collection of elite individuals is maintained such that an array of current particles can learn from them. A ring topology-based neighborhood structure is adopted to maintain the population diversity and to reduce the possibility of particles being trapped in local optima. Finally, a quasi-Newton-based local search is incorporated to enhance the fine-grained capability. The effects of these proposed strategies and their cooperation are verified step by step. The performance of LSERPSO is comprehensively studied using IEEE CEC2015 benchmark functions.

INDEX TERMS Particle swarm optimization, elitist learning, local search, ring topology, scaling mutation, swarm intelligence.

I. INTRODUCTION

Particle swarm optimization (PSO) is a simple yet efficient optimization method that has been rapidly developed in recent years, mainly due to its simple conceptualization, easy implementation, and fast convergence speed to derive a reasonable solution [1], [2]. PSO has been extensively studied in not only the complex numerical optimization problem, but also in difficult real-world optimization problems in recent years. PSO was proposed by Kennedy and Eberhart [3] in 1995 as a competitive, population-based algorithm. PSO is a bio-inspired computing algorithm that is enlightened by crowd behaviors such as bird flocking and fish schooling [4]. In the PSO method, each particle adjusts its position by gaining experience from its own past best position information and that of the global best particle in the entire population or its neighboring populations.

In recent years, various effective strategies have been developed to strengthen the performance of PSO, such as parameter tuning [5]–[7], the combination of various topological structures [4], [8]–[11], and hybridization with other optimization techniques [12]–[15]. Although noticeable progress and fruitful achievements have been attained, successfully balancing the exploration and exploitation capabilities of PSO has been the key to determining its competitive performance for optimization problems, especially for those with multimodal landscapes or high-relevance variables [16].

However, three defects exist in PSO: difficulty in maintaining diversity, limitations in local search ability, and difficulty balancing the exploration and exploitation capabilities of the population. It has been a challenge to improve PSO search capabilities because these defects generally contradict each other [17], [18]. In this study, four cooperative strategies are proposed to combat these critical defects and improve search capability: a steady framework with ring topology, an elitist select and learning strategy, a scaling mutation strategy, and a local search technique.

Firstly, the strategy of ring topology is incorporated into a standard PSO to promote and maintain the information diversity of multiple niches with the aim of escaping from local optima. A PSO with ring topology searched for optima in parallel. Accordingly, the probability of being trapped by the same local optimum may be reduced.

Secondly, a collection of elite individuals is maintained as a possible learning source for each particle. It is composed of the personal best particles and solutions with competitive fitness values. Each particle comprehensively learns from all recorded elite solutions [19].

An individual level-based conservatism and scaling mutation strategy is proposed in this study to provide more accurate guidance for the diversification of particles while improving the adaptation and flexibility of each particle [20].

PSO has strong global search capabilities; however, it lacks local search abilities. Therefore, it is only natural to incorporate a local search strategy into PSO so that the two could benefit each other. The combination of local search techniques and evolutionary algorithms has been widely investigated. Two examples include the gradient-based local search method [21]–[23] and the derivative-free local search method [24]–[27].

The four strategies collaborated and benefited each other to enhance the performance of PSO and to conquer its possible characteristic drawbacks. In order to maintain population diversity, a ring topology and elitist learning are used as non-greedy strategies in place of the traditional update method based on *gbest* and *pbest*, so that all the individuals did not necessarily need to fly to the only globally promising area. Then, a local search technique is used to find better solutions to improve the elite solution quality and to accelerate the convergence rate. Even if some particles fall into local optima, a scaling mutation strategy with a long jump would provide more opportunities to jump out of local optima and to begin to search in a new area.

The goals of this paper are as follows:

1) To provide a framework of ring topology based on modified PSO, which promotes and maintains information diversity.

2) To introduce a collection of elite solutions as learning sources for particles. The influence of an elite pool on the performance is analyzed.

3) To provide an individual level-based conservatism and scaling mutation strategy.

4) To demonstrate the benefit of adopting the local search method, which is combined with the ring topology framework.

This paper is organized as follows. Section 2 briefly introduces the canonical PSO and the related works. Section 3 describes the ring topology strategy and combined PSO framework. Section 4 briefly introduces the elitist select and learning strategy with different-sized elite pools. Section 5 provides the details of the scaling mutation strategy. Section 6 provides the algorithm framework of LSERPSO with local search techniques. Section 7 lays out the experimental and comparative studies we performed on the CEC 2015 benchmark suit. Section 8 concludes this paper.

II. RELATED STUDIES

A. CANONICAL PSO

When a particle swarm initializes, the individuals attempt to search for more and more promising solutions by learning from each other, communicating information, and interacting with the velocities of each other. Each particle adjusts its position and velocity dynamically according to its flying experience and its neighbors in the search space. The state of particle i is described by its current position $x_{ij} = [x_{i1}, x_{i2}, \ldots x_{iD}]$ and velocity $v_{ij} = [v_{i1}, v_{i2}, \ldots v_{iD}]$, where D is the number of decision variables. In the canonical PSO [28], the position and velocity of particles are represented through the following equations:

$$v_{ij}^{k+1} = w =_{ij}^{k} + c_1 r_1 (pbest_{ij}^{k} - x_{ij}^{k}) + c_2 r_2 (gbest_{ij}^{k} - x_{ij}^{k}) \quad (1)$$

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

where x_{ij}^{k} is the j-th dimension of the i-th particle at the k-th iteration and v_{ij}^{k} is its velocity component; $pbest_i$ is the personal best position found by particle i; $gbest$ is the global best position found so far by the whole swarm; c_1 and c_2 are the acceleration coefficients; r_1 and r_2 are two random uniform numbers in [0, 1]; and w is the inertia weight used to balance between flight inertia and changing momentum.

B. SOME PSO VARIANTS

PSO has become one of the most popular forms of swarm intelligence since its emergence. It has attracted a lot of attention from researchers worldwide [29]–[31]. Generally, there are four ways to strengthen the capability of PSO, as observed from the evolutionary history of PSO.

(1) Utilizing self-adaption parameters in the particle evolving process instead of the fixed parameters. Shi & Eberhart [32] designed an inertia-weight strategy with linearly decreasing over iterations and a fuzzy self-adaptive strategy to adapt the different optimization problems [33]. With respect to the acceleration parameters c_1 and c_2, Suganthan [34] suggested different values of c_1 and c_2 for different problems rather than fixed values to yield better performance. In comparison, a linearly time-varying acceleration coefficient was proposed to enhance the performance of PSO [35]. Zhan *et al.* [36] proposed a real-time evolutionary state estimation procedure. The algorithm parameters were adaptively updated based on the results of an ‘‘evolutionary factor’’ and evolutionary state estimations. Ismail and Engelbrecht [38] embedded the strategy parameters of PSO into the positions of particles, which enhanced the performance of the comprehensive learning PSO (CLPSO) [19].

(2) The niche and species methods with different topological structures have been extensively researched for PSO. Kennedy and Mendes [39] suggested that a small niche size was more suitable to maintaining diversity for complicated multimodal problems, whereas a larger niche size was shown to be more effective for unimodal problems. The selection of an appropriate niche size is hampered by various problems.

Aware of the noticeable effects of niche size, researchers have investigated the dynamic adaptation mechanisms to enhance the flexible selection for parameters [40]. Mendes *et al.* [41] presented a fully informed particle swarm in which all individuals interacted with each other and learned experiences around all their neighbors instead of the *gbest* individual and its own personal best individual only. Qu *et al.* [42] proposed a distance-based locally informed particle swarm method that attempted to eliminate the difficulties of specifying the parameters and enhance the global search ability of PSO. In order to preserve the potentially promising solutions that have been developed so far, Parsopoulos and Vrahitis [43] designed rules to check whether the solutions were satisfying or not. If the solution was satisfying, several individuals were generated around this particle for a finer search in a local area. With this modification, PSO could locate all the global optima for the selected functions. Brits *et al.* [44] proposed a NichePSO that further extended Parsopoulos and Vrahitis's model [45]. In NichePSO, multiple subswarms are produced from a main swarm population to locate multiple optimal solutions in the search space. In speciation-based PSO (SPSO), proposed in [46], subpopulation species are evaluated by independent PSOs. As species are updated around different local areas, the multiple global optima have the potential to be found successfully in parallel.

(3) Introducing the effective evolutionary techniques from other heuristic or metaheuristic algorithms to PSO is another modification direction. Gong *et al.* [47] proposed a generalized ''learning PSO'' paradigm, the GL-PSO, that included two cascading layers (exemplar generation and particle updates as per a normal PSO). Genetic operators were used to generate exemplars from which particles learned, and, in turn, historical search information of the particles provided guidance to the evolution of the exemplars. Liang *et al.* [19] described using the new learning strategy CLPSO, in which all other particles' historical best information was used to update the flying velocity. Zhan *et al.* [2] proposed the orthogonal learning PSO (OLPSO) to discover more useful information from the historical best experience and the neighborhood's best experience. OLPSO can help particles fly in better directions through the construction of a much more promising and efficient exemplar. Deb and Padhye [49] established clear and fundamental algorithmic linking between particle swarm optimization and genetic algorithms. The goal of their study was to highlight the concept of algorithmic linking in an attempt to design more efficient optimization algorithms. Chen *et al.* [50] presented an aging leader and challenger mechanism to adaptively establish a suitable leader to lead the swarm evolution.

(4) Traditional optimization methods are combined with PSO because they offer strong local search capabilities. However, PSOs usually have better exploration capabilities for global searches; however, they are weaker at fine local searches. To enhance the exploitation capability of PSO algorithms, different optimization techniques have been employed as local search strategies [25], [51]. These local search strategies are expected to drive PSO to more efficiently find a local or a global optimum. Both gradient-based local search and derivative-free local search techniques have been adopted to date. The Nelder-Mead simplex search method has been combined with PSO [52] to produce faster and more accurate convergence of both the method and particle swarm optimization. Santos *et al.* [53] presented a semi-autonomous particle swarm optimizer that used gradient-based information (fast exploitation with gradient information) and diversity control (exploration with diversity control) to optimize multimodal functions.

III. RING TOPOLOGY FRAMEWORK

In the history of PSO development, the connectivity model has been categorized as either local population or global population. In a global topology, every particle is connected with each other, which increases the risk of convergence on suboptimal local minima. As an alternative to global topology, a ring topology can be used to adjust the speed of information propagation in PSO population, thus alleviating the problem of premature convergence [54].

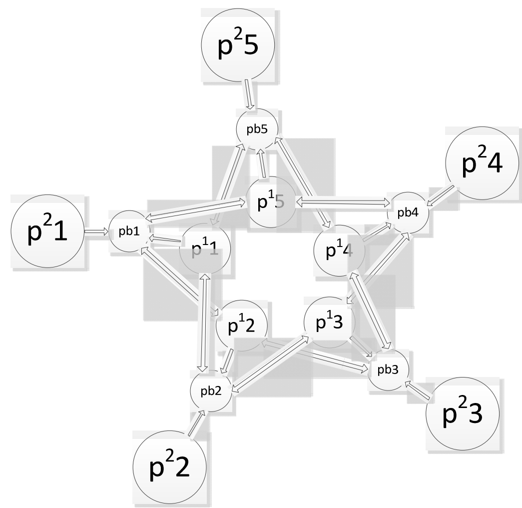

FIGURE 1. Ring topology in PSO in which each member interacts with its immediate left and right neighbors.

A ring topology with seven particles is shown in Fig.1, where $p_1^1 - p_5^1$ and $p_1^2 - p_5^2$ represent the current particles and the next generation particles, respectively, and $pb_1 - pb_5$ represents the personal best solutions of the current particles. Each particle interacted with three information sources, including the *pbest* of two neighboring particles and its own *pbest*. Each particle's *pbest* could be updated by comparing the particle itself with the particle from the next generation. Each stable topology consisted of three particles and their personal best solutions.

In order to maintain the population diversity, more stable personal best positions are retained in the topology to provide the best positions found so far. These positions could be further explored through the particles of the next generation.

In this study, we demonstrate that a PSO with ring topology is able to induce stable behavior and maintain a diverse population. Other ring communication topologies, such as two neighborhood members and one interacting member (left member or right member), have been shown to be very effective in alleviating the premature convergence of PSO [55].

PSO with ring topology operates as a niche algorithm by using particles' local memories (*pbest*) to form a stable network. It retains the best positions found so far and maintains diverse solutions with an elitist learning strategy instead of one *pbest* only. A ring topology can be described as follows. One particle interacts with two neighboring particles to form an ecological niche. The first particle is the neighbor of the last one and vice versa. The neighborhood of the$i$-th particle returns the best-fit personal best solution $nbest_i$ in its neighborhood, which represents the neighborhood best for the i-th particle.

Different particles residing on the ring may have different *nbests*. Thus, it is possible to converge to different optima for different particles over time.

$$v_{ij}^{k+1} = wv_{ij}^{k} + c_1 r_1 (pbest_{ij}^{k} - x_{ij}^{k}) + c_2 r_2 (nbest_{ij}^{k} - x_{ij}^{k}). \quad (3)$$

The ring topology neighborhood structure not only provides a mechanism to slow down the fast information propagation from the super individuals, but it also allows different neighborhood bests to coexist (rather than becoming homogeneous) over time. The reason for this phenomenon is that a particle's *nbest* will be updated only when there is a better personal best in its neighborhood. Furthermore, *nbest*-based PSOs do not require any prior knowledge of and no need to specify any niche parameters. The pseudocode of PSO with ring topology is presented in Algorithm I as follows:

Algorithm 1 Pseudocode of PSO With Ring Topology

Initialize population and parameters;
Repeat
for $i \leftarrow 1$ to population size **do**
  if $fit(x_i) < fit(pbest_i)$
    $pbest_i \leftarrow x_i$
  end if
  $nbest_i \leftarrow neighorhoodbest(x_{i-1}, x_i, x_{i+1})$
end
for $i \leftarrow 1$ to population size **do**
  Eq. (3)
  Eq. (2)
end
Until termination criterion is met

IV. ELITIST SELECTION AND LEARNING STRATEGY

It should be noted that PSO with a ring topological structure provides each particle the chance to learn from its local niche best and the *pbest* solutions. As a result, the probability of being trapped by local optimum is reduced. However, due to the inherent difficulties of a multimodal functional algorithm, it is also easy to be trapped by local optima. In addition, a particle's *pbest* is used as the learning source, and the promising particles are adopted as the potentially exemplars to guide the particle in flying. In canonical PSO, each particle discards its current *pbest* solution if an even better solution is found. However, the discarded solution may also be of relatively good quality and contain the promising information relating to the global optimal solution. In order to utilize this beneficial historical information, one elite set is constructed to act as an exemplar guidance pool. It is possible to increase the evolving diversity to yield an improved performance.

The elite set is composed of the historical *pbest*and other satisfactory suboptimal individuals with 10, 20, and 30 individuals. The satisfactory suboptimal solutions record some discarded individuals with comparable fitness with *pbest* solutions, but the location is opposite to the *pbest* solutions. After one iteration, the worst solution in the elite set is updated by the current particle, or the *pbest*, which has a better function value. Each particle learns information from the elite set (*eset*) randomly, which is described as (4) in Algorithm II as follows:

$$v_{ij}^{k+1} = wv_{ij}^{k} + c_1 r_1 (eset_{ij}^{k} - x_{ij}^{k}) + c_2 r_2 (nbest_{ij}^{k} - x_{ij}^{k}). \quad (4)$$

V. SCALING MUTATION STRATEGY

A new hybrid mutation strategy is proposed that aims to maintain the swarm diversity and prevent premature convergence. The proposed strategy also balances between exploration and exploitation by combining the mutation operation of differential evolution [56] and the global search ability of PSO. In evolutionary algorithms, large search space usually means great difficulties for convergence, because it is a challenge to give particles a large search space.

How to enlarge the exploitation areas of particles is important when improving the direction for PSO. However, overly large exploitation areas would not meet the inherent need of the algorithm to decrease the quality of particles. Mutation operation can preserve information through a crossover operation in differential evolution [26]. If a random number is larger than the crossover probability, the component of the original position of the particle will be copied to the new individual. Therefore, it is natural to combine both sides to utilize their advantages simultaneously. The conservatism and adventurism principle mutation, i.e., the scaling mutation strategy, is defined as follows:

$$l_j = \begin{cases} 0 & if\ s_j < -1 \\ 1 & else \end{cases} \quad (5)$$

$$v_{ij}^{k+1} = wv_{ij}^{k} + c_1 r_1 (eset_{ij}^{k} - x_{ij}^{k}) + c_2 r_2 (nbest_{ij}^{k} - x_{ij}^{k}) \quad (6)$$

$$x_{ij}^{k+1} = x_{ij}^{k} + |s_j| \cdot l_j v_{ij}^{k+1}, \quad (7)$$

where the Gaussian random number $s \sim \mathrm{N}(0, 1)$, $l_j \in \{0, 1\}$. If $s_j < -1$, $l_j = 0$, $x_{ij}^{k+1} = x_{ij}^{k}$. In this conservatism principle procedure, the corresponding j-th component of the i-th particle preserves the initial information from the previous position.

Algorithm 2 Pseudocode of Elitist Selection and Learning Strategy

Initialize population and parameters;
Repeat
for $i \leftarrow 1$ to population size **do**
$z_k = \arg\max \{f(z_m) | z_m \in eset\}$
  if $fit(x_i) < fit(pbest_i)$
    if $fit(z_k) > fit(pbest_i)$
      $z_k \leftarrow pbest_i$
    end
    $pbest_i \leftarrow x_i$
  else
    if $fit(z_k) > fit(x_i)$
      $z_k \leftarrow x_i$
    end
  end
end for
for $i \leftarrow 1$ to population size **do**
  Eq. (4)
  Eq. (2)
end
Until termination criterion is met

Algorithm 3 The Pseudocode of Local Search

The objective function is f(x), the starting point is x_0, the convergence tolerance is ε, the initial inverse Hessian approximation H_0, the gradient of the objective function is g_k
Set $k = 0$
While $\|g_k\| > \varepsilon$
  compute search direction $d_k = -H_k g_k$
  set $x_{k+1} = x_k + a_k d_k$, where a_k is the step and computed from line search method
  calculate $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$
  $H_k^{BFGS} = \left(I - \frac{s_k y_k^T}{s_k^T y_k}\right) H_k \left(I - \frac{y_k s_k^T}{s_k^T y_k}\right) + \frac{s_k s_k^T}{s_k^T y_k}$
  $k = k + 1$;
End while

Algorithm 4 The Flowchart of LSERPSO

Initialize maximum function evaluations MaxFEs, the elite pool of $eset$, particle size ps, the position x_i and velocity v_i for each particle
Set $pbest_i \leftarrow x_i$;
Select random $pbest$ to $eset$
Set $fitcount = ps$
While $fitcount <$ MaxFEs
  for $i \leftarrow 1$ to ps **do**
    if $fit(x_i) < fit(pbest_i)$ && $fitcount > \eta *$ MaxFEs
      execute local search on x_i, x_{i+1};
      n is the function evaluations cost by local search;
      $fitcount = fitcount + n$;
      update x_i, x_{i+1}
    end if
    $nbest_i \leftarrow neighorhoodbest(x_i, x_{i+1})$
  end for
  for $i \leftarrow 1$ to ps **do**
    update the velocity of particle i according to (6)
    update the position of particle i according to (7)
    evaluate the position;
    $fitcount = fitcount + 1$;
    $z_k = \arg\max \{f(z_m) | z_m \in eset\}$
    if $fit(x_i) < fit(pbest_i)$
      if $fit(z_k) > fit(pbest_i)$
        $z_k \leftarrow pbest_i$;
      end
      $pbest_i \leftarrow x_i$;
    else
      if $fit(x_k) > fit(x_i)$
        $z_k \leftarrow x_i$
      end
    end
    if $fit(nbest_{i+1}) < fit(nbest_i)$
      update $nbest_i$
    end if
  end for
End while
output the best individual and its fitness

If $s_j >= -1$, $l_j = 1$. Currently, this becomes the adventurism principle, in which particles can potentially move to new positions with different coefficients. Thus, it is possible for the search space to be scaled by this mutation strategy according to the above analysis.

VI. LOCAL SEARCH TECHNIQUE

PSO causes difficulty in improving the accuracy of solutions because of its exploration-biasing inherent principle. This can be enhanced through local search techniques; thus, a local search strategy based on the quasi-Newton method is proposed in this study.

By measuring the changes in a gradient, we used the quasi-Newton method [57] to construct a model of an objective function that is good enough to produce super-linear convergence. The improvement over steepest descent is dramatic, and the second order derivative is not required. In this study, if the *pbest* particle in the ring topology structure was replaced by an even better one at the later evolutionary stage of PSO (indicated by $fitcount > \eta *$ MaxEFs), the local search strategy is triggered and a fine search is conducted around the local promising neighborhood. With this modification, the newly proposed PSO make it more possible to locate the global optima for the optimization problems.

Of course, an overly strong local search technique can also pose a risk of premature convergence. To reduce such a risk, the algorithm also employs two exploration biasing strategies: the elitist selection and learning strategy and the scaling mutation strategy. These strategies is jointly used to

TABLE 1. Strategy comparison among canonical PSO, RPSO-2, and RPSO-3.

	Item	min	mean	median	std
f1	PSO	1.2908E+08	3.1585E+08	2.7442E+08	1.1496E+08
	RPSO-2	**3.3048E+03**	**1.6466E+07**	**7.2111E+06**	**2.4981E+07**
	RPSO-3	7.3632E+07	1.8161E+08	1.5517E+08	9.3139E+07
f2	PSO	7.1452E+09	2.5052E+10	2.3453E+10	1.1420E+10
	RPSO-2	**2.1636E+02**	**1.1725E+10**	**1.4912E+03**	**3.6037E+10**
	RPSO-3	4.0933E+09	1.0881E+10	8.8066E+09	6.3668E+09
f3	PSO	3.2081E+02	3.2087E+02	3.2087E+02	2.6626E-01
	RPSO-2	**3.2000E+02**	**3.2019E+02**	**3.2015E+02**	**2.0984E-01**
	RPSO-3	3.2077E+02	3.2084E+02	3.2084E+02	4.4312E-02
f4	PSO	1.2607E+03	1.4603E+03	1.4224E+03	1.3660E+02
	RPSO-2	**4.3283E+02**	**5.0556E+02**	**4.8656E+02**	**8.8587E+01**
	RPSO-3	5.9325E+02	6.6984E+02	6.7058E+02	4.3444E+01
f5	PSO	1.4710E+04	1.8411E+04	1.8501E+04	1.8613E+03
	RPSO-2	**1.7085E+03**	**3.1890E+03**	**3.2018E+03**	**5.5741E+02**
	RPSO-3	5.7104E+03	6.7865E+03	6.7954E+03	4.3054E+02
f6	PSO	6.0521E+06	2.1451E+07	1.9342E+07	1.1022E+07
	RPSO-2	**1.8469E+03**	**4.6284E+05**	**3.4352E+04**	**6.8105E+05**
	RPSO-3	3.7699E+05	3.5759E+06	3.3963E+06	2.5345E+06
f7	PSO	8.5833E+02	9.5834E+02	9.4350E+02	7.5217E+01
	RPSO-2	**7.0266E+02**	**7.1062E+02**	**7.0871E+02**	**5.7949E+00**
	RPSO-3	7.2933E+02	7.6277E+02	7.5528E+02	3.1366E+01
f8	PSO	2.5533E+06	8.5455E+06	7.5320E+06	5.6236E+06
	RPSO-2	**5.4643E+03**	**4.0613E+04**	**3.0257E+04**	**4.2026E+04**
	RPSO-3	1.1789E+05	7.4351E+05	4.8507E+05	8.4648E+05
f9	PSO	1.0717E+03	1.2107E+03	1.1189E+03	3.2721E+02
	RPSO-2	**1.0031E+03**	**1.0606E+03**	**1.0036E+03**	1.6381E+02
	RPSO-3	1.0104E+03	1.0611E+03	1.0488E+03	**6.5136E+01**
f10	PSO	7.3280E+05	2.3280E+06	2.5038E+06	2.0683E+06
	RPSO-2	**1.4995E+04**	**5.6582E+05**	**1.8819E+05**	**7.3179E+05**
	RPSO-3	3.7935E+05	2.2842E+06	1.8475E+06	1.6887E+06
f11	PSO	4.4040E+03	4.7521E+03	4.7410E+03	2.5802E+02
	RPSO-2	**1.4056E+03**	**1.5882E+03**	**1.4247E+03**	**2.9462E+02**
	RPSO-3	1.4786E+03	1.8922E+03	1.6872E+03	3.9773E+02
f12	PSO	1.3404E+03	1.3531E+03	1.3531E+03	6.1541E+01
	RPSO-2	**1.3072E+03**	**1.3228E+03**	**1.3107E+03**	3.7125E+01
	RPSO-3	1.3226E+03	1.3393E+03	1.3379E+03	**1.0358E+01**
f13	PSO	1.7562E+03	1.7829E+03	1.7811E+03	1.7736E+01
	RPSO-2	**1.4104E+03**	1.4885E+03	**1.4341E+03**	1.8138E+01
	RPSO-3	1.4235E+03	**1.4546E+03**	1.4526E+03	**1.3510E+01**

TABLE 1. *(Continued.)* Strategy comparison among canonical PSO, RPSO-2, and RPSO-3.

f14	PSO	1.4691E+05	1.8568E+05	1.7459E+05	5.7556E+04
	RPSO-2	**3.6682E+04**	**4.1289E+04**	**4.1068E+04**	**3.4750E+03**
	RPSO-3	4.3505E+04	5.7701E+04	5.5778E+04	1.0038E+04
f15	PSO	1.8638E+03	5.1378E+03	4.0604E+03	3.8611E+03
	RPSO-2	**1.6000E+03**	**1.6000E+03**	**1.6000E+03**	**6.2636E-12**
	RPSO-3	1.6402E+03	1.8341E+03	1.7341E+03	2.5698E+02

balance searches. The pseudocode of local search is given in Algorithm III as follows:

VII. FLOWCHART OF LSERPSO

The overall performance of LSERPSO is enhanced as a result of the mutual cooperation of four strategies to remedy the defects. To afford a balance between exploration and exploitation, a scaling mutation strategy is utilized to generate offspring and utilize the advantages of the conservatism and adventurism principle mutations. To further promote diversity, both the ring topology and elitist learning strategies are embedded in the framework of PSO. In addition, a local search algorithm is added to enhance the solution accuracy.

The LSERPSO flowchart based on the four cooperative strategies described previously is given in Algorithm IV.

VIII. EXPERIMENTAL STUDIES

This section documents the series of experiments are carried out to verify the feasibility and the cooperation of the proposed strategies and LSERPSO. Firstly, Section VII-A describes the benchmark functions suite and the usual parameters. Then, our observations and verification of the influence of each strategy is laid out step-by-step in Section VII-B. Section VII-C presents the detailed comparison between LSERPSO and several state-of-the-art algorithms.

A. BENCHMARK FUNCTIONS

Fifteen benchmark functions of IEEE CEC 2015 special session on real-parameter optimization are used to study the performance of the strategies and algorithms. A detailed description of these functions can be found in [58], which are divided into four classes:

1) unimodal functions f1–f2;
2) simple multimodal functions f3–f5;
3) hybrid functions f6–f8;
4) composition functions f9–f15.

To evaluate the performance of the proposed strategies and LSERPSO and to make fair comparison with the state-of-the-art algorithms, we utilized the following evaluation metrics: the best of, the mean of, the median of, and the standard variance of the final results in multiple runs. The maximum fitness evaluations (Max_Fes), the population size (NP), and the dimension number are set to the same value for all comparison algorithms as 800,000, 100, and 30, respectively. We carried out 30 independent runs of all experiments for statistical purposes.

B. OBSERVATION OF STRATEGIES

We considered the individual effects and the superposition effects of the proposed strategies. (i.e., the ring-topology, the elitist selection and learning strategy, the scaling mutation strategy, and the local search technique) in order to investigate the individual influence of the proposed strategies adopted in LSERPSO.

1) INFLUENCE OF RING TOPOLOGY SIZE

PSO, combined with a ring topology (RPSO) is used to maintain population diversity. The more ring communication topologies, the more effective performance is in preventing the premature convergence. Two differently-sized RPSOs are designed for the ring topology to verify the usefulness of the ring topology strategy. One group of ring topology contains only one neighborhood member (left member or right member), named RPSO-2 for short. The other one interacts with both members (left member and right member); it is named RPSO-3 for short.

Table 1 exhibits the comparison results among RPSOs with different ring topology size and canonical PSO based on the CEC2015 benchmark suite. The best results are highlighted in bold. The following phenomena can be observed from Table 1. Firstly, RPSO-2 and RPSO-3 show their superiority to canonical PSO for all functions. Secondly, RPSO-2 undoubtedly outstands among them on most functions. In summary, the ring topology strategy is shown to be helpful in improving the optimization algorithm, and this benefit is more evident for RPSO-2. Such a benefit comes from the potential balance between exploration and exploitation, which resulted from more small niches around the ring. So, the ring topology with one neighborhood member is considered in this paper.

2) INFLUENCE OF ELITIST SELECT AND LEARNING STRATEGY

For the elitist select and learning strategy, the elite individual is given tries with different sizes of {10, 20, 30} to enrich the pool of learning exemplars. The learning mechanism, chosen from the different local best or the elite individuals, is expected to significantly optimize the process.

TABLE 2. Strategy comparison on RPSO-2 and RPSO-2 with different elite pool sizes.

	Item	min	mean	median	std
f1	RPSO-2	**3.3048E+03**	1.6466E+07	7.2111E+06	2.4981E+07
	ERPSO-10	1.3519E+04	9.3610E+05	2.2024E+05	1.6681E+06
	ERPSO-20	1.4636E+04	**8.0606E+05**	**5.8485E+04**	**1.5870E+06**
	ERPSO-30	**4.0742E+03**	6.7934E+07	1.6949E+05	3.6662E+08
f2	RPSO-2	2.1636E+02	1.1725E+10	1.4912E+03	3.6037E+10
	ERPSO-10	2.1225E+02	**2.3300E+03**	**1.0541E+03**	3.0160E+03
	ERPSO-20	**2.0138E+02**	2.9955E+03	2.8213E+03	**2.3442E+03**
	ERPSO-30	2.2067E+02	3.1967E+03	2.9804E+03	1.7509E+04
f3	RPSO-2	**3.2000E+02**	**3.2019E+02**	**3.2015E+02**	2.0984E-01
	ERPSO-10	3.2073E+02	3.2088E+02	3.2089E+02	5.7012E-02
	ERPSO-20	3.2078E+02	3.2087E+02	3.2088E+02	**4.2718E-02**
	ERPSO-30	3.2074E+02	3.2088E+02	3.2088E+02	4.8211E-02
f4	RPSO-2	4.3283E+02	5.0556E+02	4.8656E+02	8.8587E+01
	ERPSO-10	4.1890E+02	4.3462E+02	4.3333E+02	1.0315E+01
	ERPSO-20	**4.1691E+02**	**4.3187E+02**	**4.3084E+02**	**8.9355E+00**
	ERPSO-30	4.1890E+02	4.3280E+02	4.3134E+02	1.0463E+01
f5	RPSO-2	1.7085E+03	3.1890E+03	3.2018E+03	5.5741E+02
	ERPSO-10	**1.4242E+03**	2.5072E+03	2.5781E+03	**5.3459E+02**
	ERPSO-20	1.4522E+03	**2.4008E+03**	**2.4169E+03**	5.9535E+02
	ERPSO-30	8.7912E+02	2.5030E+03	2.5669E+03	5.5995E+02
f6	RPSO-2	**1.8469E+03**	4.6284E+05	3.4352E+04	6.8105E+05
	ERPSO-10	3.6804E+03	**4.6508E+04**	1.7311E+04	1.5079E+05
	ERPSO-20	2.5520E+03	4.7779E+04	1.8708E+04	**7.6227E+04**
	ERPSO-30	3.7348E+03	5.3972E+04	**1.1652E+04**	1.6694E+05
f7	RPSO-2	7.0266E+02	7.1062E+02	7.0871E+02	5.7949E+00
	ERPSO-10	7.0154E+02	7.0400E+02	**7.0325E+02**	2.2770E+00
	ERPSO-20	**7.0120E+02**	7.2218E+02	7.0329E+02	1.0240E+02
	ERPSO-30	7.0131E+02	**7.0362E+02**	7.0340E+02	**1.6313E+00**
f8	RPSO-2	5.4643E+03	4.0613E+04	3.0257E+04	4.2026E+04
	ERPSO-10	2.6113E+03	2.2232E+04	2.1430E+04	**1.3603E+04**
	ERPSO-20	**1.7014E+03**	3.0314E+05	**1.5702E+04**	1.5646E+06
	ERPSO-30	3.7950E+03	**2.0292E+04**	1.7425E+04	1.4735E+04
f9	RPSO-2	1.0031E+03	1.0606E+03	1.0036E+03	1.6381E+02
	ERPSO-10	**1.0019E+03**	1.0249E+03	**1.0022E+03**	8.2347E+01
	ERPSO-20	1.0019E+03	**1.0135E+03**	1.0022E+03	**6.2264E+01**
	ERPSO-30	1.0019E+03	1.0232E+03	1.0023E+03	9.4760E+01
f10	RPSO-2	1.4995E+04	5.6582E+05	1.8819E+05	7.3179E+05
	ERPSO-10	**7.8369E+03**	**7.8269E+04**	**5.0577E+04**	6.9645E+04
	ERPSO-20	8.4265E+03	1.0930E+05	9.0569E+04	7.7435E+04

TABLE 2. *(Continued.)* Strategy comparison on RPSO-2 and RPSO-2 with different elite pool sizes.

	ERPSO-30	1.0495E+04	8.1636E+04	6.9440E+04	**5.8311E+04**
f11	RPSO-2	1.4056E+03	1.5882E+03	1.4247E+03	2.9462E+02
	ERPSO-10	1.4028E+03	**1.4716E+03**	1.4179E+03	8.8143E+01
	ERPSO-20	**1.4020E+03**	1.4747E+03	1.4175E+03	**8.6331E+01**
	ERPSO-30	1.4038E+03	1.5005E+03	**1.4143E+03**	1.5190E+02
f12	RPSO-2	1.3072E+03	1.3228E+03	1.3107E+03	3.7125E+01
	ERPSO-10	**1.3034E+03**	**1.3087E+03**	1.3053E+03	**1.7914E+01**
	ERPSO-20	1.3034E+03	1.3092E+03	**1.3049E+03**	2.3973E+01
	ERPSO-30	1.3042E+03	1.3089E+03	1.3052E+03	1.9794E+01
f13	RPSO-2	1.4104E+03	1.4885E+03	1.4341E+03	1.8138E+01
	ERPSO-10	**1.3833E+03**	1.4752E+03	**1.4031E+03**	2.7300E+02
	ERPSO-20	1.3859E+03	**1.4139E+03**	1.4053E+03	**5.3702E+01**
	ERPSO-30	1.3927E+03	1.4431E+03	1.4074E+03	1.2809E+02
f14	RPSO-2	3.6682E+04	4.1289E+04	4.1068E+04	3.4750E+03
	ERPSO-10	3.3332E+04	4.0864E+04	3.5407E+04	2.0756E+04
	ERPSO-20	3.3400E+04	3.6441E+04	3.6201E+04	**2.2787E+03**
	ERPSO-30	**1.5000E+03**	**3.4744E+04**	**3.5338E+04**	6.6092E+03
f15	RPSO-2	1.6000E+03	1.6000E+03	1.6000E+03	6.2636E-12
	ERPSO-10	1.6000E+03	1.6000E+03	1.6000E+03	4.9058E-13
	ERPSO-20	1.6000E+03	1.6000E+03	1.6000E+03	4.0936E-13
	ERPSO-30	1.6000E+03	1.6000E+03	1.6000E+03	4.0498E-13

In order to determine whether the results of RPSO with the elitist learning strategy (ERPSO) is consistently superior to those of RPSO, the simulation results between them are presented in Table 2. Table 2 presents the statistical comparison among RPSO-2 and RPSO-2 with the elitist learning strategy, which has three different-sized elite pools. ERPSO-10 illustrates RPSO-2 combining with the elitist learning strategy of size 10. ERPSO-20 and ERPSO-30 have a similar meaning. The best results are highlighted in bold.

Some phenomena can be observed from the earlier discussion and Table 2. (1) ERPSO achieves better results on all functions than RPSO-2, except for f3. Even for f3, they also all achieve comparable results. (2) Three ERPSOs with an elite set of different sizes all perform similarly well on fifteen functions. Among them, ERPSO-10 yields a comparatively better performance compared with ERPSO-20 and ERPSO-30. Furthermore, ERPSO-10 achieved the same best results as RPSOE-20 on functions 3–6, and both of them perform much better than ERPSO with 30 sizes on these functions. ERPSO-20 achieves a better performance on some unimodal functions than ERPSO-30. ERPSO-10 also performs well on unimodal functions, and it performs better than ERPSO-20 on functions 6, 7, 8, 10, 11, and 12. To sum up these results, it is observed that the performance of the algorithm is affected by the elite pool size.

The numerical grading method is utilized to measure which performs the best among ERPSO-10, ERPSO-20, and ERPSO-30. If the mean fitness are equal, a grade of "0" is assigned, which means that both methods have the same performance in terms of the statistical test. If the difference in mean fitness between ERPSO-10 and ERPSO-20 or ERPSO-30 is larger than zero, the corresponding algorithm is denoted as "−1." Accordingly, its component is denoted as "1." If the difference in mean fitness is larger than one or two magnitudes, it is denoted as "−2" or "−3." On the contrary, it is denoted as "2" or "3." This numerical grading method indicates that the higher the score, the stronger the performance. As a result, the grades between ERPSO-10 and ERPSO-20 are "2" and "−2," and the grades between ERPSO-10 and ERPSO-30 are "1" and "−1." This means that the performance of ERPSO-10 is better than that of ERPSO-20 and ERPSO-30. Although the size of an elite individual set is set as 10 in this study for efficiency control and faster updating speed, both sizing 20 and 30 show improved performance.

3) EFFECTS OF SCALING MUTATION AND LOCAL SEARCH

This subsection the observations on the influence of the scaling mutation and its combination with the local search are conducted. ERPSO-10 with the scaling mutation is denoted

TABLE 3. Strategies comparison on scaling mutation and local search.

	Item	min	mean	median	std
f1	ERPSO-10	1.3519E+04	9.3610E+05	2.2024E+05	1.6681E+06
	SERPSO	1.2301E+04	3.7662E+05	1.8823E+05	6.0350E+05
	LSERPSO	**1.0000E+02**	**1.0032E+02**	**1.0006E+02**	**4.0487E-01**
f2	ERPSO-10	2.1225E+02	2.3300E+03	1.0541E+03	3.0160E+03
	SERPSO	2.0107E+02	2.2682E+03	1.0874E+03	2.7708E+03
	LSERPSO	**2.0000E+02**	**2.0001E+02**	**2.0001E+02**	**5.4208E-03**
f3	ERPSO-10	3.2073E+02	3.2088E+02	3.2089E+02	5.7012E-02
	SERPSO	3.2079E+02	3.2087E+02	3.2087E+02	4.5998E-02
	LSERPSO	**3.2000E+02**	**3.2000E+02**	**3.2000E+02**	**2.1912E-05**
f4	ERPSO-10	4.1890E+02	4.3462E+02	4.3333E+02	1.0315E+01
	SERPSO	**4.1691E+02**	**4.2652E+02**	**4.2487E+02**	**6.5385E+00**
	LSERPSO	4.2487E+02	4.3801E+02	4.3681E+02	9.4588E+00
f5	ERPSO-10	**1.4242E+03**	**2.5072E+03**	2.5781E+03	**5.3459E+02**
	SERPSO	1.5684E+03	2.9089E+03	**2.4344E+03**	1.5242E+03
	LSERPSO	1.7470E+03	3.0137E+03	2.9742E+03	6.0268E+02
f6	ERPSO-10	3.6804E+03	4.6508E+04	1.7311E+04	1.5079E+05
	SERPSO	3.5000E+03	4.3916E+04	3.6856E+04	2.9396E+04
	LSERPSO	**1.1184E+03**	**1.7962E+03**	**1.7999E+03**	**4.6430E+02**
f7	ERPSO-10	**7.0154E+02**	7.0400E+02	7.0325E+02	2.2770E+00
	SERPSO	7.0232E+02	**7.0346E+02**	7.0311E+02	9.7221E-01
	LSERPSO	7.0243E+02	7.0737E+02	7.0726E+02	1.5875E+00
f8	ERPSO-10	2.6113E+03	2.2232E+04	2.1430E+04	1.3603E+04
	SERPSO	3.8548E+03	2.0946E+04	2.4858E+04	1.8408E+04
	LSERPSO	**8.0606E+02**	**1.4059E+03**	**1.3125E+03**	**3.6830E+02**
f9	ERPSO-10	1.0019E+03	1.0249E+03	1.0022E+03	8.2347E+01
	SERPSO	**1.0018E+03**	1.0246E+03	1.0022E+03	1.0023E+02
	LSERPSO	1.0020E+03	**1.0023E+03**	1.0023E+03	**1.4339E-01**
f10	ERPSO-10	7.8369E+03	7.8269E+04	5.0577E+04	6.9645E+04
	SERPSO	7.3433E+03	5.9759E+04	3.7801E+04	5.3415E+04
	LSERPSO	**1.9442E+03**	**2.6231E+03**	**2.4749E+03**	**6.2298E+02**
f11	ERPSO-10	1.4028E+03	1.4716E+03	1.4179E+03	8.8143E+01
	SERPSO	1.4033E+03	1.4893E+03	1.5006E+03	8.7230E+01
	LSERPSO	**1.4024E+03**	**1.4522E+03**	**1.4077E+03**	**8.6300E+01**
f12	ERPSO-10	1.3034E+03	1.3087E+03	1.3053E+03	1.7914E+01
	SERPSO	1.3038E+03	1.3182E+03	1.3054E+03	4.0057E+01
	LSERPSO	**1.3020E+03**	**1.3052E+03**	**1.3051E+03**	**6.7310E-01**
f13	ERPSO-10	**1.3833E+03**	1.4752E+03	**1.4031E+03**	2.7300E+02
	SERPSO	1.3905E+03	1.4478E+03	1.4139E+03	1.3689E+02
	LSERPSO	1.3890E+03	**1.4081E+03**	1.4045E+03	**1.2611E+02**

TABLE 3. *(Continued.)* **Strategies comparison on scaling mutation and local search.**

f14	ERPSO-10	3.3332E+04	4.0864E+04	3.5407E+04	2.0756E+04
	SERPSO	3.2876E+04	3.4660E+04	3.4760E+04	1.1769E+03
	LSERPSO	**1.8864E+04**	**2.1468E+04**	**2.0058E+04**	2.8109E+03
f15	ERPSO-10	1.6000E+03	1.6000E+03	1.6000E+03	4.9058E-13
	SERPSO	1.6000E+03	1.6000E+03	1.6000E+03	4.1403E-13
	LSERPSO	1.6000E+03	1.6000E+03	1.6000E+03	**3.7615E-13**

as SERPSO, and SERPSO with the local search is denoted as LSERPSO. Table 3 presents the results of SERPSO and LSERPSO using the IEEE CEC2015 benchmark. The parameter of η is set as 0.05. The best items are highlighted in bold.

As Table 3 shows, LSERPSO defeated SERPSO in 11 functions (f1–f3, f6, and f8–f14), and SERPSO slightly outperforms LSERPSO in f4, f5, and f7. Both of them perform similarly in f15. Secondly, SERPSO defeats ERPSO on most functions except for f5, f11, and f12.

As a whole, these results indicates that the combination of scaling mutation and local search provides potential help for most of the functions, and that the benefit is more obvious in hybrid and complicated functions. Secondly, the local search scheme makes the significant improvement on the algorithm. In short, it can be concluded that both the scaling mutation strategy and the local search are shown to be very beneficial for ERPSO in locating the global optima.

C. OVERALL PERFORMANCE COMPARISON AMONG LSERPSO AND STATE-OF-THE-ART PSO VARIANTS

After observing the influence of four innovative strategies and their cooperation from the above series of experiments, overall performance comparison between LSERPSO and several state-of-the-art PSO variants are presented. The well-known competitors of SPSO2011 [59], CLPSO [19], OLPSO [43], and DMSDLPSO [26] are compared with LSERPSO on CEC 2015 benchmark functions.

Furthermore, the performance of the proposed LSERPSO is compared with three state-of-the-art peer algorithms, namely GAPSO [60], DEPSO [61], and GLPSO [47] on the second test suite (f101–f112) [62]. In the second test suite, f101–f104 are unimodal functions, f105 is unimodal in 2-D and 3-D space but has multiple optima when $D > 3$, f106 is a step function, and f107 is a noisy quartic function. f108–f112 are multimodal functions with different landscapes whose local optima increase exponentially with the increasing of dimension size.

To make a fair comparison, the population size and the maximum number of fitness evaluations are set to the same value for all algorithms. Other parameters are set as the recommendation references.

Table 4 shows the comparison results with respect to "min", "mean," "median," and "std" terms, which represent the best, the mean, the median, and the standard deviation for the final results, respectively, in 30 independent runs. The best performance is highlighted bold. The signals of '1,' '-1,' and '0' in the last row for each function indicate whether or not LSERPSO performs significantly better, significantly worse, or comparably in comparing with its competitors according to the Wilcoxon rank-sum test (significance level $\alpha = 0.05$) among LSERPSO and state-of-the-art PSO variants.

For the purposes of the illustration, the online evolving convergence curves for f1–f15 of five algorithms are plotted in Fig. 2. The horizontal axis is the number of function evaluations, and the vertical axis is the mean best fitness at each iteration over the course of multiple runs.

The mean errors and standard deviation obtained by four algorithms are presented in Table 5, where the performance rank of LSERPSO among the algorithms is also given.

Based on Table 4 and Fig. 2, LSERPSO possess the ability to find the best or second-best solutions for all benchmark functions with a high convergence rate and robust reliability, except for f5. According to the comparison results, the conclusions can be drawn as follows. For unimodal functions f1–f2, SPSO2011, CLPSO, and OLPSO have significantly worse results than those of DMSDLPSO and LSERPSO. LSERPSO obtains the best result for function f2 and achieves the best "min" and comparable "mean" item for f1. According to the average evolving curve and the statistical result of f1, LSERPSO performs much better than that of DMSDLPSO for most of the evolving generations, and DMSDLPSO outperforms LSERPSO only at the last generation. At the final stage, DMSDLPSO outperforms LSERPSO slightly. This demonstrates that LSERPSO can significantly improve the exploitation capability, which is beneficial to solving unimodal problems.

For multi-modal functions, LSERPSO, OLPSO, and DMSDLPSO variants all showed significantly better results than SPSO2011 and CLPSO in f3. DMSDLPSO exhibits the best performance in f5. However, their performance showed no significant difference in f4. The good performance of LSERPSO in the unimodal functions and some multi-modal functions are attributed to the cooperation of the strategies their ability to balance between exploration and exploitation, which is crucial for solving different kinds of problems.

TABLE 4. Comparison among SPSO2011, CLPSO, OLPSO, DMSDLPSO and LSERPSO over CEC 2015 benchmark.

	Item	SPSO2011	CLPSO	OLPSO	DMSDLPSO	LSERPSO
f1	min	1.0215E+05	1.3514E+06	5.6221E+05	1.0000E+02	**1.0000E+02**
	mean	1.3012E+05	2.6298E+06	1.7374E+06	**1.0000E+02**	1.0032E+02
	median	1.1264E+05	2.4173E+06	1.7848E+06	**1.0000E+02**	1.0006E+02
	std	4.0791E+04	1.3202E+06	9.3548E+05	**2.1972E-04**	4.0487E-01
		1	1	1	0	
f2	min	3.2259E+02	2.0116E+02	2.0352E+02	2.0002E+02	**2.0000E+02**
	mean	2.6346E+03	2.0322E+02	3.2022E+03	2.0002E+02	**2.0001E+02**
	median	1.4055E+03	2.0273E+02	2.1454E+03	2.0002E+02	**2.0001E+02**
	std	3.3784E+03	2.2228E+00	3.3028E+03	**2.7377E-03**	5.4208E-03
		1	1	1	1	
f3	min	3.2018E+02	3.2033E+02	3.2000E+02	3.2000E+02	**3.2000E+02**
	mean	3.2022E+02	3.2039E+02	3.2000E+02	3.2000E+02	**3.2000E+02**
	median	3.2018E+02	3.2041E+02	3.2000E+02	3.2000E+02	**3.2000E+02**
	std	4.6739E-02	4.0127E-02	4.4528E-07	**1.4845E-06**	2.1912E-05
		1	1	0	0	
f4	min	4.3030E+02	4.3537E+02	4.3681E+02	**4.2487E+02**	4.2628E+02
	mean	**4.3471E+02**	4.4739E+02	4.4557E+02	4.3801E+02	4.3542E+02
	median	**4.3482E+02**	4.4496E+02	4.4278E+02	4.3681E+02	4.3582E+02
	std	3.2471E+00	9.0520E+00	7.8784E+00	9.4588E+00	**2.5945E+00**
		0	1	1	1	
f5	min	3.0644E+03	2.7560E+03	2.5185E+03	**1.7211E+03**	1.7470E+03
	mean	3.2141E+03	2.8763E+03	2.8313E+03	**1.8647E+03**	3.0137E+03
	median	3.2314E+03	2.8870E+03	2.7976E+03	**1.8903E+03**	2.9742E+03
	std	1.4005E+02	**1.1278E+02**	3.0459E+02	1.3291E+02	6.2455E+02
		0	0	0	-1	
f6	min	1.1764E+04	2.4817E+05	2.2888E+05	1.2224E+03	**1.1184E+03**
	mean	1.4551E+04	4.6777E+05	1.1340E+06	**1.7139E+03**	1.7862E+03
	median	1.2966E+04	3.9683E+05	7.8018E+05	**1.6836E+03**	1.7939E+03
	std	2.9823E+03	2.4320E+05	1.0415E+06	**3.2770E+02**	4.6499E+02
		1	1	1	0	
f7	min	7.1215E+02	7.0605E+02	7.0426E+02	7.0481E+02	**7.0243E+02**
	mean	7.1393E+02	7.0688E+02	7.0737E+02	**7.0553E+02**	7.0577E+02
	median	7.1361E+02	7.0638E+02	7.0737E+02	**7.0569E+02**	7.0571E+02
	std	1.7322E+00	9.2080E-01	1.5105E+00	**5.8240E-01**	1.7912E+00
		1	1	1	0	
f8	min	8.5929E+03	8.3372E+04	6.8292E+04	1.1906E+03	**8.0606E+02**
	mean	1.3917E+04	1.2826E+05	2.5988E+05	1.4687E+03	**1.4059E+03**
	median	1.4024E+04	1.3559E+05	2.2755E+05	1.3198E+03	**1.3125E+03**

TABLE 4. *(Continued.)* **Comparison among SPSO2011, CLPSO, OLPSO, DMSDLPSO and LSERPSO over CEC 2015 benchmark.**

	std	3.8406E+03	2.6475E+04	1.7235E+05	**2.0623E+02**	3.6245E+02
		1	1	1	0	
f9	min	1.0020E+03	1.0029E+03	1.0029E+03	1.0027E+03	**1.0020E+03**
	mean	1.0024E+03	1.0031E+03	1.0031E+03	1.0027E+03	**1.0023E+03**
	median	1.0025E+03	1.0031E+03	1.0030E+03	1.0028E+03	**1.0023E+03**
	std	2.6431E-01	1.6881E-01	2.0766E-01	6.3249E-02	**1.4339E-01**
		0	1	1	0	
f10	min	1.6925E+04	6.8983E+04	4.8487E+04	2.6145E+03	**1.9442E+03**
	mean	3.6441E+04	1.2728E+05	3.2827E+05	2.7841E+03	**2.6231E+03**
	median	3.8859E+04	7.4003E+04	2.7315E+05	2.8144E+03	**2.4749E+03**
	std	1.2294E+04	1.0054E+05	2.6936E+05	**1.2862E+02**	6.2298E+02
		1	1	1	0	
f11	min	1.6402E+03	1.4115E+03	1.6209E+03	1.4041E+03	**1.4024E+03**
	mean	1.6929+03	1.4274E+03	1.6307E+03	1.6461E+03	**1.4252E+03**
	median	1.6827E+03	1.4225E+03	1.6450E+03	1.4115E+03	**1.4077E+03**
	std	9.8834E+01	**6.0829E+00**	9.0282E+01	1.0310E+02	8.6300E+01
		1	1	1	1	
f12	min	1.3027E+03	1.3053E+03	1.3044E+03	1.3042E+03	**1.3020E+03**
	mean	1.3065E+03	1.3063E+03	1.3063E+03	1.3056E+03	**1.3052E+03**
	median	**1.3044E+03**	1.3063E+03	1.3063E+03	1.3056E+03	1.3049E+03
	std	4.4187E-01	**4.1021E-01**	9.9362E-01	4.6075E-01	6.7310E-01
		1	1	1	1	
f13	min	1.3958E+03	1.3890E+03	1.3990E+03	**1.3881E+03**	1.3957E+03
	mean	1.4049E+03	1.4081E+03	1.4126E+03	**1.3940E+03**	1.4027E+03
	median	1.4048E+03	1.4045E+03	1.4125E+03	**1.3946E+03**	1.4025E+03
	std	4.5464E+00	1.2611E+02	5.3592E+00	2.8635E+00	**2.1485E+00**
		1	1	1	-1	
f14	min	3.2838E+04	3.2610E+04	3.2512E+04	2.0252E+04	**1.8864E+04**
	mean	3.5003E+04	3.2860E+04	3.4688E+04	2.3181E+04	**2.1468E+04**
	median	3.5170E+04	3.2854E+04	3.4818E+04	2.2879E+04	**2.0058E+04**
	std	1.3648E+03	1.9208E+02	**1.1781E+03**	2.2561E+03	2.8109E+03
		1	1	1	0	
f15	min	1.6000E+03	1.6000E+03	1.6000E+03	1.6000E+03	1.6000E+03
	mean	1.6000E+03	1.6000E+03	1.6000E+03	1.6000E+03	1.6000E+03
	median	1.6000E+03	1.6000E+03	1.6000E+03	1.6000E+03	1.6000E+03
	std	0.0000E+00	4.6636E-13	1.4243E-12	0.0000E+00	3.7615E-13
		0	0	1	0	

For the hybrid optimization functions f6–f8, which are formed with different multimodal types, LSERPSO dominate its competitors in f8 and performs comparably with competitors in f7. The reliability and robustness of LSERPSO and DMS-DLPSO are almost the same in function f6, in which it performs significantly better than the other three competitors.

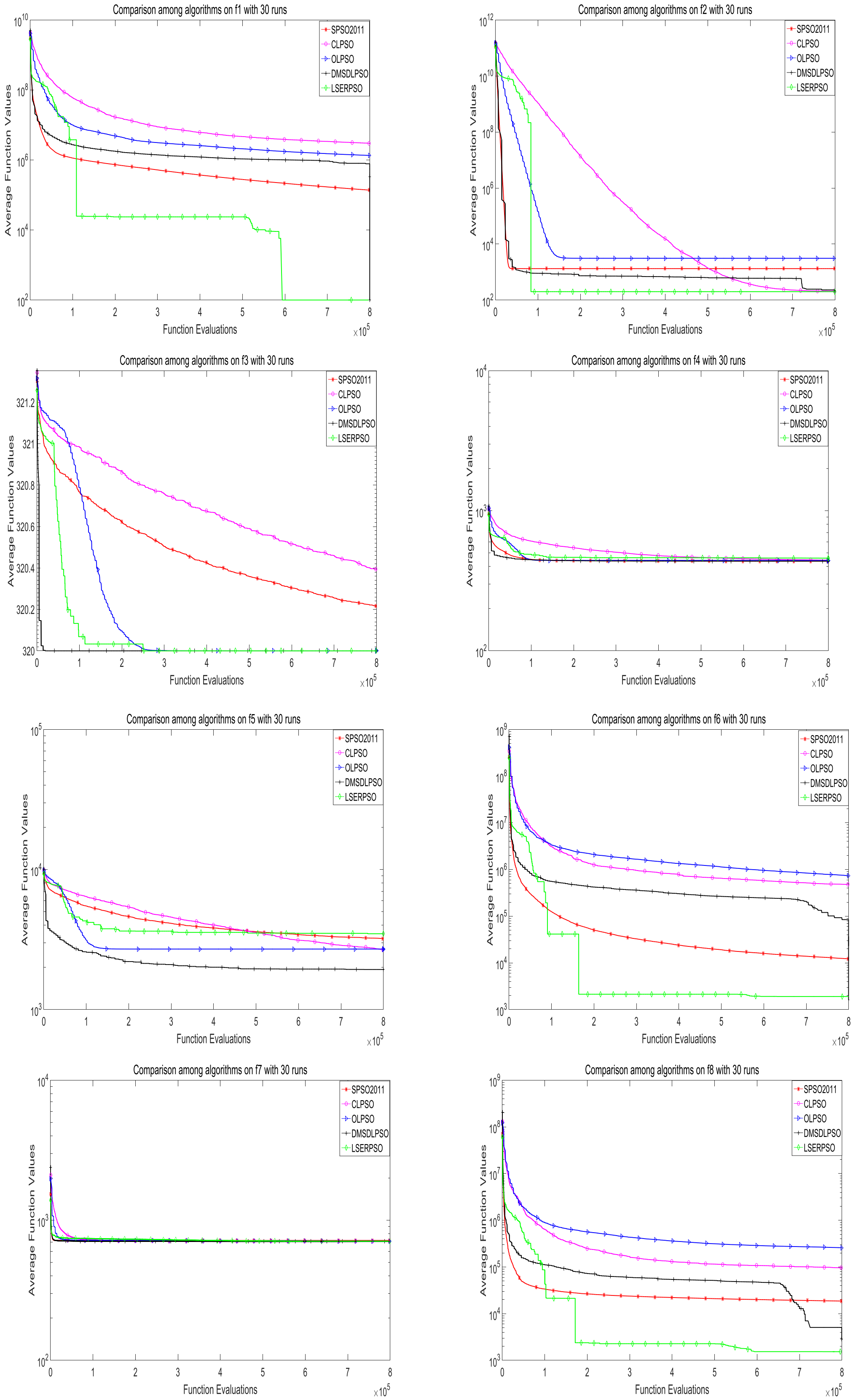

FIGURE 2. Online evolving convergence comparison among algorithms.

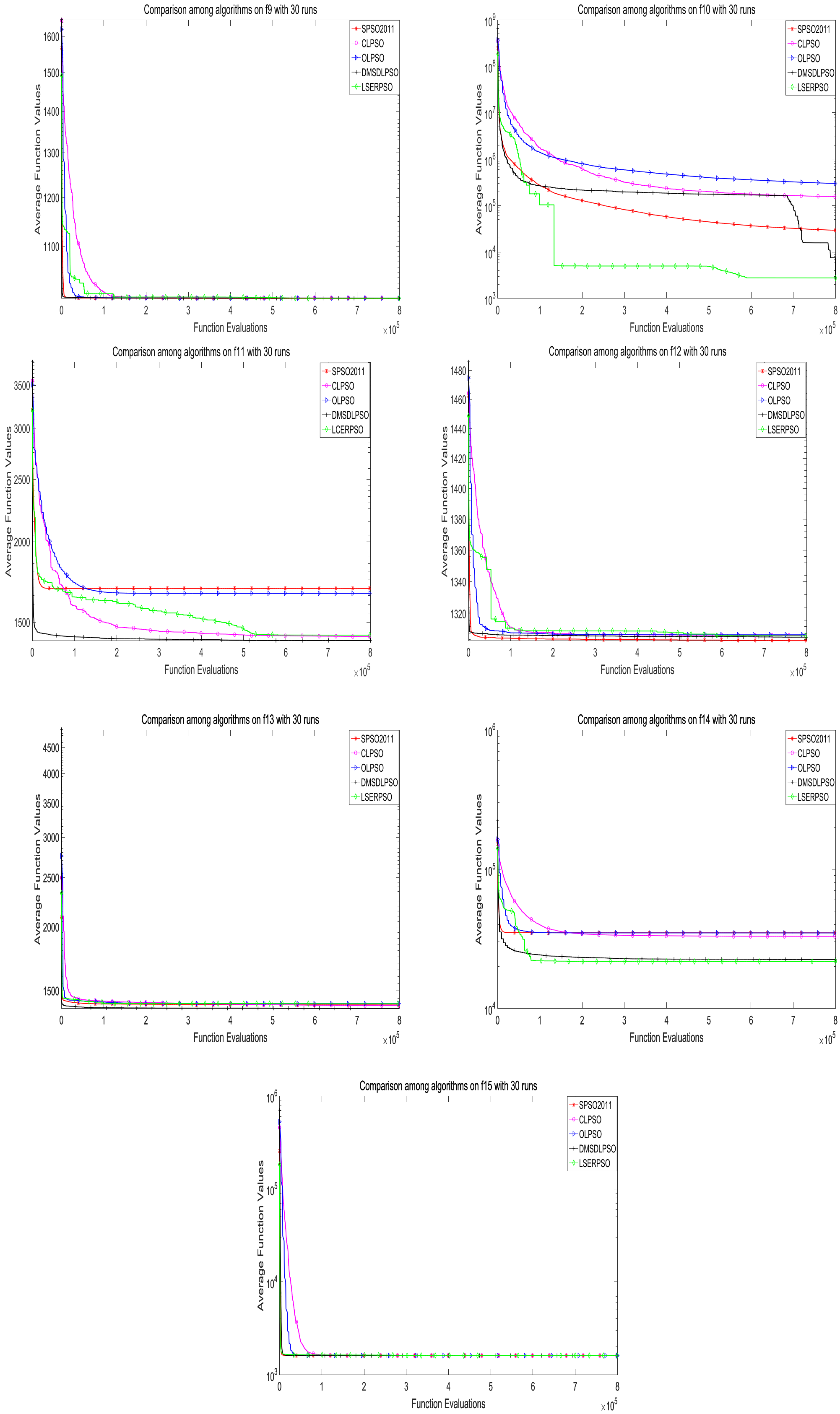

FIGURE 2. ***(Continued.)*** **Online evolving convergence comparison among algorithms.**

TABLE 5. Comparison among GAPSO, DEPSO, GLPSO and LSERPSO over second test suite (mean error ± standard deviation).

	GAPSO	DEPSO	GLPSO	LSERPSO	Rank of LSERPSO
f101	2.15E-11±3.27E-11	1.23E-26±2.98E-26	**1.32E-81±1.64E-81**	5.09E-73±6.00E-73	2
f102	1.12E-04±6.13E-04	3.38E-15±6.66E-15	**1.81E-46±1.14E-46**	2.62E-40±4.81E-40	2
f103	1.47E+02±2.23E+02	2.79E+02±1.34E+02	9.75E-16±1.45E-15	**2.11E-18±9.97E-19**	**1**
f104	9.36E-01±3.83E-01	4.60E-02±3.68E-02	3.52E-06±7.27E-06	**3.52E-08±2.71E-08**	**1**
f105	3.77E+01±2.83E+01	2.94E+01±2.44E+01	3.67E+00±3.34E+00	**1.98E+00±5.18E-01**	**1**
f106	0±0	0±0	0±0	0±0	0
f107	1.15E-02±3.32E-03	1.08E-02±3.10E-03	1.61E-02±8.71E-03	**2.69E-04±9.35E-05**	**1**
f108	1.84E+00±9.16E-01	4.59E+03±9.93E+02	**3.82E-04±0**	1.77E+00±5.42E-01	2
f109	0.50E-00±0.18E-00	1.14E+02±3.16E+01	9.47E-15±3.94E-14	**7.99E-15±3.89E-15**	**1**
f110	5.38E-07±4.15E-07	2.56E-14±1.55E-14	7.46E-15±1.85E-15	6.16E-10±2.41E-10	3
f111	1.25E-02±1.92E-02	1.45E-02±1.32E-02	5.91E-03±7.33E-03	**1.59E-32±0**	**1**
f112	3.35E-13±7.61E-13	1.19E-26±6.16E-26	1.57E-32±2.78E-48	**1.34E-32±0**	**1**

For composition optimization problems f9–f15, LSERPSO generally performs better than SPSO2011, CLPSO, and OLPSO. This is because the landscapes of composition functions are highly complex, and this makes it hard for many PSO variants to find the proper optima. Therefore, once the local best individuals are very close to the global best individual, the updating mechanism of PSO makes the particles' evolution stand still and usually result in prematurity. By contrast, LSERPSO and DMSDLPSO with the mutation strategy and the local search can more effectively enhance search directions for the particles to jump out from local traps. Based on the above analysis, PSO variants with a suitable mutation strategy and local search technique are usually more efficient than those without these strategies.

It can be observed from Table 5, the proposed algorithm is a competitive PSO variant on the second test suite when compares with its competitors, especially for the classical GLPSO [47]. During optimization of this test suite of 12 functions, LSERPSO ranks first seven times, second six times, and third once. This suggested that LSERPSO achieves very competitive results in most of functions when compared with algorithms such as GLPSO and the others. This highlighted the benefit of the cooperation of strategies and their ability to balance between exploration and exploitation.

Based on the overall performance comparison, LSERPSO shows a competitive performance when compared with several state-of-the-art algorithms. The superior performance of LSERPSO is attributed to its powerful global search ability and its accurate local search ability, which resulted from the proposed cooperative strategies laid out in this paper. Equipped with these strategies, LSERPSO can make a good balance between exploration and exploitation, which results in the efficiency and effectiveness in improving the performance of PSO.

IX. CONCLUSIONS

An elitist learning PSO with scaling mutation and ring topology is proposed in this study to balance the contradicting concepts of exploration and exploitation. A ring topology frame and an elitist select and learning strategy are introduced to maintain population diversity. The analysis on ring topology size indicates that RPSO-2 with only one neighbor has comprehensively better performance. It also suggests that multiple niches would lead to enhanced search ability, especially for multimodal functions. Different sizes of elite pools are discussed in terms of their influence on the performance of the algorithm. The comparison results demonstrates that ERPSO with elite sizes 10 and 20 have a similar performance. The scaling mutation strategy aims to reduce the flying scope with a high probability, and it stands still or enlarges the exploring areas with a small probability. The local search technique is designed to search for an accurate solution around local optimal seeds.

The influence of the proposed strategies is fully considered one by one, and their individual effects and cooperation are verified through simulation experiments. The comparison results with state-of-the-art algorithms based on two different test suites demonstrate the superiority and efficiency of LSERPSO.

ACKNOWLEDGMENT

The authors would like to express their awfully thanks to the Swarm Intelligence Research Team of BeiYou University.

REFERENCES

[1] C. Li, S. Yang, and T. T. Nguyen, "A self-learning particle swarm optimizer for global optimization problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 627–646, Jun. 2012.

[2] Z.-H. Zhan, J. Zhang, Y. Li, and Y.-H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, pp. 832–847, Dec. 2011.

[3] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci.*, Oct. 1995, pp. 39–43.

[4] X. Zhao, W. Lin, J. Hao, X. Zuo, and J. Yuan, "Clustering and pattern search for enhancing particle swarm optimization with Euclidean spatial neighborhood search," *Neurocomputing*, vol. 171, pp. 966–981, Jan. 2016.

[5] K. R. Harrison, A. P. Engelbrecht, and B. M. Ombuki-Berman, "Optimal parameter regions and the time-dependence of control parameter values for the particle swarm optimization algorithm," *Swarm Evol. Comput.*, vol. 41, pp. 20–35, Aug. 2018.

[6] M. S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, and G. Pasi, "Fuzzy self-tuning PSO: A settings-free algorithm for global optimization," *Swarm Evol. Comput.*, vol. 39, pp. 70–85, Apr. 2018.

[7] B. Tang, Z. Zhu, H.-S. Shin, A. Tsourdos, and J. Luo, "A framework for multi-objective optimisation based on a new self-adaptive particle swarm optimisation algorithm," *Inf. Sci.*, vol. 420, pp. 364–385, Dec. 2017.

[8] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer with local search," in *Proc. IEEE Congr. Evol. Comput.*, vol. 1, Sep. 2005, pp. 522–528.

[9] X. Xia, L. Gui, and Z.-H. Zhan, "A multi-swarm particle swarm optimization algorithm based on dynamical topology and purposeful detecting," *Appl. Soft Comput.*, vol. 67, pp. 126–140, Jun. 2018.

[10] M. Vora and T. T. Mirnalinee, "Dynamic small world particle swarm optimizer for function optimization," *Natural Comput.*, vol. 17, no. 4, pp. 901–917, 2018, doi: 10.1007/s11047-017-9639-9.

[11] X. Liang, W. Li, Y. Zhang, and M. Zhou, "An adaptive particle swarm optimization method based on clustering," *Soft Comput.*, vol. 19, no. 2, pp. 431–448, 2015.

[12] Q. Kang, C. Xiong, M. Zhou, and L. Meng, "Opposition-based hybrid strategy for particle swarm optimization in noisy environments," *IEEE Access*, vol. 6, pp. 21888–21900, 2018.

[13] W.-B. Du, Y. Gao, C. Liu, Z. Zheng, and Z. Wang, "Adequate is better: Particle swarm optimization with limited-information," *Appl. Math. Comput.*, vol. 268, pp. 832–838, Oct. 2015.

[14] W. Du and F. Zhang, "Genetic mechanism-enhanced standard particle swarm optimization 2011," *Soft Comput.*, vol. 22, no. 21, pp. 7207–7225, 2018, doi: 10.1007/s00500-017-2724-2.

[15] M. R. Bonyadi and Z. Michalewicz, "Analysis of stability, local convergence, and transformation sensitivity of a variant of the particle swarm optimization algorithm," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 370–385, Jun. 2016.

[16] Y.-J. Gong *et al.*, "Distributed evolutionary algorithms and their models: A survey of the state-of-the-art," *Appl. Soft Comput.*, vol. 34, pp. 286–300, Sep. 2015.

[17] Y. V. Pehlivanoglu, "A new particle swarm optimization method enhanced with a periodic mutation strategy and neural networks," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 436–452, Jun. 2013.

[18] Q. Yang *et al.*, "Segment-based predominant learning swarm optimizer for large-scale optimization," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2896–2910, Sep. 2017.

[19] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.

[20] G. Xu, R. Li, X. Zhao, and X. Zuo, "Conservatism and adventurism in particle swarm optimization algorithm," in *Proc. Asia–Pacific Conf. Simulated Evol. Learn.*, Shenzhen, China. Cham, Switzerland: Springer, 2017, pp. 1016–1025.

[21] M. Hu, T. Wu, and J. D. Weir, "An intelligent augmentation of particle swarm optimization with multiple adaptive methods," *Inf. Sci.*, vol. 213, pp. 68–83, Dec. 2012.

[22] F. Han and Q. Liu, "A diversity-guided hybrid particle swarm optimization based on gradient search," *Neurocomputing*, vol. 137, no. 4, pp. 234–240, Aug. 2014.

[23] B. Chen, W. Zeng, Y. Lin, and D. Zhang, "A new local search-based multiobjective optimization algorithm," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 50–73, Feb. 2015.

[24] A. Meng, Z. Li, H. Yin, S. Chen, and Z. Guo, "Accelerating particle swarm optimization using crisscross search," *Inf. Sci.*, vol. 329, pp. 52–72, Feb. 2016.

[25] G. Wu, D. Qiu, Y. Yu, W. Pedrycz, M. Ma, and H. Li, "Superior solution guided particle swarm optimization combined with local search techniques," *Expert Syst. Appl.*, vol. 41, no. 16, pp. 7536–7548, 2014.

[26] Y. Chen, L. Li, H. Peng, J. Xiao, and Q. Wu, "Dynamic multi-swarm differential learning particle swarm optimizer," *Swarm Evol. Comput.*, vol. 39, pp. 209–221, Apr. 2018.

[27] X. Zhao, W. Lin, and Q. Zhang, "Enhanced particle swarm optimization based on principal component analysis and line search," *Appl. Math. Comput.*, vol. 229, pp. 440–456, Feb. 2014.

[28] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 1998, pp. 69–73.

[29] Y. Lin, Y.-S. Jiang, Y.-J. Gong, Z.-H. Zhan, and J. Zhang, "A discrete multiobjective particle swarm optimizer for automated assembly of parallel cognitive diagnosis tests," *IEEE Trans. Cybern.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/8383956, doi: 10.1109/TCYB.2018.2836388.

[30] N. Lynn, M. Z. Ali, and P. N. Suganthan, "Population topologies for particle swarm optimization and differential evolution," *Swarm Evol. Comput.*, vol. 39, pp. 24–35, Apr. 2018.

[31] M. S. Kiran, "Particle swarm optimization with a new update mechanism," *Appl. Soft Comput.*, vol. 60, pp. 670–678, Nov. 2017.

[32] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, vol. 3, Jul. 1999, pp. 1945–1950.

[33] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, vol. 1, May 2001, pp. 101–106.

[34] P. N. Suganthan, "Particle swarm optimiser with neighbourhood operator," in *Proc. IEEE Congr. Evol. Comput.*, vol. 3, Jul. 1999, pp. 1958–1962.

[35] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 240–255, Jun. 2004.

[36] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1362–1381, Dec. 2009.

[37] K. R. Harrison, A. P. Engelbrecht, and B. M. Ombuki-Berman, "Self-adaptive particle swarm optimization: A review and analysis of convergence," *Swarm Intell.*, vol. 12, no. 3, pp. 187–226, 2018.

[38] A. Ismail and A. P. Engelbrecht, "The self-adaptive comprehensive learning particle swarm optimizer," in *Proc. Int. Conf. Swarm Intell.*, 2012, pp. 156–167.

[39] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proc. IEEE Congr. Evol. Comput.*, vol. 2, May 2002, pp. 1671–1676.

[40] X. Peng and Y. Wu, "Large-scale cooperative co-evolution using niching-based multi-modal optimization and adaptive fast clustering," *Swarm Evol. Comput.*, vol. 35, pp. 65–77, Aug. 2017.

[41] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 204–210, Jun. 2004.

[42] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 387–402, Jun. 2013.

[43] K. E. Parsopoulos and M. N. Vrahatis, "Recent approaches to global optimization problems through particle swarm optimization," *Natural Comput.*, vol. 1, nos. 2–3, pp. 235–306, 2002.

[44] R. Brits, A. P. Engelbrecht, and F. Van Den Bergh, "A niching particle swarm optimizer," in *Proc. Conf. Simulated Evol. Learn.*, vol. 2, 2002, pp. 692–696.

[45] K. E. Parsopoulos and M. N. Vrahatis, "Modification of the particle swarm optimizer for locating all the global minima," in *Proc. Int. Conf. Artif. Neural Netw. Genet. Algorithms*, Prague, Czech Republic. Vienna, Austria: Springer, 2001, pp. 324–327.

[46] D. Parrott and X. Li, "Locating and tracking multiple dynamic optima by a particle swarm model using speciation," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 440–458, Aug. 2006.

[47] Y.-J. Gong, *et al.*, "Genetic learning particle swarm optimization," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2277–2290, Oct. 2016.

[48] W. E. Costa, M. C. Goldbarg, and E. G. Goldbarg, "Hybridizing VNS and path-relinking on a particle swarm framework to minimize total flowtime," *Expert Syst. Appl.*, vol. 39, no. 18, pp. 13118–13126, 2012.

[49] K. Deb and N. Padhye, "Enhancing performance of particle swarm optimization through an algorithmic link with genetic algorithms," *Comput. Optim. Appl.*, vol. 57, no. 3, pp. 761–794, 2014.
[50] W.-N. Chen *et al.*, "Particle swarm optimization with an aging leader and challengers," *IEEE Trans. Evol. Comput.*, vol. 17, no. 2, pp. 241–258, Apr. 2013.
[51] B. Y. Qu, J. J. Liang, and P. N. Suganthan, "Niching particle swarm optimization with local search for multi-modal optimization," *Inf. Sci.*, vol. 197, pp. 131–143, Aug. 2012.
[52] S.-K. S. Fan and E. Zahara, "A hybrid simplex search and particle swarm optimization for unconstrained optimization," *Eur. J. Oper. Res.*, vol. 181, no. 2, pp. 527–548, 2007.
[53] R. Santos, G. Borges, A. Santos, M. Silva, C. Sales, and J. C. W. A. Costaa, "A semi-autonomous particle swarm optimizer based on gradient information and diversity control for global optimization," *Appl. Soft Comput.*, vol. 69, pp. 330–343, Aug. 2018.
[54] C. Yue, B. Qu, and J. Liang, "A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 805–817, Oct. 2018.
[55] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 150–169, Feb. 2010.
[56] S. X. Zhang, S. Y. Zheng, and L. M. Zheng, "An efficient multiple variants coordination framework for differential evolution," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2780–2793, Sep. 2017.
[57] L. Luksan and J. Vlcek, "New quasi-Newton method for solving systems of nonlinear equations," *Appl. Math.*, vol. 62, no. 2, pp. 121–134, 2017.
[58] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization," Comput. Intell. Lab., Nanyang Technol. Univ., Singapore, Tech. Rep. 201411A, 2014.
[59] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, "Standard particle swarm optimisation 2011 at CEC-2013: A baseline for future PSO improvements," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 2337–2344.
[60] K. Premalatha and A. M. Natarajan, "Hybrid PSO and GA for global maximization," *Int. J. Open Problems Compt. Math.*, vol. 2, no. 4, pp. 597–608, 2009.
[61] C. Zhang, J. Ning, S. Lu, D. Ouyang, and T. Ding, "A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization," *Oper. Res. Lett.*, vol. 37, no. 2, pp. 117–122, 2009.
[62] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

GUANGZHI XU was born in Datong, Shanxi, China, in 1987. He received the M.S. degree in control theory and control engineering from the Taiyuan University of Science and Technology, China, in 2015. He is currently pursuing the Ph.D. degree in control science and engineering with the Beijing University of Posts and Telecommunications, Beijing, China.

From 2016 to 2017, he was a joined Ph.D. student with the Engineering Department, University of Glasgow, U.K. His research interests include evolutionary computation and intelligent control, big data system, and industry 4.0.

XINCHAO ZHAO received the B.S. and M.S. degrees from the Department of Mathematics, Capital Normal University, Beijing, China, in 1999 and 2002, and the Ph.D. degree from the Key Laboratory of Mathematics Mechanization, Chinese Academy of Sciences, Beijing, in 2005. He is currently a Professor with the School of Science, Beijing University of Posts and Telecommunications.

From 2013 to 2014, he was a Visiting Scholar with the School of Electrical Engineering and Computer Science, University of Essex, U.K., and with the School of Computer Science, University of Birmingham, U.K. He has published more than 60 research papers in EC&SI journals and conferences. His research interests include evolutionary computation, swarm intelligence, nonlinear optimization, and engineering applications. He is a member of CCF and serves in the program committee of several conferences.

TONG WU was born in Baoding, Hebei, China, in 1992. She received the M.S. degree in computer science from the Beijing University of Posts and Telecommunications, Beijing, China, where she is currently pursuing the Ph.D. degree with the Information Security Center, School of Cyberspace Security. Her research interests include network security and machine learning.

RUI LI was born in Qingdao, Shandong, China, in 1991. He received the B.S. degree from the Department of Mathematics, Shandong University of Science and Technology, China, in 2015. He is currently pursuing the Ph.D. degree in systems science and engineering with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include evolutionary computation and swarm intelligence.

XINGMEI LI received the B.S. degree in mathematics from Beijing Normal University, the M.S. degree in mathematics from Capital Normal University, and the Ph.D. degree in management science from North China Electric Power University. She is a Professor with the School of Economics and Management, North China Electric Power University. She teaches Operations Research and Technical Economics. Being a Visiting Scholar, she is visiting EDWARD P. FITTS Department of Industrial and Systems Engineering.

Her current research interests are in the area of project portfolio selection, project scheduling, project planning, human factors in project management, and so on. She has published more than 40 papers in various journals. She is a member of the Chinese Society of Optimization, Overall Planning and Economic Mathematics.