Enhanced Comprehensive Learning Particle Swarm Optimization with Dimensional Independent and Adaptive Parameters

Xiang Yu and Yu Qiao

1Provincial Key Laboratory for Water Information Cooperative Sensing and Intelligent Processing, Nanchang Institute of Technology, Nanchang, Jiangxi 330099, China
2School of Mathematics and Information Science, Shaanxi Normal University, Xi’an, Shaanxi 710119, China

Correspondence should be addressed to Xiang Yu; xiang.yu@nit.edu.cn

Received 6 October 2020; Revised 3 January 2021; Accepted 25 January 2021; Published 5 February 2021

Copyright © 2021 Xiang Yu and Yu Qiao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Comprehensive learning particle swarm optimization (CLPSO) and enhanced CLPSO (ECLPSO) are two literature metaheuristics for global optimization. ECLPSO significantly improves the exploitation and convergence performance of CLPSO by perturbation-based exploitation and adaptive learning probabilities. However, ECLPSO still cannot locate the global optimum or find a near-optimum solution for a number of problems. In this paper, we study further bettering the exploration performance of ECLPSO. We propose to assign an independent inertia weight and an independent acceleration coefficient corresponding to each dimension of the search space, as well as an independent learning probability for each particle on each dimension. The dimensional independent maximum velocities, inertia weights, acceleration coefficients, and learning probabilities are proposed to be adaptively updated based on the dimensional normative intervals in order to facilitate exploration, exploitation, and convergence, particularly exploration. Our proposed metaheuristic, called adaptive CLPSO (ACLPSO), is evaluated on various benchmark functions. Experimental results demonstrate that the dimensional independent and adaptive maximum velocities, inertia weights, acceleration coefficients, and learning probabilities help to significantly mend ECLPSO’s exploration performance, and ACLPSO is able to derive the global optimum or a near-optimum solution on all the benchmark functions for all the runs with parameters appropriately set.

1. Introduction

Particle swarm optimization (PSO) [1, 2] is a powerful class of metaheuristics for global optimization. PSO simulates the social behavior of sharing individual knowledge when a flock of birds search for food. In PSO, flock and bird are, respectively, termed as swarm and particle, and each particle represents a candidate solution. Suppose the problem to be solved has \( D \) decision variables, each particle, denoted as \( i \), "flies" in a \( D \)-dimensional search space and is accordingly associated with a \( D \)-dimensional velocity \( V_i = \{V_{i,1}, V_{i,2}, \ldots, V_{i,D}\} \) and a fitness \( f(P_i) \) indicating the optimization performance of \( P_i \). The swarm of particles randomly initialize velocities and positions and search the global optimum iteratively, and the final solution found is the historical position that exhibits the best fitness value among all the particles. In each iteration (or generation), \( i \) updates \( V_i \) according to the present value of \( V_i \), the historical position giving \( i \)'s best fitness value so far (i.e., \( i \)'s personal best position), and the personal best positions of other particles.

Many different PSO variants have been proposed in the literature since the introduction of PSO in 1995 [3]. For the
earliest proposed global PSO (GPSO) [3, 4], the global best position with the best fitness value among all the particles’ personal best positions is used for particle velocity update. To be specific, in each generation, i’s velocity $V_i$ and position $P_i$ are adjusted on each dimension as follows:

$$V_{i,d} = wV_{i,d} + ar(B_{i,d} - P_{i,d}) + bs(G_d - P_{i,d}),$$

$$P_{i,d} = P_{i,d} + V_{i,d},$$

where $d (1 \leq d \leq D)$ is the dimension index; $w$ is the inertia weight; $a$ and $b$ are the acceleration coefficients; $r$ and $s$ are two random numbers uniformly distributed in $[0, 1]$; $B_i = \{B_{i,1}, B_{i,2}, \ldots, B_{i,D}\}$ is i’s personal best position; and $G = \{G_1, G_2, \ldots, G_D\}$ is the global best position. GPSO is liable to get stuck in a local optimum if the global best position is far from the global optimum. Local PSO (LPSO) [5] sets up a social topology with the shape of, e.g., ring, star, and pyramid. i’s neighborhood comprises i itself and the particles that are directly connected with i in the topology. Unlike GPSO, LPSO takes advantage of i’s local best position $L_i = \{L_{i,1}, L_{i,2}, \ldots, L_{i,D}\}$ that gives the best fitness value among i’s neighborhood to guide the flight trajectory update of i, as can be seen from the following equation:

$$V_{i,d} = wV_{i,d} + ar(B_{i,d} - P_{i,d}) + bs(L_{i,d} - P_{i,d}).$$

Compared with GPSO, LPSO reduces the chance of resulting in a local optimum. With respect to both GPSO and LPSO, i’s personal best position and the global/local best position are used for updating $V_i$ on all the dimensions. However, the personal best position and the global/local best position actually do not always contribute to the velocity update on each dimension. Comprehensive learning PSO (CLPSO) [6] and orthogonal learning PSO (OLPSO) [7] encourage i to learn from different exemplars on different dimensions according to equation (4) when updating $V_i$:

$$V_{i,d} = wV_{i,d} + ar(E_{i,d} - P_{i,d}),$$

where $E_i = \{E_{i,1}, E_{i,2}, \ldots, E_{i,D}\}$ is i’s exemplar’s position. In CLPSO, i is additionally associated with a fixed learning probability that controls $E_{i,d} = B_{i,d}$ or $E_{i,d}$ on each dimension $d$, where $j$ is a particle randomly selected, and $j \neq i$. OLPSO sets $E_{i,d}$ as i’s personal best position or the global/local best position on each dimension $d$ with the aid of orthogonal experimental design; OLPSO therefore has two versions: the global version OLPSO-G and the local version OLPSO-L. CLPSO and OLPSO redetermine $E_i$ if i’s personal best fitness value $f(B_i)$ does not improve for a consecutive number of generations. CLPSO and OLPSO significantly outperform GPSO and LPSO in terms of preserving the particles’ diversity and probing different regions of the search space to obtain a promising solution.

Metaheuristics including PSO need to address three important issues, namely, exploration, exploitation, and convergence. Exploration means searching diversely to locate a small region that possibly contains the global optimum, while exploitation refers to concentrating the search around the small region for solution refinement. Convergence is the gradual transition from initial exploration to ensuing exploitation. We studied enhancing the exploitation and convergence performance of CLPSO in [8], and our proposed PSO variant is called enhanced CLPSO (ECLPSO). ECLPSO calculates $B_d$ and $P_d$ which are, respectively, the lower bound and the upper bound of all the particles’ personal best positions on each dimension $d$ in each generation as follows:

$$B_d = \min\{B_{1,d}, B_{2,d}, \ldots, B_{N,d}\};$$

$$P_d = \max\{B_{1,d}, B_{2,d}, \ldots, B_{N,d}\};$$

where $N$ is the number of particles. $[B_d, P_d]$ is termed as the normative interval of dimension $d$. Let the search space on dimension $d$ be $[P_d, P_d]$ with $P_d$ being the lower bound and $P_d$ being the upper bound; ECLPSO deems that when $[B_d, P_d]$ becomes indeed small (i.e., no greater than 1% of $[P_d, P_d]$ and no greater than absolute value 2 simultaneously), the swarm of particles enter the exploitation phase on dimension $d$ (i.e., the global optimum or a near-optimum solution has been identified to be likely around the normative interval on dimension $d$); otherwise, the particles are still in the exploration phase on dimension $d$ (i.e., searching different regions on dimension $d$). ECLPSO adaptively updates the learning probability of each particle based on the ranking of all the particles’ personal best fitness values and the number of the dimensions that have entered the exploitation phase. In addition, ECLPSO conducts perturbation on each dimension $d$ that has entered the exploitation phase in order to find a high-quality solution around the normative interval on that dimension.

For a PSO variant, the velocity $V_{i,d}$ of each particle $i$ on each dimension $d$ is usually clamped by a maximum velocity $\nabla_d$, i.e.,

$$V_{i,d} = \begin{cases} \nabla_d, & \text{if } V_{i,d} > \nabla_d, \\ -\nabla_d, & \text{if } V_{i,d} < -\nabla_d, \\ V_{i,d}, & \text{otherwise.} \end{cases}$$

If $\nabla_d$ was too large, the particles might miss some promising solutions on dimension $d$; on the contrary, too small $\nabla_d$ would slow down the search process on dimension $d$. $\nabla_d$ is fixed at 20% of the dimensional search space $[P_d, P_d]$ by many literature PSO variants including GPSO, LPSO, CLPSO, OLPSO, and ECLPSO. The experimental results on various benchmark functions reported in [8] have demonstrated that though ECLPSO significantly improves the exploitation and convergence performance of CLPSO, it still cannot locate the global optimum or a near-optimum solution on a number of functions including Rosenbrock’s function, rotated Schwefel’s function, and rotated Rastrigin’s function. The flight trajectory and search behavior of all the particles in ECLPSO are directly affected by each dimension $d$’s maximum velocity $\nabla_d$, the inertia weight $w$, the acceleration coefficient $a$, and each particle $i$’s learning probability $L_i$. The experimental results reported in [8]
have also indicated that the search process of the particles evolves differently on each dimension; hence, in this paper, we propose to assign an independent inertia weight and an independent acceleration coefficient corresponding to each dimension, as well as an independent learning probability for each particle on each dimension. The dimensional independent maximum velocities, inertia weights, acceleration coefficients, and learning probabilities are adaptively updated based on the dimensional normative intervals in order to facilitate exploration, exploitation, and convergence, particularly exploration. We call the variant with the dimensional independent and adaptive parameters as adaptive CLPSO (ACLPSO).

We note that existing PSO variants, e.g., [3–42], have rarely considered using dimensional independent parameters other than the dimensional independent maximum velocities, and we find only one work [26] as an exception. In [26], Taherkhani and Safabakhsh modified GPSO, CLPSO, and OLPSO with an independent inertia weight and an independent acceleration coefficient for each particle $i$ on each dimension $d$; the inertia weight and the acceleration coefficient are adaptively adjusted according to the improvement status of $i$'s personal best fitness value and the distance between $i$'s dimensional position $P_{i,d}$ and $i$'s dimensional personal best position $B_{i,d}$ to achieve better exploration and faster convergence.

The rest of this paper is organized as follows. Section 2 reviews the related work on PSO. The more detailed working principles of CLPSO and ECLPSO are elaborated in Section 3. Section 4 presents our proposed dimensional independent and adaptive parameters and the space and time complexity analysis of ACLPSO. Performance evaluation of ACLPSO on a variety of benchmark functions is given in Section 5. Section 6 concludes this paper.

2. Related Work

A lot of researchers worldwide have studied PSO. The status quo and research trend of PSO relevant research are to investigate multistrategy and adaptivity based on the 4 typical PSO variants, i.e., GPSO, LPSO, CLPSO, and OLPSO. Multistrategy refers to employing multiple strategies, while adaptivity stands for adaptively setting some parameters as well as appropriately invoking and switching the strategies. Multistrategy and adaptivity aim to realize goals such as exploration, exploitation, and convergence and help the particles efficiently find the global optimum or a near-optimum solution.

2.1. Related Work Based on GPSO/LPSO. Zhan [9] proposed adaptive GPSO; the variant identifies the swarm’s evolution status based on the distribution of the distances among each particle and all the other particles; the inertia weight and acceleration coefficient are adaptively adjusted according to the swarm’s evolution status for expediting convergence; the variant additionally takes advantage of Gaussian mutation to appropriately impose some momentum on the global best position to help escaping from a local optimum. Median-oriented GPSO was studied in [10]; the variant assigns an independent acceleration coefficient for each particle $i$; $i$ is intentionally guided away from the swarm’s median position that gives the median fitness value among all the particles’ fitness values during the flight velocity update, and $i$’s associated acceleration coefficient is adaptively updated based on $i$’s fitness value, the swarm’s worst fitness value, and the swarm’s median fitness value so as to benefit jumping out of premature stagnancy in a local optimum and accelerating convergence. Chen et al. [11] introduced an aging mechanism with an aging leader and challengers for GPSO to address exploration; by evaluating the improvement status of the global best fitness value $f(G)$, all the particles’ personal best fitness values, and the leader’s fitness value, the variant adaptively analyzes the leader’s leading capability, adjusts the leader’s life span, and generates a challenger through uniform mutation to possibly replace the leader when the leader’s span becomes exhausted. GPSO augmented with multiple adaptive strategies was presented in [12]; non-uniform mutation and adaptive subgradient are alternatively applied to the global best position, respectively, contributing to escaping from a local optimum and conducting local search; the variant also performs Cauchy mutation on a randomly selected particle; as Cauchy mutation hinders convergence, the variant assigns an independent inertia weight and an independent acceleration coefficient for each particle and minimizes the sum of the distances between each particle and the global best position such that the inertia weights and acceleration coefficients are adaptively set and convergence is accordingly accelerated. In [13], LPSO with adaptive time-varying topology connectivity was investigated; for each particle $i$, the variant determines $i$’s historical contribution status to the global best position and the historical status of $i$’s topology connectivity getting stuck in a threshold value for every 5 consecutive generations and then adaptively updates $i$’s connectivity in the topology; the variant relies on neighborhood search to help the particles with their personal best fitness values ceasing improving in the present generation to jump out of stagnancy. Xia et al. [14] discussed GPSO with tabu detection and local search in a shrunk space; each dimension $d$ is segmented into 7 regions of equivalent sizes; for every 5 consecutive generations, the variant calculates the excellence level of each region on dimension $d$ based on the ranking of all the particles’ personal fitness values and the distribution of all the particles’ personal best positions in the regions; according to the excellence level of the region that the global best position belongs to, the variant appropriately randomly generates a possible replacement from some other region to assist escaping from a local optimum; when the global best position falls in a region on dimension $d$ for 80 consecutive generations, the variant shrinks the dimensional search space to that specific region for the purpose of speeding up convergence; moreover, the variant conducts local search with the aid of differential evolution. Other recent works related to integrating GPSO/LPSO with multistrategy and/or adaptivity include [15–32].
2.2. Related Work Based on CLPSO/OLPSO. Liang and Suganthan [33] proposed adaptive CLPSO with history learning; for every 20 consecutive generations, the variant adaptively updates each particle’s learning probability based on the best learning probability out of all the particles’ learning probabilities (i.e., having resulted in the biggest improvement for the personal best fitness value) and Gaussian distribution. Memetic CLPSO was introduced in [34]; the variant employs chaotic local search to help each particle that cannot improve the personal best fitness value for 10 consecutive generations getting out of stagnancy and applies simulated annealing to the particle whose personal best fitness value continues improving for 3 consecutive generations and whose personal best position is actually the global best position for solution refinement. Zheng et al. [35] studied adaptively determining the inertia weight for CLPSO according to the relative ratio of the number of particles with improved personal fitness values in the present generation and adaptively setting the acceleration coefficient by considering the sum of the ratio of each particle’s fitness change over the particle’s position change in the present generation. Superior solution guided CLPSO was presented in [36]; for the variant, the set of superior solutions includes not only each particle’s personal best position but also other historically experienced positions with excellent fitness values; each particle learns from the superior solutions for velocity update; the variant applies nonuniform mutation on each particle to help escaping from a local optimum, and the mutation is activated only when i’s personal best fitness value ceases improving for 50 consecutive generations, and the average distance between i’s position in the present generation and i’s position in the previous 5 generation is less than a threshold value; the variant additionally takes advantage of some local search techniques (e.g., quasi-Newton, pattern search, and simplex search) to refine the global best position after 80% of the search progress. Qin et al. [37] investigated 4 auxiliary strategies for OLPSO to generate an appropriate exemplar position, respectively, for the purpose of preserving diversity, jumping out of premature stagnancy, accelerating convergence, and local search; the variant mutates the global best position to further strengthen exploration. Other recent works including [38–42] are also related to multistrategy and/or adaptivity research based on CLPSO/OLPSO.

3. Background

3.1. Comprehensive Learning Particle Swarm Optimization. In equation (4), the inertia weight $w$ linearly decreases in each generation, and the acceleration coefficient $c$ is a constant value equivalent to 1.5. Let $\kappa_{\text{max}}$ be the predefined maximum number of generations; $w$ is updated in each generation $k$ according to the following equation:

$$w = w_{\text{max}} - \frac{k}{\kappa_{\text{max}}}(w_{\text{max}} - w_{\text{min}}),$$  (7)

where $w_{\text{max}} = 0.9$ and $w_{\text{min}} = 0.4$ are, respectively, the maximum and minimum inertia weights.

Equation (8) is the empirical expression for setting each particle $i$’s learning probability $L_i$. All the particles are associated with different learning probabilities.

$$L_i = L_{\text{min}} + \left(L_{\text{max}} - L_{\text{min}}\right) \frac{\exp\left(10\left(i - 1\right)/N - 1\right) - 1}{\exp(10) - 1},$$  (8)

where $L_{\text{max}} = 0.5$ is the maximum learning probability and $L_{\text{min}} = 0.05$ is the minimum learning probability.

For each particle $i$ on each dimension $d$, a random number uniformly distributed in $[0, 1]$ is generated; if the number is no less than $L_i$, the dimensional exemplar $E_{i,d} = B_{i,d}$; otherwise, $E_{i,d} = B_{j,d}$ with $j \neq i$. To determine $j$, two different particles excluding $i$ are randomly selected, and $j$ is the winner with a better fitness value out of the two particles. If $E_{i,d}$ is the same as $B_{i,d}$ on all the dimensions, CLPSO randomly chooses one dimension to learn from some other particle’s personal best position. CLPSO redetermines $i$’s exemplar position $E_i$ if $i$’s personal best fitness value ceases improving for 7 consecutive generations.

CLPSO calculates the fitness value of $i$ only if $i$ is feasible (i.e., within the dimensional search space $[B_{i,d}, \overline{B}_{i,d}]$ on each dimension $d$). If $i$ is infeasible, as all the dimensional exemplars are feasible, $i$ will eventually be drawn back to the search space.

3.2. Enhanced Comprehensive Learning Particle Swarm Optimization. ECLPSO introduces two enhancements, namely, perturbation-based exploitation (PbE) and adaptive learning probabilities (ALPs), to improve the exploitation and convergence performance of CLPSO.

In each generation, regarding each dimension $d$, if the dimensional normative interval $[B_{i,d}, \overline{B}_{i,d}]$ is indeed small, the PbE enhancement updates the dimensional position $V_{i,d}$ for each particle $i$ according to equation (9) instead of equation (4):

$$V_{i,d} = w_{\text{PbE}}V_{i,d} + a_{\text{PbE}}E_{i,d} + c\left(B_{i,d} - V_{i,d}\right),$$

where $w_{\text{PbE}} = 0.5$ is the inertia weight used exclusively for the PbE enhancement; $a_{\text{PbE}} = 1.5$ is the acceleration coefficient used exclusively for the PbE enhancement; and $c$ is the perturbation coefficient. $c$ is randomly generated from a Gaussian distribution with mean 1 and standard deviation 0.65, and $c$ is clamped to 10 times of the standard deviation on both sides of the mean. Each particle $i$ is pulled towards $V_{i,d}$ plus a perturbation term $c((B_{i,d} + \overline{B}_{i,d})/2 - V_{i,d})$ on dimension $d$. The PbE enhancement contributes to sufficient exploitation around the indeed small dimensional normative interval. Note that $V_{i,d}$ updated by equation (9) is not limited by the dimensional maximum velocity $\nabla_{i,d}$.

The minimum learning probability $L_{\text{min}}$ is fixed at 0.05. As expressed in equation (10), the maximum learning probability $L_{\text{max}}$ logarithmically increases in each generation $k$:
\[ L_{\text{max}} = L_{\text{min}} + h + q \log_{10}(D+1)(M_k + 1), \]

(10)

where \( M_k \) is the number of exploitation valid dimensions (i.e., the number of the dimensions whose normative intervals have ever become indeed small) before or just in generation \( k \); \( h = 0.25 \) is the difference coefficient; and \( q = 0.45 \) is the rate coefficient. \( L_{\text{max}} \) is small (i.e., 0.3) when \( M_k = 0 \) and benefits initial exploration. \( L_{\text{max}} \) increases rapidly with the particles’ exploitation progress to facilitate convergence. The ALP enhancement adaptively determines all the particles’ learning probabilities based on the ranking of the particles’ personal best fitness values, i.e.,

\[ L_i = L_{\text{min}} + (L_{\text{max}} - L_{\text{min}}) \frac{\exp(10(T_i - 1)/N - 1) - 1}{\exp(10) - 1}, \]

(11)

where \( T_i \) is \( i \)'s rank. If \( i \) gives the best personal best fitness value, then \( T_i = 1 \). A low-ranked particle is often better on more dimensions with respect to the personal best position than a high-ranked particle.

4. Adaptive Comprehensive Learning Particle Swarm Optimization

4.1. Dimensional Independent and Adaptive Maximum Velocities. Suppose the optimization problem to be solved is \( f(X) \) with \( X \) being the D-dimensional decision vector, and the global optimum is \( X^* = (X_1^*, X_2^*, \ldots, X_D^*) \). CLPSO and ECLPSO fail to observe and address the fact that, on a dimension \( d \), if the dimensional global optimum \( X_{d}^* \) is located near either bound of the dimensional search space \([P_d, \overline{P}_d]\) and all the particles’ dimensional personal best positions are scattered (i.e., the dimensional normative interval \([B_d, \overline{B}_d]\) is large), then it would be difficult for the swarm of particles to locate \( X_{d}^* \); this is because the dimensional maximum velocity \( V_d \) updated by equation (4) is restricted to be 20% of \([P_d, \overline{P}_d]\). Figure 1 illustrates this phenomenon. In Figure 1(a), a particle \( i \)'s dimensional position \( P_{i,d} \) and \( X_{d}^* \) are close to different bounds of \([P_d, \overline{P}_d]\), and \( i \)'s dimensional exemplar position \( E_{i,d} \) is located in between \( P_{i,d} \) and \( X_{d}^* \); the distance \( E_{i,d} - P_{i,d} \) and the distance \( X_{d}^* - E_{i,d} \) both are equal to 40% of \( \overline{P}_d - P_d \); and \( i \) needs at least 2 generations to reach around \( E_{i,d} \). As can be seen from Figure 1(b), when \( i \) flies past \( E_{i,d} \), its dimensional velocity update is influenced by two forces, i.e., the inertia force \( w V_{d,i} \) and the exemplar force \( (E_{i,d} - P_{i,d}) \); the more \( i \) being away from \( E_{i,d} \), the more the exemplar force is to pull it back to \( E_{i,d} \). In Figure 1(c), \( X_{d}^* \) is around \( \overline{P}_d \), and \( P_{i,d} \) is not that far from \( X_{d}^* \); however, \( E_{i,d} \) is close to \( P_d \); \( E_{i,d} \) guides \( i \) to fly away from \( X_{d}^* \). As a result, the chance for a particle to reach close to the dimensional global optimum is small. Furthermore, in case the dimensional global optimum is located near the dimensional search space bounds and the dimensional normative interval is large on a significant number of dimensions, CLPSO and ECLPSO would fail to find the global optimum or a near-optimum solution, e.g., on Rosenbrock’s function, rotated Schewefel’s function, and rotated Rastrigin’s function, for all the runs as reported in [8]. Therefore, \( V_d \) should not be fixed at 20% of \([P_d, \overline{P}_d]\). We propose to adaptively adjust \( V_d \) in each generation according to the following equation:

\[ V_d = s (\overline{B}_d - B_d), \]

(12)

where \( s \) is the scaling coefficient and is a positive value. \( V_d \) is positively related with the dimensional normative interval’s size \([\overline{B}_d, B_d]\). When \([\overline{B}_d, B_d]\) is large, \( V_d \) is large and contributes to timely flight for getting close to \( X_{d}^* \); on the contrary, \( V_d \) is small when \([\overline{B}_d, B_d]\) becomes small in order to benefit fine-grained search.

Allowing each particle \( i \)'s position \( P_{i,d} \) on each dimension \( d \) to be infeasible also inhibits the particles to move close to \( X_{d}^* \). Figure 1(d) shows an example; \( X_{d}^* \) is near \( \overline{P}_d \), \( P_{i,d} \) trespasses \( P_d \) and is infeasible, and \( E_{i,d} \) is around \( \overline{P}_d \); because of the force imposed by \( E_{i,d} \), \( P_{i,d} \) is pulled to a feasible dimensional position far from \( X_{d}^* \). Accordingly, an infeasible dimensional position is proposed to be repaired immediately by reinitialization between the previous feasible dimensional position and the trespassed dimensional search space bound [43, 44].

4.2. Dimensional Independent and Adaptive Inertia Weights and Acceleration Coefficients. For CLPSO and ECLPSO, the inertia weight \( w \) used in equation (4) is initially large to result in a large inertia force and is helpful for exploration, and \( w \) linearly decreases in each generation to gradually decrease the inertia force for the purpose of facilitating convergence and solution refinement. As \( w \) is dynamically updated according to the generation counter \( k \) in equation (7), \( w \) might obstruct exploration if the swarm of particles had not found the global optimum or a near-optimum solution even when \( k \) is large, and \( w \) might also impede convergence if a promising solution had already been located, and the particles can thus start solution refinement even when \( k \) is small. In addition, the same \( w \) is used with respect to all the dimensions. The search processes of the particles often evolve differently on different dimensions, i.e., taking different number of generations for the exploration phase. We thus propose to assign an independent weight \( w_d \) for each dimension \( d \) to replace \( w \) in equation (4) and adaptively set \( w_d \) when the dimensional normative interval \([B_d, \overline{B}_d]\) is greater than 1% of the dimensional search space \([P_d, \overline{P}_d]\) or greater than 2 as follows:

\[ w_d = \frac{\overline{B}_d - B_d}{\overline{P}_d - P_d} + (1 - u) \left( w_{\text{max}} - \frac{k}{k_{\text{max}}} (w_{\text{max}} - w_{\text{min}}) \right), \]

(13)

\[ w_d = \begin{cases} w_{\text{max}}, & \text{if } w_d > w_{\text{max}}, \\ w_{\text{min}}, & \text{if } w_d < w_{\text{min}}, \\ w_d, & \text{otherwise}, \end{cases} \]

(14)

where \( u \) is the tradeoff coefficient and is a positive number. \( u \) adjusts the tradeoff between the term \((\overline{B}_d - B_d)/(\overline{P}_d - P_d)\) and the term \( w_{\text{max}} - (k/k_{\text{max}})(w_{\text{max}} - w_{\text{min}}) \). The empirical
value chosen for \( u \) is 0.3. The incorporation of the term \((B_d - \bar{B}_d)/(\bar{P}_d - \bar{B}_d)\) aims to improve the particles’ exploration and convergence capabilities. If the dimensional normative interval \([\bar{B}_d, \bar{B}_d]\) is large, the particles are still exploring different regions of the dimensional search space, and accordingly, \( w_d \) needs to be large; when \([\bar{B}_d, \bar{B}_d]\) becomes small, \( w_d \) also grows small to facilitate convergence.

We further propose to assign an independent acceleration coefficient \( a_d \) for each dimension \( d \) to replace \( a \) in equation (4). \( w_d \) and \( a_d \) must satisfy the following so-called stability condition [26, 45, 46]:

\[
1 - w_d \geq 0 \quad \text{and} \quad w_d + 1 - ra_d \geq 0.
\]

Hence, \( a_d \) is simply adaptively adjusted according to the following equation:

\[
a_d = w_d + 1.
\]

4.3. Dimensional Independent and Adaptive Learning Probabilities. Regarding each particle \( i \) in CLPSO and ECLPSO, a large value for \( i \)'s learning probability \( L_i \) enables \( i \) to learn more from its own personal best position for velocity update and hence is beneficial for solution refinement, while a small value for \( L_i \) will let \( i \) to learn more from other particles’ personal best positions and accordingly encourages \( i \) to search diversely. \( L_i \) is adaptively updated based on \( i \)'s fitness rank \( T_i \) and the number of exploitation valid dimensions \( M_d \) in each generation \( k \). A serious issue occurs if \( M_k \) is 0 or a small value in all the generations, e.g., as reported on Rosenbrock’s function, rotated Schwefel’s function, and rotated Rastrigin’s function in [8]; small \( M_k \) leads to small learning probabilities for the swarm of particles and fails to realize convergence. We propose to assign an independent learning probability \( L_{i,d} \) for each particle \( i \) on each dimension \( d \) and adaptively set \( L_{i,d} \) in each generation \( k \) as follows:

\[
L_{i,d} = \nu \log_{\max} k + \frac{\bar{B}_d - B_d}{\bar{P}_d - P_d} \exp(D(T_i - 1)/N - 1) - 1, \exp(D) - 1.
\]

\[
L_{i,d} =\begin{cases} 
L_{\max}, & \text{if } L_{i,d} > L_{\max}, \\
L_{\min}, & \text{if } L_{i,d} < L_{\min}, \\
L_{i,d}, & \text{otherwise}, 
\end{cases}
\]

where \( L_{\min} = 0.05; L_{\max} = 0.75; \) and \( \nu \) is the learning probability-based coefficient and is a positive number no greater than 1. The term \( \log_{\max} k \) grows logarithmically with the generator counter \( k \) in order to facilitate convergence. The term \((\bar{B}_d - B_d)/(\bar{P}_d - P_d)\), being positively related with \( \bar{B}_d - B_d \), also benefits convergence when the dimensional normative interval \([\bar{B}_d, \bar{B}_d]\) is large.

4.4. Workflow and Complexity Analysis. ACLPSO is our proposed PSO variant based on ECLPSO with dimensional independent and adaptive maximum velocities, inertia weights, acceleration coefficients, and learning probabilities. The detailed step-by-step workflow of ACLPSO is as follows:

Step 1: for each particle \( i \) and each dimension \( d \), randomly initialize \( i \)'s dimensional velocity \( V_{i,d} \) and dimensional position \( P_{i,d} \) based on the dimensional search space \([\bar{P}_d, \bar{P}_d]\), calculate \( i \)'s fitness value \( f(P_i) \), and set \( i \)'s personal best fitness value \( f(B_i) = f(P_i) \), \( i \)'s dimensional personal best position \( B_{i,d} = P_{i,d} \), \( i \)'s cessation counter \( W_{i,d} = 0 \), the generation counter \( k = 1 \), the maximum number of generations \( k_{\max} \) and all the other parameters.

Step 2: if \( k \leq k_{\max} \) go to Step 3; otherwise, go to Step 7.
Step 3: for each dimension \( d \), determine the dimensional normative interval \([\bar{B}_d, \bar{B}_d]\), and update the dimensional maximum velocity \( V_d \) according to equation (12).
Step 4: for each particle \( i \), calculate \( i \)'s fitness rank \( T_i \) and if \( W_{i,d} = 0 \), reset \( W_{i,d} = 1 \) and reassign \( i \)'s dimensional exemplar \( E_{i,d} \) on each dimension \( d \).
Step 5: for each particle \( i \) and each dimension \( d \), adjust \( i \)'s dimensional learning probability \( L_{i,d} \) according to equations (17) and (18); if \([\bar{B}_d, \bar{B}_d]\) is indeed small, update \( V_{i,d} \) according to equation (9); otherwise, update the dimensional inertia weight \( w_d \) according to equations (13) and (14), the dimensional acceleration coefficient \( a_d \) according to equation (16), and \( V_{i,d} \) according to equations (4) and (6); update \( P_{i,d} \) according to equation (2), and repair \( P_{i,d} \) if \( P_{i,d} \) trespasses \([\bar{P}_d, \bar{P}_d]\).
Step 6: for each particle \( i \), calculate \( f(P_i) \); if \( f(P_i) > f(B_i) \), update \( W_{i,d} = W_{i,d} + 1 \) otherwise, set \( f(B_i) = f(P_i) \) and \( B_{i,d} = P_{i,d} \) on each dimension \( d \).
Step 7: output the global best position with the best fitness value among all the particles’ personal best positions.

As analyzed in [8], the time and space complexities of ECLPSO are, respectively, \( O(ND) \) bytes and \( O(k_{\max})(\log N + ND)) \) basic operations plus \( O(k_{\max}N) \) function evaluations (FEs). Concerning ACLPSO, storing the dimensional independent inertia weights and acceleration weights requires \( O(D) \) bytes, and storing the dimensional independent learning probabilities needs \( O(ND) \) bytes. Adaptively updating the dimensional independent maximum velocities, inertia weights, and acceleration coefficients calls for \( O(k_{\max}D) \) basic operations, and adaptively adjusting the dimensional independent learning probabilities demands for \( O(k_{\max}ND) \) basic operations. Therefore, the time and space complexities of ACLPSO are the same as those of ECLPSO.

5. Experimental Studies

5.1. Experimental Settings. The experimental hardware platform is a Microsoft Surface Pro laptop computer with an Intel Core i5-7300U central processor at the frequency of 2.6 GHz, 8 GB internal memory, and 256 GB solid-state disk
do the key parameters impact the performance of ACLPSO? (2) How do the dimensional independent and adaptive maximum velocities, inertia weights, acceleration coefficients, and learning probabilities improve the performance of ACLPSO? (3) How is the performance of ACLPSO as compared with other PSO variants? We consider 3 variants of ACLPSO, i.e., ACLPSO-1, ACLPSO-2, and ACLPSO-3, ACLPSO-1, ACLPSO-2, and ACLPSO-3 are the same as ACLPSO, except that ACLPSO-1 does not repair the dimensional position \( P_{1,d} \) for each particle \( i \) on each dimension \( d \) if \( P_{1,d} \) trespasses the dimensional search space \([P_d, \bar{P}_d]\). ACLPSO-2 does not adopt the dimensional independent and adaptive inertia weights and acceleration coefficients, and ACLPSO-3 does not take advantage of the dimensional independent and adaptive learning probabilities. Besides ACLPSO-1, ACLPSO-2, and ACLPSO-3, ACLPSO is further compared with CLPSO [6], ECLPSO [8], OLPSO-L [7], adaptive GPSO (AGPSO) [9], feedback learning GPSO with quadratic inertia weight (FLGPSO-QIW) [15], and GPSO with an aging leader and challengers (ALC-GPSO) [11]. For ACLPSO, ACLPSO-1, ACLPSO-2, ACLPSO-3, CLPSO, and ECLPSO, they are all implemented by Java, the number of particles \( N \) is set as 40, and 25 runs are executed on each function. The parameters of CLPSO, ECLPSO, OLPSO-L, AGPSO, FLPGPSO-QIW, and ALC-GPSO take the recommended values that were empirically determined based on extensive experiments on various benchmark functions in [6–9, 11, 15]. Note that the value of \( N \) could be different for different PSO variants. \( N \) is fixed at 40 for CLPSO, ECLPSO, and OLPSO-L in [6–8], while it is equal to 20 for AGPSO, FLPGPSO-QIW, and ALC-GPSO in [9, 11, 15]. As we do not have the source codes of OLPSO-L, AGPSO, FLPGPSO-QIW, and ALC-GPSO, we directly copy the results of these 4 variants from [7, 24] for performance comparison. Concerning all the PSO variants compared, each run consumes 200,000 FEs.

5.2. Experimental Results and Discussion. The mean and standard deviation (SD) global best fitness value results of

Figure 1: Illustration of why it is difficult for particle \( i \) to reach the dimensional global optimum \( X_d^* \) on dimension \( d \). (a) \( P_{1,d} \) and \( X_d^* \) are close to different bounds of \([P_d, \bar{P}_d]\), and \( E_{1,d} \) is located in between \( P_{1,d} \) and \( X_d^* \). (b) \( i \) flies past \( E_{1,d} \). (c) \( X_d^* \) is around \( \bar{P}_d \), \( P_{1,d} \) is not that far from \( X_d^* \), and \( E_{1,d} \) is close to \( P_{1,d} \). (d) \( X_d^* \) is around \( \bar{P}_d \), \( P_{1,d} \) trespasses \( \bar{P}_d \), and \( E_{1,d} \) is close to \( P_{1,d} \).
| Category         | Name                | Benchmark function                                                                 | Expression                                                                 | $X^*$ | $f(X^*)$ | Search space                | Initialization space |
|------------------|---------------------|------------------------------------------------------------------------------------|----------------------------------------------------------------------------|-------|----------|-----------------------------|---------------------|
| **Unimodal**     | Sphere              | $f_1(X) = \sum_{d=1}^{D} X_d^2$                                                   | $[0]^D$ 0 [-100, 100] $^D$                                               |       |          |                             |                     |
|                  | Schwenel P2.22      | $f_2(X) = \sum_{d=1}^{D} |X_d| + \prod_{d=1}^{D} |X_d|$                                                        | $[0]^D$ 0 [-10, 10] $^D$                                               |       |          |                             |                     |
|                  | Rosenbrock          | $f_3(X) = \sum_{d=1}^{D-1} (100 (X_{d+1} - X_d^2)^2 + (X_d - 1)^2)$              | $[1]^D$ 0 [-10, 10] $^D$                                               |       |          |                             |                     |
|                  | Schwenel P1.2       | $f_4(X) = \sum_{d=1}^{D} (X_d^2)^2$                                             | $[0]^D$ 0 [-100, 100] $^D$                                               |       |          |                             |                     |
|                  | Rastrigin           | $f_5(X) = 10D + \sum_{d=1}^{D} (X_d^2 - 10 \cos(2\pi X_d))$                      | $[0]^D$ 0 [-5.12, 5.12] $^D$                                            |       |          |                             |                     |
|                  | Noncontinuous Rastrigin | $f_6(X) = f_5(Y)$, $Y_d = \begin{cases} X_d, \text{ if } |X_d| < 0.5 \\ \text{Round}(2X_d)/2, \text{ otherwise} \end{cases}$ | $[0]^D$ 0 [-5.12, 5.12] $^D$                                            |       |          |                             |                     |
|                  | Ackley              | $f_7(X) = -20 \exp(-0.2\sqrt{\sum_{d=1}^{D} X_d^2}) - \exp\left(\frac{\sum_{d=1}^{D} \cos(2\pi X_d)}{\sqrt{D}}\right) + 20 + e$ | $[0]^D$ 0 [-32, 20] $^D$                                               |       |          |                             |                     |
| **Multimodal**   | Griewank            | $f_8(X) = 1/4000 \sum_{d=1}^{D} X_d^2 - \prod_{d=1}^{D} \cos(X_d / \sqrt{d}) + 1$ | $[0]^D$ 0 [-600, 600] $^D$                                              |       |          |                             |                     |
|                  | Schwenel            | $f_9(X) = 418.9829D - \sum_{d=1}^{D} \sin(\sqrt{|X_d|})$                         | $[420.96]^D$ 0 [-500, 500] $^D$                                         |       |          |                             |                     |
| **Shifted**      | Shifted Rosenbrock  | $f_{10}(X) = f_5(Y) + 390, Y = X - Z + 1$                                        | $Z$ 390 [-100, 100] $^D$                                               |       |          |                             |                     |
|                  | Shifted Griewank    | $f_{11}(X) = f_4(Y), Y = X - Z$                                                  | $Z$ 0 [-600, 600] $^D$                                               |       |          |                             |                     |
|                  | Shifted Schwenel P1.2 | $f_{12}(X) = f_4(Y) - 450, Y = X - Z$                                           | $Z$ -450 [-100, 100] $^D$                                             |       |          |                             |                     |
| **Rotated**      | Rotated Rastrigin   | $f_{14}(X) = f_5(Y), Y = X$                                                       | $[0]^D$ 0 [-5.12, 5.12] $^D$                                            |       |          |                             |                     |
|                  | Rotated Schwenel    | $Y_d = \begin{cases} Z_d \sin(\sqrt{|Z_d|}), \text{ if } |Z_d| \leq 500 \\ 0, \text{ otherwise} \end{cases}$, $f_{15}(X) = f_5(Y), Y = X$ | $[420.96]^D$ 0 [-500, 500] $^D$                                         |       |          |                             |                     |
|                  | Rotated Griewank    | $f_{16}(X) = f_4(Y), Y = X$                                                       | $[0]^D$ 0 [-600, 600] $^D$                                             |       |          |                             |                     |
|                  | Shifted rotated     | $f_{17}(X) = f_7(Y), Y = (X - Z)O$                                               | $Z$ 0 [-32, 32] $^D$                                               |       |          |                             |                     |
|                  | Shifted rotated Ackley | $f_{18}(X) = f_7(Y), Y = (X - Z)O$                                           | $[0]^D$ 0 [-32, 32] $^D$                                             |       |          |                             |                     |
Table 2: Mean and standard deviation global best fitness value results of ACLPSO with different values for the normative interval scaling coefficient $s$ and the learning probability-based coefficient $v$ on all the benchmark functions.

| Function | Global best fitness value | ACLPSO |
|----------|---------------------------|--------|
|          | I: $s = 1.1; \ v = 0.05$ | II: $s = 1.1; \ v = 0.3$ | III: $s = 0.1; \ v = 0.05$ | IV: $s = 0.1; \ v = 0.3$ |
| $f_1$    | Mean 1.23E-10             | 3.37E-105 | 2.87E-12 | 3.28E-115 |
|          | SD 6.08E-10               | 9.36E-105 | 8.18E-12 | 6.11E-115 |
| $f_2$    | Mean 1.48E-17             | 4.74E-22  | 3.09E-29 | 7.25E-37  |
|          | SD 2.19E-17               | 9.70E-22  | 4.27E-29 | 6.84E-37  |
| $f_3$    | Mean 4.23                 | 6.56E1    | 1.84     | 2.61E1    |
|          | SD 2.43                   | 2.49E1    | 1.63     | 2.87E1    |
| $f_4$    | Mean 1.56E4               | 1.71E4    | 3.39E2   | 7.87E-2   |
|          | SD 3.44E3                 | 8.24E3    | 8.84E1   | 8.72E-2   |
| $f_5$    | Mean 0                    | 1.55      | 5.17E-1  | 3.02      |
|          | SD 0                      | 9.12E-1   | 5.83E-1  | 1.45      |
| $f_6$    | Mean 4.19E-1              | 0         | 5.66E-1  | 2.00E-1   |
|          | SD 7.19E-1                | 0         | 6.63E-1  | 4.08E-1   |
| $f_7$    | Mean 1.92E-6              | 3.11E-15  | 1.15E-8  | 3.25E-15  |
|          | SD 6.96E-6                | 0         | 5.77E-8  | 7.11E-16  |
| $f_8$    | Mean 3.35E-8              | 0         | 1.87E-8  | 1.38E-11  |
|          | SD 3.88E-8                | 0         | 3.45E-8  | 4.56E-11  |
| $f_9$    | Mean 3.82E-4              | 4.29E2    | 2.23E2   | 5.87E2    |
|          | SD 9.15E-7                | 2.71E2    | 1.38E2   | 2.04E2    |
| $f_{10}$ | Mean 4.05E2               | 5.12E2    | 4.23E2   | 5.86E2    |
|          | SD 1.60E1                 | 1.02E2    | 2.32E2   | 2.56E2    |
| $f_{11}$ | Mean 1.26E-7              | 0         | 1.28E-7  | 1.96E-3   |
|          | SD 3.30E-7                | 0         | 2.34E-7  | 8.01E-3   |
| $f_{12}$ | Mean 4.19E3               | 3.26E2    | 6.37E2   | -4.48E2   |
|          | SD 9.97E2                 | 2.28E2    | 2.59E2   | 3.04      |
| $f_{13}$ | Mean 3.65E1               | 3.25      | 2.04E1   | 1.39E1    |
|          | SD 7.40                   | 1.88      | 3.70     | 2.53      |
| $f_{14}$ | Mean 5.78E2               | 2.53E2    | 1.35E3   | 1.32E3    |
|          | SD 4.00E2                 | 2.09E2    | 1.85E2   | 2.19E2    |
| $f_{15}$ | Mean 2.05E-3              | 5.33E-17  | 1.65E-3  | 2.76E-3   |
|          | SD 2.71E-3                | 6.51E-17  | 2.70E-3  | 5.91E-3   |
| $f_{16}$ | Mean 2.73E-5              | 3.96E-15  | 1.74E-6  | 3.82E-15  |
|          | SD 9.24E-5                | 1.55E-15  | 6.94E-6  | 1.45E-15  |

ACLPSO with different combinations for the values of the normative interval scaling coefficient $s$ and the learning probability-based coefficient $v$ on all the benchmark functions are listed in Table 2. Four different combinations are considered, i.e., I ($s = 1.1, \ v = 0.05$), II ($s = 1.1, \ v = 0.3$), III ($s = 0.1, \ v = 0.05$), and IV ($s = 0.1, \ v = 0.3$), and the best combination on each function is marked in bold. Table 3 gives the mean and SD final number of exploitation valid dimensions’ (i.e., $M_{\text{valid}}$) results of ACLPSO with 4 different ($s, v$) combinations on all the functions.

Table 7 lists the mean and SD global best fitness value results of ACLPSO with other parameter settings on $f_3, f_4, f_{12},$ and $f_{14}$. The mean and SD global best fitness value and execution time results of CLPSO with other parameter settings on $f_1$ and $f_2$ are given in Table 8. Figure 2 illustrates the changes of the global best fitness value during the search process of ACLPSO with the best ($s, v$) combination and in the best run on $f_2, f_3, f_4, f_{12}, f_{13},$ and $f_{14}$.

As can be seen from Table 2, the best ($s, v$) combinations are, respectively, IV, IV, IV, III, IV, I, II, I, I, I, II, IV, II, II, and IV on the 16 functions. ACLPSO with the best ($s, v$) combination is able to find the global optimum or a near-optimum solution on each function for all the 25 runs. ACLPSO is likely to get trapped in an unsatisfactory local optimum on $f_1$ with combinations II and IV, on $f_4$ with combinations I, II, and III, on $f_5$ with combinations II and IV, on $f_9$ with combinations II, III, and IV, on $f_{10}$ with combinations II and IV, on $f_{12}$ with combinations I and II, on $f_{13}$ with combinations I, III, and IV, and on $f_{14}$ with
combinations III and IV. The accuracy of the mean global best fitness value with the best \((s, v)\) combination is noticeably excellent on \(f_1, f_2, f_5, f_6, f_7, f_8, f_{11}, f_{15},\) and \(f_{16}\). The observations indicate that \(s\) and \(v\) are the key parameters for ACLPSO, and the performance of ACLPSO is sensitive to the values of \(s\) and \(v\). The normative interval scaling coefficient \(s\) determines the search granularity; large \(s\) encourages the particles to search in a large granularity so as to escape from an unsatisfactory local optimum and locate the global optimum or a near-optimum solution, whereas small \(s\) will let the particles to search in a small granularity so as not to miss the deep global optimum or a deep near-optimum solution during the search process. The learning probability-based coefficient \(v\) controls for a particle the number of dimensions learning from the particle’s own personal best position; large and small values for \(v\), respectively, contribute to the exchange of valuable information among the particles and preserving valuable information embodied in each particle; large \(v\) also benefits convergence and exploration but might lead to premature stagnancy and hinders exploration in case some valuable information about the global optimum or a near-optimum solution was not preserved. We can see from Table 3 that the mean number of exploitation valid dimensions’ results of ACLPSO with \(v\) being set as 0.3 is 30 or close to 30 on all the functions except \(f_6\); in contrast, the mean number of exploitation valid dimensions’ results of ACLPSO is less than 11 or even 0 with \(v\) being fixed at 0.05 on \(f_3, f_4, f_6, f_8, f_{10}, f_{11}, f_{12}, f_{13}, f_{14},\) and \(f_{15}\). The functions have different landscapes; thus, different \((s, v)\) combinations achieve the best performance on different functions.

In Table 4, ACLPSO-1, ACLPSO-2, and ACLPSO-3 take the same values for \(s\) and/or \(v\) as the best \((s, v)\) combination of ACLPSO on each function. The mean global best fitness value results of ACLPSO-1 are worse than those of ACLPSO on \(f_1, f_2, f_3, f_5, f_6, f_{10}, f_{11}, f_{12}, f_{13}, f_{14}, \) and \(f_{16}\). ACLPSO-2 performs inferior to ACLPSO in terms of the mean global best fitness value results on \(f_1, f_2, f_3, f_4, f_5, f_{10}, f_{11}, f_{12}, f_{13}, f_{14}, f_{15},\) and \(f_{16}\). The mean global best fitness value results of ACLPSO-3 are also worse than those of ACLPSO.
Table 4: Mean and standard deviation global best fitness value results of ACLPSO with the best (s, v) combination and other PSO variants on all the benchmark functions.

| Function | Global best fitness value | ACLPSO | ACLPSO-1 | ACLPSO-2 | ACLPSO-3 | CLPSO | ECLPSO | OLSO-L | AGPSO | FLGPSO-QIW | ALC-GPSO |
|----------|---------------------------|--------|----------|----------|----------|-------|--------|--------|-------|-------------|-----------|
| f₁       | Mean                      | 3.28E−11 | 1.85E−11 | 3.38E−11 | 2.19E−12 | 3.11E−14 | 2.74E−93 | 1.11E−38 | 1.70E−1 | 4.07E−48   | 1.67E−11 |
|          | SD                        | 6.11E−11 | 9.13E−11 | 5.01E−11 | 7.47E−12 | 5.98E−14 | 4.71E−93 | 1.28E−38 | 1.28E−1 | 1.12E−47   | 8.20E−11 |
| f₂       | Mean                      | 7.25E−37 | 1.14E−36 | 6.25E−36 | 1.51E−32 | 8.22E−10 | 4.24E−30 | 7.67E−22 | —     | —           | 1.16E−90 |
|          | SD                        | 6.84E−37 | 2.06E−36 | 6.52E−36 | 1.82E−32 | 3.84E−10 | 3.02E−10 | 5.63E−22 | —     | —           | 4.14E−90 |
| f₃       | Mean                      | 1.84     | 2.67     | 8.64     | 2.53     | 3.73E1   | 3.25E1   | 1.26     | 3.07E1 | 2.47E1      | 7.61      |
|          | SD                        | 1.63     | 2.95     | 8.04     | 4.16     | 2.04E1   | 1.63E1   | 1.40     | 1.35E1 | 5.07E−11    | 6.66      |
| f₄       | Mean                      | 7.87E−2  | 4.41E−2  | 1.07E−1  | 3.63E2   | 5.42E2   | 5.46E2   | —       | 3.87E2 | 5.27E1      | —         |
|          | SD                        | 8.72E−2  | 4.80E−2  | 1.43E−1  | 9.89E1   | 1.81E2   | 1.60E2   | —       | 1.39E2 | 2.66E1      | —         |
| f₅       | Mean                      | 7.87E−2  | 4.41E−2  | 1.07E−1  | 3.63E2   | 5.42E2   | 5.46E2   | —       | 3.87E2 | 5.27E1      | —         |
|          | SD                        | 8.72E−2  | 4.80E−2  | 1.43E−1  | 9.89E1   | 1.81E2   | 1.60E2   | —       | 1.39E2 | 2.66E1      | —         |
| f₆       | Mean                      | 3.11E−15 | 3.11E−15 | 3.11E−15 | 2.05E−4  | 5.10E−8  | 3.11E−15 | 4.14E−15 | 6.53E−2 | 4.77E−14    | 1.15E−14  |
|          | SD                        | 3.11E−15 | 3.11E−15 | 3.11E−15 | 2.05E−4  | 5.10E−8  | 3.11E−15 | 4.14E−15 | 6.53E−2 | 4.77E−14    | 1.15E−14  |
| f₇       | Mean                      | 3.82E−4  | 3.82E−4  | 3.82E−4  | 3.82E−4  | 3.82E−4  | 3.82E−4  | 3.82E−4  | —     | —           | 2.10E1    |
|          | SD                        | 9.15E−7  | 4.26E−10 | 4.51E−10 | 4.66E−7  | 1.54E−12 | 5.04E−12 | 0       | —     | —           | 5.41E1    |
| f₈       | Mean                      | 4.05E2   | 4.06E2   | 5.05E2   | 4.25E2   | 4.13E2   | 5.22E2   | 2.60E3   | —     | —           | 2.18E1    |
|          | SD                        | 1.60E1   | 9.45     | 6.17E1   | 4.09E1   | 2.26E1   | 3.07E1   | 2.40E1   | —     | —           | 4.04E1    |
| f₉       | Mean                      | 3.52     | 5.10     | 5.22     | 4.16E1   | 2.86E1   | 2.50E1   | 5.34E1   | 4.52E2 | 3.76E2      | —         |
|          | SD                        | 1.88     | 2.27     | 3.05     | 6.76     | 4.00     | 4.58     | 1.34E1   | 1.62E1 | 8.88        | —         |
| f₁₀      | Mean                      | 2.53E2   | 1.27E3   | 3.56E2   | 8.83E2   | 1.29E3   | 1.18E3   | 3.13E3   | —     | —           | —         |
|          | SD                        | 2.09E2   | 2.09E2   | 4.89E2   | 4.17E2   | 1.38E2   | 1.17E2   | 1.24E3   | —     | —           | —         |
| f₁₁      | Mean                      | 5.33E−17 | 1.80E−11 | 1.42E−13 | 1.14E−4  | 5.76E−5  | 1.49E−3  | 4.19E−8  | 1.01E2 | 1.40        | —         |
|          | SD                        | 6.51E−17 | 8.98E−11 | 7.08E−13 | 1.71E−4  | 5.25E−5  | 4.37E−3  | 2.06E−7  | 7.50E1 | 4.35E−1     | —         |
| f₁₂      | Mean                      | 3.82E−15 | 7.45E−2  | 3.96E−15 | 3.68E−15 | 1.03E−6  | 3.96E−15 | —       | 2.06E1 | 2.10E1      | —         |
|          | SD                        | 1.45E−15 | 2.58E−1  | 1.55E−15 | 1.33E−15 | 1.54E−6  | 1.55E−15 | —       | 1.24E−1 | 5.25E−2     | —         |
on all the functions except $f_3$ and $f_{16}$. ACLPSO-1 finds an unsatisfactory local optimum on $f_{14}$ for all the runs, ACLPSO-2 cannot locate the global optimum or a near-optimum solution on $f_5$, $f_6$, $f_{12}$, and $f_{14}$ for all or some of the runs, and ACLPSO-3 is not able to effectively solve $f_4$, $f_6$, $f_{12}$, $f_{13}$, and $f_{14}$ as the solutions found are unsatisfactory for all or some of the runs. The comparisons among ACLPSO, ACLPSO-1, ACLPSO-2, and ACLPSO-3 validate that the repairing of a particle’s dimension position when trespassing the dimensional search space, the dimensional independent and adaptive inertia weights and acceleration coefficients, and the dimensional independent and adaptive learning probabilities is appropriate to be employed by ACLPSO.

### Table 5: Two-tailed $t$-test results for the comparison of the global best fitness value results between ACLPSO with the best ($s, v$) combination and ECLPSO on all the benchmark functions.

| Function | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|
| $t$-test | —     | $3.01E-7$ | $1.56E-9$ | $6.68E-15$ | —     | 0.20  | 0.33  | —     |
| $t$-test | 0.06  | $1.21E-18$ | —     | $1.77E-17$ | $8.49E-21$ | $3.01E-21$ | 0.10  | 0.74  |

### Table 6: Mean and standard deviation execution time results of ACLPSO with the best ($s, v$) combination, ECLPSO, and CLPSO on all the benchmark functions.

| Function | Mean (ms) | SD (ms) |
|----------|-----------|---------|
| ACLPSO   | 784.20    | 32.11   |
| ECLPSO   | 732.20    | 24.49   |
| CLPSO    | 322.68    | 16.75   |

### Table 7: Mean and standard deviation global best fitness value results of ACLPSO with other parameter settings on $f_3$, $f_4$, $f_{12}$, and $f_{14}$.

| Function | Parameter setting | Global best fitness value |
|----------|-------------------|---------------------------|
| $f_3$    | $N = 40; s = 0.1, v = 0.2$ | 1.89E1                    |
|         | $N = 40; s = 0.1, v = 0.1$ | 1.44E1                    |
| $f_4$    | $N = 20; s = 0.1, v = 0.3$ | 6.32E2                    |
|         | $N = 40; s = 0.3, v = 0.3$ | 1.72E2                    |
| $f_{12}$ | $N = 40; s = 0.1, v = 0.1$ | 2.15E2                    |
|         | $N = 40; s = 0.1, v = 0.2$ | 4.18E2                    |
| $f_{14}$ | $N = 20; s = 1.1, v = 0.3$ | 7.56E2                    |
|         | $N = 40; s = 0.9, v = 0.3$ | 4.25E2                    |
|         | $N = 40; s = 0.3, v = 0.3$ | 5.61E2                    |

### Table 8: Mean and standard deviation global best fitness value and execution time results of CLPSO with other parameter settings on $f_1$ and $f_2$.

| Function | Parameter setting | Global best fitness value | Execution time (ms) |
|----------|-------------------|---------------------------|---------------------|
| $f_1$    | $N = 40$; the number of FEs is 500,000 | $7.13E-29$ | 815.32 |
|         | $N = 80$; the number of FEs is 500,000 | $3.13E-19$ | 805.48 |
| $f_2$    | $N = 40$; the number of FEs is 500,000 | $9.39E-22$ | 844.08 |
|         | $N = 80$; the number of FEs is 500,000 | $8.38E-13$ | 861.88 |
Figure 2: The changes of the global best fitness value during the search process of ACLPSO with the best \((s, v)\) combination and in the best run. (a) \(f_2\). (b) \(f_3\). (c) \(f_4\). (d) \(f_{12}\). (e) \(f_{13}\). (f) \(f_{14}\).
It can be seen from Table 4 that ACLPSO, in general, outperforms the literature PSO variants including CLPSO, ECLPSO, OLPSO-L, AGPSO, FLGPSO-QIW, and ALCGPSO in terms of the mean global best fitness value results. CLPSO and ECLPSO both fail to obtain the global optimum or a near-optimum solution on $f_{13}$, $f_{14}$, $f_{15}$, and $f_{16}$ for all the runs. ECLPSO fails additionally on $f_{10}$ with the finding of an unsatisfactory local optimum for all the runs. As the mean and SD global best fitness value results of OLPSO-L, AGPSO, FLGPSO-QIW, and ALCGPSO are directly copied from [7, 24], the symbol “—” represents an unavailable result in Table 4. We can see unsatisfactory mean global best fitness value results for OLPSO-L on $f_{10}$, $f_{13}$, and $f_{14}$, for AGPSO on $f_{3}$, $f_{4}$, $f_{5}$, $f_{12}$, $f_{13}$, and $f_{15}$, and $f_{16}$, for FLGPSO-QIW on $f_{3}$, $f_{4}$, $f_{5}$, $f_{6}$, $f_{12}$, $f_{13}$, and $f_{15}$, and $f_{16}$, and for ALC-GPSO on $f_{3}$, $f_{4}$, and $f_{10}$. The accuracies of ALCPSO’s mean global best fitness value results are the best on $f_{1}$, $f_{2}$, $f_{3}$, $f_{6}$, $f_{9}$, $f_{10}$, $f_{11}$, $f_{12}$, $f_{13}$, $f_{14}$, and $f_{15}$. The symbol “—” also appears in Table 5 and denotes a division by zero error. The $t$-test results are less than 0.05 on $f_{3}$, $f_{4}$, $f_{5}$, $f_{10}$, $f_{12}$, $f_{13}$, and $f_{14}$, and therefore, the global best fitness value results of ALCPSO are significantly different from those of ECLPSO on these 7 functions according to the statistics perspective. Based on the observations, though ECLPSO enhances the exploitation and convergence performance of CLPSO, the exploration performance of ECLPSO is as weak as that of CLPSO on some complex problems; and ALCPSO successfully addresses significantly bettering the exploration performance of ECLPSO, and ALCPSO is still good at exploitation and convergence, owing to the adoption of the dimension-independent and adaptive maximum velocities, inertia weights, acceleration coefficients, and learning probabilities.

ACLPSO and ECLPSO take advantage of more strategies than CLPSO to significantly achieve better accuracy; as a result, the mean execution time results of ALCPSO and ECLPSO are more than those of CLPSO on all the functions in Table 6. The differences are most noticeable on $f_{1}$, $f_{2}$, $f_{3}$, $f_{6}$, $f_{11}$, $f_{15}$, and $f_{16}$ and the mean execution time of ALCPSO and that of ECLPSO are both about 300 to 500 ms more than that of CLPSO on each of these 7 functions. It must be pointed out that, for many real-world complex problems, the function evaluation or, in other words, evaluating the fitness of a position could be much time consuming, and accordingly, the execution time difference caused by more strategies of ALCPSO and ECLPSO than CLPSO would become relatively very small.

The mean execution time results of ALCPSO are slightly more than those of ECLPSO on $f_{1}$, $f_{2}$, $f_{3}$, $f_{6}$, $f_{11}$, and $f_{13}$, considerably more on $f_{4}$, $f_{12}$, $f_{14}$, $f_{15}$, and $f_{16}$, slightly less on $f_{5}$ and $f_{6}$, and considerably less on $f_{8}$ and $f_{10}$, meaning that the dimensional independent and adaptive maximum velocities, inertia weights, acceleration coefficients, and learning probabilities essentially do not increase execution time. With respect to ALCPSO, ECLPSO, and CLPSO, the mean execution time spent on a rotated function is more than that consumed on the corresponding original function, as can be observed from the pair of the mean execution time results on $f_{13}$ and $f_{5}$, the pair of $f_{14}$ and $f_{8}$, the pair of $f_{15}$ and $f_{6}$, and the pair of $f_{16}$ and $f_{2}$. The SD execution time results of ALCPSO, ECLPSO, and CLPSO are rather small as compared to the mean execution time results on all the functions.

The number of particles $N$ is also a key parameter for ALCPSO. As can be seen from Table 7, setting $N$ as 20 on $f_{4}$ and $f_{13}$ renders much worse mean and SD global best fitness value results, and ALCPSO cannot find the global optimum or a near-optimum solution for all or most of the runs because less particles lead to insufficient diversity. We can also observe from Table 7 that letting the learning probability-based coefficient $\nu$ to be more than 0.05 on $f_{4}$, the normative interval scaling coefficient $s$ to be more than 0.1 on $f_{4}$, $\nu$ to be less than 0.3 on $f_{12}$, and $s$ to be less than 1.1 on $f_{14}$ causes ALCPSO to be unable to find the optimum or a near-optimum solution for all or some of the runs. The mean and SD global best fitness value and execution time results of CLPSO on $f_{1}$ and $f_{2}$ in Table 8 indicate that simply by increasing the number of FEs to 500,000 or even increasing $N$ to 80, CLPSO still cannot achieve high-accuracy mean global best fitness value as ECLPSO, while the mean execution time results of CLPSO are close to those of ECLPSO. The PbE and ALP strategies as well as the empirical values chosen for $N$ and the best $(s, \nu)$ combination are thus appropriate.

As shown in Figure 2, ECLPSO is liable to get stuck in premature stagnancy on $f_{3}$, $f_{4}$, $f_{12}$, $f_{13}$, and $f_{14}$. The exploitation performance of ALCPSO is quite better than that of ECLPSO on $f_{2}$. ALCPSO escapes from an unsatisfactory local optimum at the early stage of the search process on $f_{4}$ and $f_{14}$ at the middle stage on $f_{3}$ and $f_{12}$, and at the late stage on $f_{13}$. According to Tables 2 and 3, ALCPSO takes the same best $s$ values on $f_{3}$, $f_{4}$, and $f_{14}$, and the best $\nu$ values on $f_{4}$, $f_{12}$, $f_{13}$, and $f_{14}$, and the same best $s$ values on $f_{13}$ and $f_{14}$; the mean number of exploitation valid dimensions’ results is 30 for the best $(s, \nu)$ combinations of ALCPSO on $f_{4}$ and $f_{12}$, slightly smaller than 30 on $f_{13}$ and $f_{14}$, and considerably smaller than 30 on $f_{3}$. It is challenging to develop a unified setting for $s$ and $\nu$ based on the generation counter and the dimensional normative intervals.

6. Conclusions

In this paper, we have proposed ALCPSO for the purpose of further significantly mending the exploration performance of ECLPSO. ALCPSO introduces an independent inertia weight and an independent acceleration coefficient corresponding to each dimension and an independent learning probability for each particle on each dimension. ALCPSO determines the normative interval with respect to each dimension in each generation. Based on the dimensional normative intervals, ALCPSO adaptively adjusts the dimensional independent maximum velocities, inertia weights, acceleration coefficients, and learning probabilities. Experiments on a variety of unimodal, multimodal, shifted, rotated, and shifted rotated benchmark functions have demonstrated that ALCPSO successfully addresses exploration as well as exploitation and convergence as ALCPSO is able to derive the global optimum or a near-optimum solution on all the functions for all the runs with the normative interval scaling coefficient and the learning probability-based coefficient appropriately set. ALCPSO is a promising
metaheuristic for global optimization. In the future, we plan to dig out more critical information inherently embodied in the search experience of the particles and try to develop a high-performance PSO variant based on ACLPSO with a unified parameter setting that works well on most global optimization problems and complex real-world applications, e.g., optimal operation of power systems [50, 51].

**Data Availability**
The data used to support the findings of this study are available from the corresponding author upon request.

**Conflicts of Interest**
The authors declare that they have no conflicts of interest.

**Acknowledgments**
This work was financially supported by the National Natural Science Foundation of China Project (61703199), the Jiangxi Natural Science Foundation Basic Research Project (2020JM-278), and the Central Universities Fundamental Research Foundation Project (GK202003006).

**References**

[1] A. Banks, J. Vincent, and C. Anyakoha, “A review of particle swarm optimization. Part I: background and development,” Natural Computing, vol. 6, no. 4, pp. 467–484, 2007.

[2] A. Banks, J. Vincent, and C. Anyakoha, “A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications,” Natural Computing, vol. 7, no. 1, pp. 109–124, 2008.

[3] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” in Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948, Perth, WA, Australia, November 1995.

[4] Y. H. Shi and R. Eberhart, “A modified particle swarm optimizer,” in Proceedings of the IEEE Congress on Evolutionary Computation, pp. 69–73, Anchorage, AK, USA, May 1998.

[5] J. Kennedy and R. Mendes, “Population structure and particle swarm performance,” in Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1671–1676, Honolulu, HI, USA, May 2002.

[6] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, “Comprehensive learning particle swarm optimizer for global optimization of multimodal functions,” IEEE Transactions on Evolutionary Computation, vol. 10, no. 3, pp. 281–295, 2006.

[7] Z.-H. Zhan, J. Zhang, Y. Li, and Y.-H. Shi, “Orthogonal learning particle swarm optimization,” IEEE Transactions on Evolutionary Computation, vol. 15, no. 6, pp. 832–847, 2011.

[8] X. Yu and X. Zhang, “Enhanced comprehensive learning particle swarm optimization,” Applied Mathematics and Computation, vol. 242, pp. 265–276, 2014.

[9] Z. H. Zhan, “Adaptive particle swarm optimization,” IEEE Transactions on Systems, Man, Cybernetics, Part B, vol. 39, no. 6, pp. 1362–1381, 2009.

[10] Z. Beheshti, S. M. H. Shamsuddin, and S. Hasan, “MPSO: median-oriented particle swarm optimization,” Applied Mathematics and Computation, vol. 219, no. 11, pp. 5817–5836, 2013.

[11] W.-N. Chen, J. Zhang, Y. Lin et al., “Particle swarm optimization with an aging leader and challengers,” IEEE Transactions on Evolutionary Computation, vol. 17, no. 2, pp. 241–258, 2013.

[12] M. Hu, T. Wu, and J. D. Weir, “An adaptive particle swarm optimization with multiple adaptive methods,” IEEE Transactions on Evolutionary Computation, vol. 17, no. 5, pp. 705–720, 2013.

[13] W. H. Lim and N. A. Mat Isa, “Particle swarm optimization with adaptive time-varying topology connectivity,” Applied Soft Computing, vol. 24, pp. 623–642, 2014.

[14] X. Xia, J. Liu, and Z. Hu, “An improved particle swarm optimizer based on tabu detecting and local learning strategy in a shrunk search space,” Applied Soft Computing, vol. 23, pp. 76–90, 2014.

[15] Y. Tang, Z. Wang, and J.-s. Fang, “Feedback learning particle swarm optimization,” Applied Soft Computing, vol. 11, no. 8, pp. 4713–4725, 2011.

[16] M.-S. Leu, M.-F. Yeh, and S.-C. Wang, “Particle swarm optimization with grey evolutionary analysis,” Applied Soft Computing, vol. 13, no. 10, pp. 4047–4062, 2013.

[17] H. Wang, W. Wang, and Z. Wu, “Particle swarm optimization with adaptive mutation for multimodal optimization,” Applied Mathematics and Computation, vol. 221, pp. 296–305, 2013.

[18] W. H. Lim and N. A. M. Isa, “Particle swarm optimization with dual-level task allocation,” Engineering Applications of Artificial Intelligence, vol. 38, pp. 88–110, 2015.

[19] N. Netjinda, T. Achalakul, and B. Sirinaovakul, “Particle swarm optimization inspired by starling flock behavior,” Applied Soft Computing, vol. 35, pp. 411–422, 2015.

[20] K. Tang, Z. Li, L. Luo, and B. Liu, “Multi-strategy adaptive particle swarm optimization for numerical optimization,” Engineering Applications of Artificial Intelligence, vol. 37, pp. 9–19, 2015.

[21] L. N. Vitorino, S. F. Ribeiro, and C. J. A. Bastos-Filho, “A mechanism based on artificial bee colony to generate diversity in particle swarm optimization,” Neurocomputing, vol. 148, pp. 39–45, 2015.

[22] X.-P. Wei, J.-X. Zhang, D.-S. Zhou, and Q. Zhang, “Multi-swarm particle swarm optimization with transfer of the best particle,” Computational Intelligence and Neuroscience, vol. 2015, Article ID 904713, 2015.

[23] R. Jensl and G. W. Jiji, “An enhanced particle swarm optimization with levy flight for global optimization,” Applied Soft Computing, vol. 43, pp. 248–261, 2016.

[24] A. Meng, Z. Li, H. Yin, S. Chen, and Z. Guo, “Accelerating particle swarm optimization using crisscross search,” Information Sciences, vol. 329, pp. 52–72, 2016.

[25] Y. X. Shen, L. Wei, C. Zheng, and J. Chen, “Particle swarm optimization with double learning patterns,” Computational Intelligence and Neuroscience, vol. 2016, Article ID 6510303, 2016.

[26] M. Taherkhani and R. Safabakhsh, “A novel stability-based adaptive inertia weight for particle swarm optimization,” Applied Soft Computing, vol. 38, pp. 281–295, 2016.

[27] C. F. Wang and K. Liu, “A novel particle swarm optimization algorithm for global optimization,” Computational Intelligence and Neuroscience, vol. 2016, Article ID 9482073, 2016.

[28] L. Wang, B. Yang, and J. Orchard, “Particle swarm optimization using dynamic tournament topology,” Applied Soft Computing, vol. 48, pp. 584–596, 2016.
[29] X. Y. Lv, Y. Wang, J. Deng, G. Zhang, and L. Zhang, “Improved particle swarm optimization algorithm based on last-eliminated principle and enhanced information sharing,” *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 5025672, 2018.

[30] R. Lan, L. Zhang, Z. Tang, Z. Liu, and X. Luo, “A hierarchical sorting swarm optimizer for large-scale optimization,” *IEEE Access*, vol. 7, pp. 40625–40635, 2019.

[31] X. Xia, Y. Tang, B. Wei, and L. Gui, “Dynamic multi-swarm particle swarm optimization based on elite learning,” *IEEE Access*, vol. 7, pp. 184849–184865, 2019.

[32] Y. Hou, G.-S. Hao, Y. Zhang, F. Gu, X. Wang, and T.-T. Zhang, “A molecular interactions-based social learning particle swarm optimization algorithm,” *IEEE Access*, vol. 8, pp. 135661–135674, 2020.

[33] J. J. Liang and P. N. Suganthan, “Adaptive comprehensive learning particle swarm optimizer with history learning,” in *Proceedings of the Asia-Paciﬁc Conference on Simulated Evolution and Learning*, pp. 213–220, Hefei, China, October 2006.

[34] J. C. Ni, “A novel memetic algorithm based on the comprehensive learning PSO,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1–8, Brisbane, Australia, June 2012.

[35] Y. J. Zheng, H. F. Ling, and Q. Guan, “Adaptive parameters for a modiﬁed comprehensive learning particle optimizer,” *Mathematical Problems in Engineering*, vol. 2012, Article ID 207318, 2012.

[36] 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 Systems with Applications*, vol. 41, no. 16, pp. 7536–7548, 2014.

[37] Q. Qin, S. Cheng, Q. Zhang, Y. Wei, and Y. Shi, “Multiple strategies based orthogonal design particle swarm optimizer for numerical optimization,” *Computers & Operations Research*, vol. 60, pp. 91–110, 2015.

[38] Y. Hu, Y. Ding, K. Hao, L. Ren, and H. Han, “An immune orthogonal learning particle swarm optimisation algorithm for routing recovery of wireless sensor networks with mobile sink,” *International Journal of Systems Science*, vol. 45, no. 3, pp. 337–350, 2014.

[39] N. Lynn and P. N. Suganthan, “Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation,” *Swarm and Evolutionary Computation*, vol. 24, pp. 11–24, 2015.

[40] Y. Cao, H. Zhang, W. Li, M. Zhou, Y. Zhang, and W. A. Chaovilittongse, “Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 718–731, 2019.

[41] K. Zhang, Q. Huang, and Y. Zhang, “Enhancing comprehensive learning particle swarm optimization with local optima topology,” *Information Sciences*, vol. 471, pp. 1–18, 2019.

[42] Q. Zhao and C. Li, “Two-stage multi-swarm particle swarm optimizer for unconstrained and constrained global optimization,” *IEEE Access*, vol. 8, pp. 124905–124927, 2020.

[43] X. Yu and X. Q. Zhang, “Multiswarm comprehensive learning particle swarm optimization for solving multiobjective optimization problems,” *PLoS One*, vol. 12, no. 2, Article ID e0172033, 2017.

[44] H. Li and Q. F. Zhang, “Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[45] E. García-Gonzalo and J. L. Fernández-Martínez, “Convergence and stochastic stability analysis of particle swarm optimization variants with generic parameter distributions,” *Applied Mathematics and Computation*, vol. 249, pp. 286–302, 2014.

[46] D. P. Tian, “A review of convergence analysis of particle swarm optimization,” *International Journal of Grid and Distributed Computing*, vol. 6, no. 6, pp. 117–128, 2013.

[47] P. N. Suganthan, “Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,” KanGAL Report, 2005.

[48] S. Kok and C. Sandrock, “Locating and characterizing the stationary points of the extended Rosenbrock function,” *Evolutionary Computation*, vol. 17, no. 3, pp. 437–453, 2009.

[49] R. Salomon, “Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions: a survey of some theoretical and practical aspects of genetic algorithms,” *BioSystems*, vol. 39, no. 3, pp. 263–278, 1996.

[50] X. Zhang, X. Yu, and H. Qin, “Optimal operation of multi-reservoir hydropower systems using enhanced comprehensive learning particle swarm optimization,” *Journal of Hydro-Environment Research*, vol. 10, pp. 50–63, 2016.

[51] M. Nazari-Heris, A. F. Babaei, B. Mohammadi-Ivatloo, and S. Asadi, “Improved harmony search algorithm for the solution of non-linear non-convex short-term hydrothermal scheduling,” *Energy*, vol. 151, pp. 226–237, 2018.