Hindawi
Computational Intelligence and Neuroscience
Volume 2021, Article ID 6628564, 16 pages
https://doi.org/10.1155/2021/6628564

Research Article

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

Xiang Yu[1] **and Yu Qiao**[2]

[1] *Provincial Key Laboratory for Water Information Cooperative Sensing and Intelligent Processing, Nanchang Institute of Technology, Nanchang, Jiangxi 330099, China*
[2] *School 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

Academic Editor: Raşit Köker

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. Like ECLPSO, a normative interval bounded by the minimum and maximum personal best positions is determined with respect to each dimension in each generation. 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 $V_i = \{V_{i,1}, V_{i,2}, \ldots, V_{i,D}\}$, a D-dimensional position $P_i = \{P_{i,1}, P_{i,2}, \ldots, P_{i,D}\}$, 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:

$$V_{i,d} = wV_{i,d} + ar\left(B_{i,d} - P_{i,d}\right) + bs\left(G_d - P_{i,d}\right), \tag{1}$$

$$P_{i,d} = P_{i,d} + V_{i,d}, \tag{2}$$

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]; $B_i = \left\{B_{i,1}, B_{i,2}, \ldots, B_{i,D}\right\}$ 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 $L_i = \left\{L_{i,1}, L_{i,2}, \ldots, L_{i,D}\right\}$ 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:

$$V_{i,d} = wV_{i,d} + ar\left(B_{i,d} - P_{i,d}\right) + bs\left(L_{i,d} - P_{i,d}\right). \tag{3}$$

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.

$$V_{i,d} = wV_{i,d} + ar\left(E_{i,d} - P_{i,d}\right), \tag{4}$$

where $E_i = \left\{E_{i,1}, E_{i,2}, \ldots, E_{i,D}\right\}$ is i's exemplar position. In CLPSO, i is additionally associated with a fixed learning probability that controls $E_{i,d} = B_{i,d}$ or $B_{j,d}$ 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 $\underline{B_d}$ and $\overline{B_d}$ 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:

$$\begin{aligned} \underline{B}_d &= \min\left\{B_{1,d}, B_{2,d}, \ldots, B_{N,d}\right\}, \\ \overline{B}_d &= \max\left\{B_{1,d}, B_{2,d}, \ldots, B_{N,d}\right\}, \end{aligned} \tag{5}$$

where N is the number of particles. $[\underline{B}_d, \overline{B}_d]$ is termed as the normative interval of dimension d. Let the search space on dimension d be $[\underline{P}_d, \overline{P}_d]$ with $\underline{P_d}$ being the lower bound and $\overline{P}_d$ being the upper bound; ECLPSO deems that when $[\underline{B}_d, \overline{B}_d]$ becomes indeed small (i.e., no greater than 1% of $[\underline{P}_d, \overline{P}_d]$ 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 $\overline{V}_d$, i.e.,

$$V_{i,d} = \begin{cases} \overline{V}_d, \text{if } V_{i,d} > \overline{V}_d, \\ -\overline{V}_d, \text{else if } V_{i,d} < -\overline{V}_d, \\ V_{i,d}, \text{otherwise.} \end{cases} \tag{6}$$

If $\overline{V}_d$ was too large, the particles might miss some promising solutions on dimension d; on the contrary, too small $\overline{V}_d$ would slow down the search process on dimension d. $\overline{V}_d$ is fixed at 20% of the dimensional search space $[\underline{P}_d, \overline{P}_d]$ 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 fight trajectory and search behavior of all the particles in ECLPSO are directly affected by each dimension d's maximum velocity $\overline{V}_d$, 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]; nonuniform 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 i 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 a is a constant value equivalent to 1.5. Let $k_{\max}$ be the predefined maximum number of generations; w is updated in each generation k according to the following equation:

$$w = w_{\max} - \frac{k}{k_{\max}} \left(w_{\max} - w_{\min}\right), \tag{7}$$

where $w_{\max} = 0.9$ and $w_{\min} = 0.4$ 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.

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

where $L_{\max} = 0.5$ is the maximum learning probability and $L_{\min} = 0.05$ 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 $[\underline{P}_d, \overline{P}_d]$ 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 $[\underline{B}_d, \overline{B}_d]$ 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):

$$V_{i,d} = w_{PbE} V_{i,d} + a_{PbE} r \left(E_{i,d} + c \left(\frac{\underline{B}_d + \overline{B}_d}{2} - E_{i,d} \right) - P_{i,d} \right), \tag{9}$$

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 $E_{i,d}$ plus a perturbation term $c((\underline{B}_d + \overline{B}_d)/2 - E_{i,d})$ 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 $\overline{V}_d$.

The minimum learning probability $L_{\min}$ is fixed at 0.05. As expressed in equation (10), the maximum learning probability $L_{\max}$ logarithmically increases in each generation k:

$$L_{\max} = L_{\min} + h + q\,\log_{(D+1)}\left(M_k + 1\right), \tag{10}$$

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. $L_{\max}$ is small (i.e., 0.3) when $M_k = 0$ and benefits initial exploration. $L_{\max}$ 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.,

$$L_i = L_{\min} + \left(L_{\max} - L_{\min}\right)\frac{\exp\left(10\left(T_i - 1\right)/N - 1\right) - 1}{\exp\left(10\right) - 1}, \tag{11}$$

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 $[\underline{P}_d, \overline{P}_d]$ and all the particles' dimensional personal best positions are scattered (i.e., the dimensional normative interval $[\underline{B}_d, \overline{B}_d]$ is large), then it would be difficult for the swarm of particles to locate X_D^*; this is because the dimensional maximum velocity $\overline{V}_d$ updated by equation (4) is restricted to be 20% of $[\underline{P}_d, \overline{P}_d]$. 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 $[\underline{P}_d, \overline{P}_d]$, 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 $\overline{P}_d - \underline{P}_d$; 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 $E_{i,d}$, i's dimensional velocity update is influenced by two forces, i.e., the inertia force $wV_{i,d}$ and the exemplar force $ar\,(E_{i,d}$–$P_{i,d})$; 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 $\underline{P}_d$; $E_{i,d}$ 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 bound 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 Schwefel's function, and rotated Rastrigin's function, for all the runs as reported in [8]. Therefore, $\overline{V}_d$ should not be fixed at 20% of $[\underline{P}_d, \overline{P}_d]$. We propose to adaptively adjust $\overline{V}_d$ in each generation according to the following equation:

$$\overline{V}_d = s\left(\overline{B}_d - \underline{B}_d\right), \tag{12}$$

where s is the scaling coefficient and is a positive value. $\overline{V}_d$ is positively related with the dimensional normative interval's size $\overline{B}_d - \underline{B}_d$. When $[\underline{B}_d, \overline{B}_d]$ is large, $\overline{V}_d$ is large and contributes to timely flight for getting close to X_D^*; on the contrary, $\overline{V}_d$ is small when $[\underline{B}_d, \overline{B}_d]$ 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 $\overline{P}_d$ and is infeasible, and $E_{i,d}$ is around $\underline{P}_d$; 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 $[\underline{B}_d, \overline{B}_d]$is greater than 1% of the dimensional search space $[\underline{P}_d, \overline{P}_d]$ or greater than 2 as follows:

$$w_d = u\frac{\overline{B}_d - \underline{B}_d}{\overline{P}_d - \underline{P}_d} + (1-u)\left(w_{\max} - \frac{k}{k_{\max}}\left(w_{\max} - w_{\min}\right)\right), \tag{13}$$

$$w_d = \begin{cases} w_{\max}, \text{if } w_d > w_{\max}, \\ w_{\min}, \text{else if } w_d < w_{\min}, \\ w_d, \text{otherwise}, \end{cases} \tag{14}$$

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

$$1 - w_d \geq 0 \text{ and } w_d + 1 - ra_d \geq 0. \tag{15}$$

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

$$a_d = w_d + 1. \tag{16}$$

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_k 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:

$$L_{i,d} = \nu \log_{k_{\max}} k + \frac{\overline{B}_d - \underline{B}_d}{\overline{P}_d - \underline{P}_d} \frac{\exp(D(T_i - 1)/N - 1) - 1}{\exp(D) - 1}, \tag{17}$$

$$L_{i,d} = \begin{cases} L_{\max}, \text{ if } L_{i,d} > L_{\max}, \\ L_{\min}, \text{ else if } L_{i,d} < L_{\min}, \\ L_{i,d}, \text{ otherwise,} \end{cases} \tag{18}$$

where $L_{\min} = 0.05$; $L_{\max} = 0.75$; and ν is the learning probability-based coefficient and is a positive number no greater than 1. The term $\log_{k_{\max}} k$ grows logarithmically with the generator counter k in order to facilitate convergence. The term $(\overline{B}_d - \underline{B}_d)/(\overline{P}_d - \underline{P}_d)$, being positively related with $\overline{B}_d - \underline{B}_d$, also benefits convergence when the dimensional normative interval $[\underline{B}_d, \overline{B}_d]$ 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 $[\underline{P}_d, \overline{P}_d]$, 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 $W_i = 0$, 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 $[\underline{B}_d, \overline{B}_d]$, and update the dimensional maximum velocity $\overline{V}_d$ according to equation (12)

Step 4: for each particle i, calculate i's fitness rank T_i; and if $W_i\%\ (g + 1) = 0$, reset $W_i = 1$ 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 $[\underline{B}_d, \overline{B}_d]$ 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 $[\underline{P}_d, \overline{P}_d]$

Step 6: for each particle i, calculate $f(P_i)$; if $f(P_i) \geq f(B_i)$, update $W_i = W_i + 1$; 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 $O(k_{\max}(N\log N + ND))$ 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

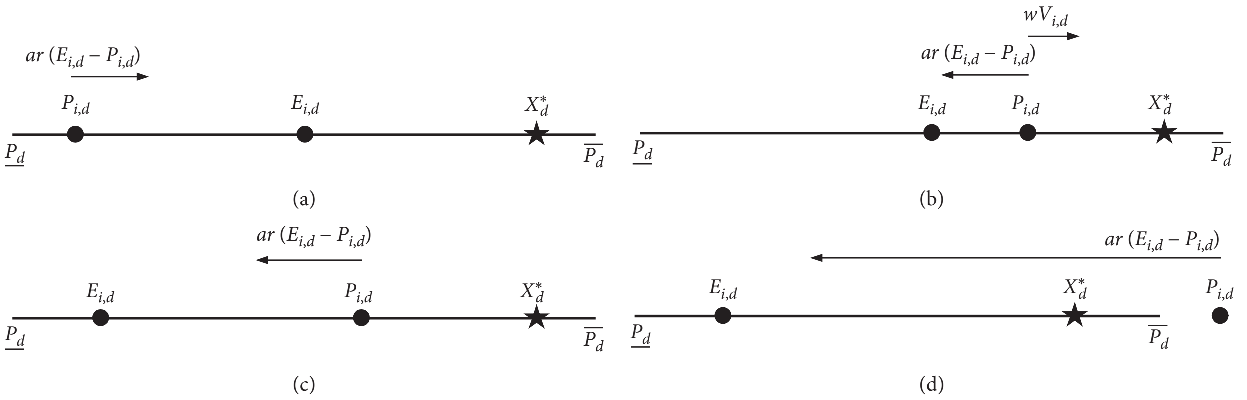


Figure 1: Illustration of why it is difficult for particle i to reach the dimensional global optimum X_D^* on dimension d. (a) $P_{i,d}$ and X_D^* are close to different bounds of $[\underline{P}_d, \overline{P}_d]$, and $E_{i,d}$ is located in between $P_{i,d}$ and X_D^*. (b) i flies past $E_{i,d}$. (c) X_D^* is around $\overline{P}_d$, $P_{i,d}$ is not that far from X_D^*, and $E_{i,d}$ is close to $\underline{P}_d$. (d) X_D^* is around $\overline{P}_d$, $P_{i,d}$ trespasses $\overline{P}_d$, and $E_{i,d}$ is close to $\underline{P}_d$.

external memory. The operating system is 64 bit Windows 10.

16 commonly studied 30-dimensional functions [6–8, 24, 47] are used in this paper for benchmarking ACLPSO and other PSO variants. The name, the expression, the global optimum, the function value of the global optimum, the search space, and the initialization space of each function are listed in Table 1. The functions are classified into 5 categories, namely, unimodal, multimodal, shifted, rotated, and shifted rotated. Rosenbrock's function f_3 is unimodal in a 2-dimensional or 3-dimensional search space, but is multimodal in higher-dimensional cases [48]; it features a narrow valley from perceived local optima to the global optimum. With the incorporation of the cosine term $\cos(2\pi X_d)$, there are a large number of regularly distributed local optima for Rastrigin's function f_5. Ackley's function f_7 has a deep global optimum and many minor local optima. Griewank's function f_8 contains a cosine multiplication term $\prod_{d=1}^{D}\cos(X_d/\sqrt{d})$ that causes linkages among the decision variables; f_8 is similar to f_5 in terms of many regularly distributed local optima. Schwefel's function f_9 has a global optimum that is distant from the local optima. With respect to the unimodal and multimodal functions f_1 to f_9, the dimensional values of the global optimum are the same on all the dimensions. A shifted function shifts the global optimum X^* to a vector Z that can be different on each dimension. A rotated function multiplies the original decision vector X by an orthogonal matrix O to get a rotated decision vector $Y = XO$; because of the rotation, if one dimension of X changes, all the dimensions of Y get affected. A shifted rotated function is both shifted and rotated. The shifted global optima of the shifted functions f_{10} to f_{12} and the shifted rotated function f_{16} can be found in [47]. The orthogonal matrices of the rotated functions f_{13} to f_{15} and the shifted rotated function f_{16} are generated by Salomon's method [49]. The initialization spaces of the functions f_1, f_2, f_4, f_5, f_6, f_7, f_8, f_{13}, and f_{15} are intentionally set to be asymmetric.

We conduct experiments to investigate the following 3 issues: (1) what are the key parameters of ACLPSO and how 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 $P_{i,d}$ for each particle i on each dimension d if $P_{i,d}$ trespasses the dimensional search space $[\underline{P}_d, \overline{P}_d]$, 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, FLGPSO-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, FLGPSO-QIW, and ALC-GPSO in [9, 11, 15]. As we do not have the source codes of OLPSO-L, AGPSO, FLGPSO-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

Table 1: Benchmark functions.

Category	Benchmark function		X^*	$f(X^*)$	Search space	Initialization space
	Name	Expression				
Unimodal	Sphere	$f_1(X) = \sum_{d=1}^{D} X_D^2$	$\{0\}^D$	0	$[-100, 100]^D$	$[-100, 50]^D$
	Schwefel P2.22	$f_2(X) = \sum_{d=1}^{D} \lvert X_d \rvert + \prod_{d=1}^{D} \lvert X_d \rvert$	$\{0\}^D$	0	$[-10, 10]^D$	$[-10, 5]^D$
	Rosenbrock	$f_3(X) = \sum_{d=1}^{D-1} (100(X_{d+1} - X_D^2)^2 + (X_d - 1)^2)$	$\{1\}^D$	0	$[-10, 10]^D$	$[-10, 10]^D$
	Schwefel P1.2	$f_4(X) = \sum_{d=1}^{D} (\sum_{j=1}^{d} X_j)^2$	$\{0\}^D$	0	$[-100, 100]^D$	$[-100, 50]^D$
Multimodal	Rastrigin	$f_5(X) = 10\, D + \sum_{d=1}^{D} (X_D^2 - 10 \cos(2\pi X_d))$	$\{0\}^D$	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
	Noncontinuous Rastrigin	$f_6(X) = f_5(Y)$, $Y_d = \begin{cases} X_d, \text{ if } \lvert X_d \rvert < 0.5 \\ \text{Round}(2X_d)/2, \text{ otherwise} \end{cases}$	$\{0\}^D$	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
	Ackley	$f_7(X) = -20 \exp(-0.2\sqrt{(1/D)\sum_{d=1}^{D} X_D^2}) -$ $\exp((1/D)\sum_{d=1}^{D} \cos(2\pi X_d)) + 20 + e$	$\{0\}^D$	0	$[-32, 32]^D$	$[-32, 20]^D$
	Griewank	$f_8(X) = 1/4000\sum_{d=1}^{D} X_D^2 - \prod_{d=1}^{D} \cos(X_d \quad / \sqrt{d}) + 1$	$\{0\}^D$	0	$[-600, 600]^D$	$[-600, 200]^D$
	Schwefel	$f_9(X) = 418.9829D - \sum_{d=1}^{D} X_d \sin(\sqrt{\lvert X_d \rvert})$	$\{420.96\}^D$	0	$[-500, 500]^D$	$[-500, 500]^D$
Shifted	Shifted Rosenbrock	$f_{10}(X) = f_3(Y) + 390, Y = X - Z + 1$	Z	390	$[-100, 100]^D$	$[-100, 100]^D$
	Shifted Griewank	$f_{11}(X) = f_8(Y), Y = X - Z$	Z	0	$[-600, 600]^D$	$[-600, 600]^D$
	Shifted Schwefel P1.2	$f_{12}(X) = f_4(Y) - 450, Y = X - Z$	Z	−450	$[-100, 100]^D$	$[-100, 100]^D$
Rotated	Rotated Rastrigin	$f_{13}(X) = f_5(Y), Y = XO$	$\{0\}^D$	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
	Rotated Schwefel	$f_{14}(X) = f_9(Y)$, $Y_d = \begin{cases} Z_d \sin(\sqrt{\lvert Z_d \rvert}), \text{ if } \lvert Z_d \rvert \le 500 \\ 0, \text{ otherwise} \end{cases}, Z = (X - \{420.96\}_{1\times D})O + \{420.96\}_{1\times D}$	$\{420.96\}^D$	0	$[-500, 500]^D$	$[-500, 500]^D$
	Rotated Griewank	$f_{15}(X) = f_8(Y), Y = XO$	$\{0\}^D$	0	$[-600, 600]^D$	$[-600, 200]^D$
Shifted rotated	Shifted rotated Ackley	$f_{16}(X) = f_7(Y), Y = (X - Z)O$	Z	0	$[-32, 32]^D$	$[-32, 32]^D$

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	$1.23E-10$	$3.37E-105$	$2.87E-12$	$\mathbf{3.28E-115}$
	SD	$6.08E-10$	$9.36E-105$	$8.18E-12$	$\mathbf{6.11E-115}$
f_2	Mean	$1.48E-17$	$4.74E-22$	$3.09E-29$	$\mathbf{7.25E-37}$
	SD	$2.19E-17$	$9.70E-22$	$4.27E-29$	$\mathbf{6.84E-37}$
f_3	Mean	4.23	$6.56E1$	**1.84**	$2.61E1$
	SD	2.43	$2.49E1$	**1.63**	$2.87E1$
f_4	Mean	$1.56E4$	$1.71E4$	$3.39E2$	$\mathbf{7.87E-2}$
	SD	$3.44E3$	$8.24E3$	$8.84E1$	$\mathbf{8.72E-2}$
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	$1.92E-6$	$\mathbf{3.11E-15}$	$1.15E-8$	$3.25E-15$
	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	$\mathbf{3.82E-4}$	$4.29E2$	$2.23E2$	$5.87E2$
	SD	$\mathbf{9.15E-7}$	$2.71E2$	$1.38E2$	$2.04E2$
f_{10}	Mean	$\mathbf{4.05E2}$	$5.12E2$	$4.23E2$	$5.86E2$
	SD	$\mathbf{1.60E1}$	$1.02E2$	$2.32E1$	$2.56E2$
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	$4.19E3$	$3.26E2$	$6.37E2$	$\mathbf{-4.48E2}$
	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	$5.78E2$	$\mathbf{2.53E2}$	$1.35E3$	$1.32E3$
	SD	$4.00E2$	$\mathbf{2.09E2}$	$1.85E2$	$2.19E2$
f_{15}	Mean	$2.05E-3$	$\mathbf{5.33E-17}$	$1.65E-3$	$2.76E-3$
	SD	$2.71E-3$	$\mathbf{6.51E-17}$	$2.70E-3$	$5.91E-3$
f_{16}	Mean	$2.73E-5$	$3.96E-15$	$1.74E-6$	$\mathbf{3.82E-15}$
	SD	$9.24E-5$	$1.55E-15$	$6.94E-6$	$\mathbf{1.45E-15}$

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., $M_{k_{\max}}$) results of ACLPSO with 4 different (s, v) combinations on all the functions. The mean and SD global best fitness value results of ACLPSO with the best (s, v) combination, ACLPSO-1, ACLPSO-2, ACLPSO-3, CLPSO, ECLPSO, OLPSO-L, AGPSO, FLGPSO-QIW, and ALC-GPSO on all the functions are compared in Table 4. A two-tailed t-test with degree of freedom 48 and significance level 0.05 is performed between the global best fitness value results of ACLPSO with the best (s, v) combination and those of ECLPSO on each function, and the t-test results on all the functions are listed in Table 5. Table 6 gives the mean and SD execution time results of ACLPSO with the best (s, v) combination, ECLPSO, and CLPSO 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, III, IV, I, II, II, II, I, I, II, IV, II, 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_3 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

Table 3: Mean and standard deviation number of exploitation valid dimensions' results of ACLPSO with different (s, v) combinations on all the benchmark functions.

Function	Number of exploitation valid dimensions	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	26.4	30	24	30
	SD	9.95	0	12.25	0
f_2	Mean	30	30	30	30
	SD	0	0	0	0
f_3	Mean	1.08	29.4	6.04	30
	SD	5.40	1.83	10.13	0
f_4	Mean	0	0	0	30
	SD	0	0	0	0
f_5	Mean	30	30	10.48	29.72
	SD	0	0	3.08	0.46
f_6	Mean	0	0.2	0	0.12
	SD	0	0.41	0	0.44
f_7	Mean	27.56	30	28.8	30
	SD	8.30	0	6.00	0
f_8	Mean	2.2	30	2.36	29.52
	SD	6.09	0	5.16	1.16
f_9	Mean	22.8	30	5.24	30
	SD	13.08	0	7.90	0
f_{10}	Mean	6.36	29.76	7.4	29.96
	SD	11.55	0.72	12.12	0.2
f_{11}	Mean	0.68	30	1.6	29.6
	SD	3.00	0	3.88	0.82
f_{12}	Mean	0	6.28	0	30
	SD	0	6.28	0	0
f_{13}	Mean	0	25.44	0	8.72
	SD	0	8.73	0	7.83
f_{14}	Mean	3.28	28.36	0	27.12
	SD	5.12	3.62	0	7.37
f_{15}	Mean	3.92	30	2.52	29.56
	SD	5.10	0	4.74	1.08
f_{16}	Mean	26.4	30	27.6	30
	SD	9.95	0	8.31	0

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 exploitation 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_{10}, f_{13}, f_{14}, f_{15}, 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	OLPSO-L	AGPSO	FLGPSO-QIW	ALC-GPSO
f_1	Mean	3.28E − 115	1.85E − 113	3.38E − 114	2.19E − 12	3.11E − 14	2.74E − 93	1.11E − 38	1.70E − 1	4.07E − 48	1.67E − 161
	SD	6.11E − 115	9.13E − 113	5.01E − 114	7.47E − 12	5.98E − 14	4.71E − 93	1.28E − 38	1.28E − 1	1.12E − 47	8.20E − 161
f_2	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 − 30	5.63E − 22	—	—	4.14E − 90
f_3	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 − 1	6.66
f_4	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	—
f_5	Mean	0	0	1.48E − 9	1.43E − 5	1.59E − 6	0	0	1.02	2.64	2.53E − 14
	SD	0	0	5.70E − 9	6.78E − 5	1.15E − 6	0	0	1.01	1.61	1.38E − 14
f_6	Mean	0	0	0	5.80E − 1	0	1.12E − 3	—	6.15E − 2	3.70	1.25E − 11
	SD	0	0	0	6.78E − 1	0	4.27E − 3	—	1.82E − 1	2.14	6.75E − 11
f_7	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	0	0	0	8.86E − 4	3.38E − 8	0	0	3.34E − 2	3.94E − 14	2.94E − 14
f_8	Mean	0	0	0	4.60E − 9	1.58E − 9	0	0	1.90E − 1	1.97E − 3	1.22E − 2
	SD	0	0	0	8.73E − 9	2.60E − 9	0	0	1.90E − 1	1.97E − 3	1.58E − 2
f_9	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
f_{10}	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
f_{11}	Mean	0	0	2.49E − 1	2.30E − 8	2.26E − 9	0	—	0	4.85E − 3	
	SD	0	0	4.68E − 1	5.02E − 8	4.30E − 9	0	—	0	4.12E − 3	—
f_{12}	Mean	−4.48E2	−4.48E2	−4.49E2	6.07E2	1.22E3	1.19E3	—	3.33E2	1.56E2	7.34E − 10
	SD	3.04	1.95	1.26	2.02E2	3.63E2	3.70E2	—	1.73E2	7.97E1	2.73E − 12
f_{13}	Mean	3.25	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	—
f_{14}	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	—	—	—
f_{15}	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	—
f_{16}	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	—

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	—
Function	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
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		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
ACLPSO	Execution time (in ms)	Mean	784.20	847.88	485.52	765.48	617.16	545.16	823.52	888.64
		SD	32.11	37.07	63.98	50.57	26.41	39.47	41.14	39.62
ECLPSO	Execution time (in ms)	Mean	732.20	794.84	474.76	550.76	680.96	563.28	795.84	827.12
		SD	24.49	28.65	22.78	26.11	31.31	26.37	30.06	40.16
CLPSO	Execution time (in ms)	Mean	322.68	363.08	370.40	411.36	448.40	428.16	443.64	497.96
		SD	16.75	22.64	22.28	20.09	21.93	21.40	22.16	15.45
	Function		f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
ACLPSO	Execution time (in ms)	Mean	672.04	531.96	887.88	745.08	821.24	1138.20	1102.04	1095.20
		SD	46.41	86.05	40.16	39.98	42.10	79.25	37.37	46.11
ECLPSO	Execution time (in ms)	Mean	853.60	728.88	848.08	576.72	804.80	945.84	1010.16	1002.08
		SD	58.39	36.64	90.89	30.85	38.12	33.49	60.72	39.65
CLPSO	Execution time (in ms)	Mean	487.24	380.84	519.04	435.84	660.28	794.28	738.80	663.08
		SD	15.41	26.48	23.56	33.77	40.25	34.91	20.50	40.10

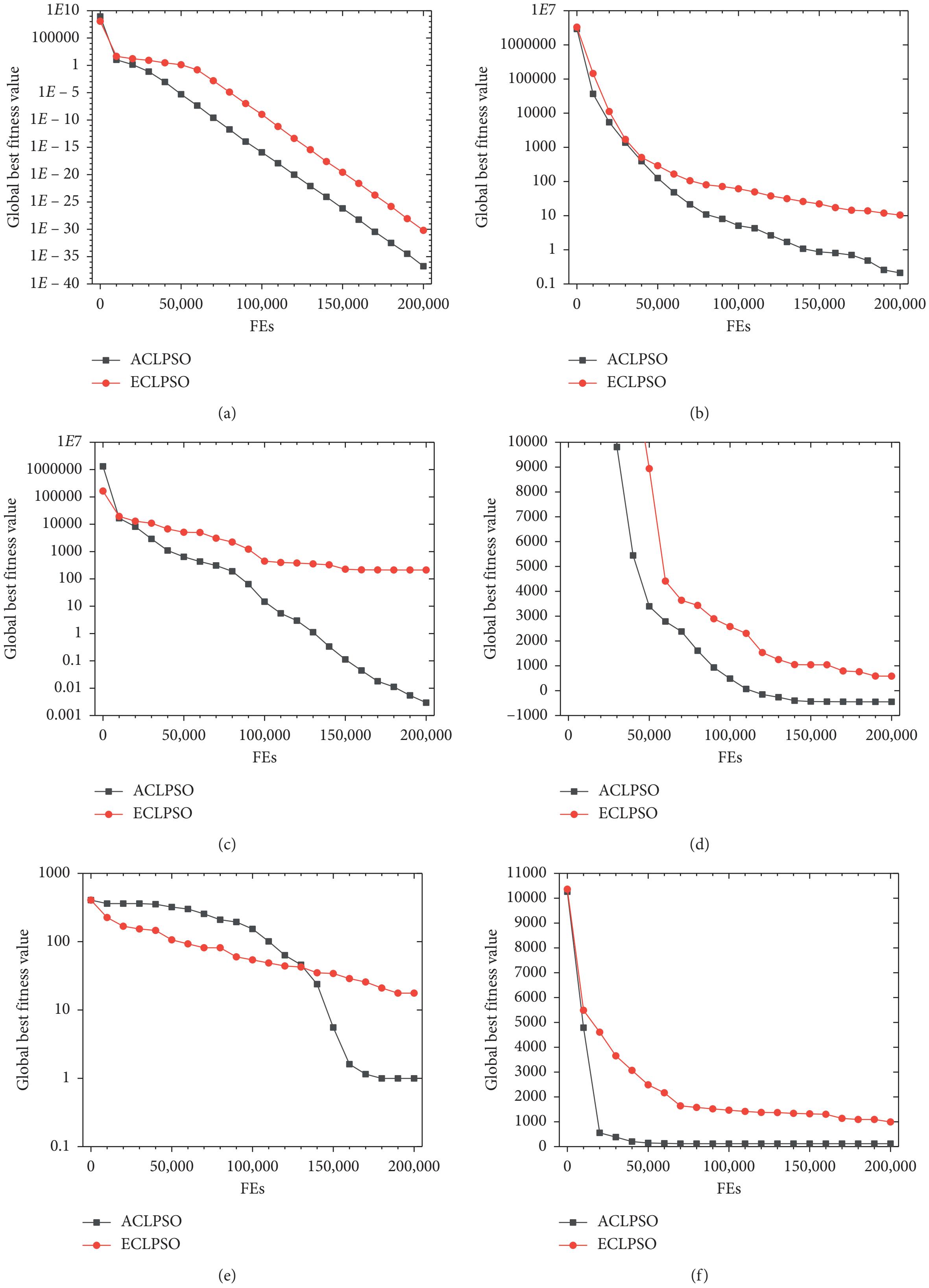
Table 7: Mean and standard deviation global best fitness value results of ACLPSO with other parameter settings on f_3, f_4, f_{12}, and f_{14}.

Function	Parameter setting	Global best fitness value	
		Mean	SD
f_3	$N = 40$, $s = 0.1$, $v = 0.2$	$1.89E1$	$2.66E1$
	$N = 40$, $s = 0.1$, $v = 0.1$	$1.44E1$	$2.24E1$
f_4	$N = 20$, $s = 0.1$, $v = 0.3$	$6.32E2$	$3.62E2$
	$N = 40$, $s = 0.3$, $v = 0.3$	$1.72E2$	$1.34E2$
	$N = 40$, $s = 0.9$, $v = 0.3$	$2.69E4$	$5.48E3$
f_{12}	$N = 40$, $s = 0.1$, $v = 0.1$	$2.15E2$	$1.90E2$
	$N = 40$, $s = 0.1$, $v = 0.2$	$-4.18E2$	$1.18E2$
f_{14}	$N = 20$, $s = 1.1$, $v = 0.3$	$7.56E2$	$5.21E2$
	$N = 40$, $s = 0.9$, $v = 0.3$	$4.25E2$	$4.34E2$
	$N = 40$, $s = 0.3$, $v = 0.3$	$5.61E2$	$6.43E2$

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 (in ms)	
		Mean	SD	Mean	SD
f_1	$N = 40$; the number of FEs is 500,000	$7.13E-29$	$2.41E-28$	815.32	27.00
	$N = 80$; the number of FEs is 500,000	$3.13E-19$	$5.15E-19$	805.48	28.88
f_2	$N = 40$; the number of FEs is 500,000	$9.39E-22$	$1.72E-21$	844.08	13.77
	$N = 80$; the number of FEs is 500,000	$8.38E-13$	$2.69E-13$	861.88	22.36

on all the functions except f_9 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_3, f_{10}, f_{11}, 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 dimensional 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.



(a)

(b)

(c)

(d)

(e)

(f)

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 ALC-GPSO 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_3, f_4, f_{12}, f_{13}, and f_{14} 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 ALC-GPSO 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 $f_3, f_4, f_5, f_{12}, f_{13}, f_{15}$, and f_{16}, for FLGPSO-QIW on $f_3, f_4, f_5, f_6, f_{12}, f_{13}, f_{15}$, and f_{16}, and for ALC-GPSO on f_3, f_9, and f_{10}. The accuracies of ACLPSO's mean global best fitness value results are the best on $f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}, f_{14}$, 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 $f_2, f_3, f_4, f_{10}, f_{12}, f_{13}$, and f_{14}; therefore, the global best fitness value results of ACLPSO 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 ACLPSO successfully addresses significantly bettering the exploration performance of ECLPSO, and ACLPSO is still good at exploitation and convergence, owing to the adoption of the dimensional 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 performance; as a result, the mean execution time results of ACLPSO and ECLPSO are more than those of CLPSO on all the functions in Table 6. The differences are most noticeable on $f_1, f_2, f_7, f_8, f_{11}, f_{15}$, and f_{16}, and the mean execution time of ACLPSO 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 ACLPSO and ECLPSO than CLPSO would become relatively very small. The mean execution time results of ACLPSO are slightly more than those of ECLPSO on $f_1, f_2, f_3, f_7, f_8, f_{11}$, 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_9 and f_{10}, meaning that the dimensional independent and adaptive maximum velocities, inertia weights, acceleration coefficients, and leaning probabilities essentially do not increase execution time. With respect to ACLPSO, 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_9, the pair of f_{15} and f_8, and the pair of f_{16} and f_7. The SD execution time results of ACLPSO, 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 ACLPSO. As can be seen from Table 7, setting N as 20 on f_4 and f_{15} renders much worse mean and SD global best fitness value results, and ACLPSO 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 v to be more than 0.05 on f_3, the normative interval scaling coefficient s to be more than 0.1 on f_4, v to be less than 0.3 on f_{12}, and s to be less than 1.1 on f_{14} causes ACLPSO 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, v) 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 ACLPSO is quite better than that of ECLPSO on f_2. ACLPSO 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, ACLPSO takes the same best s values on f_3, f_4, and f_{12}, the same best v 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, v) combinations of ACLPSO 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 v based on the generation counter and the dimensional normative intervals.

6. Conclusions

In this paper, we have proposed ACLPSO for the purpose of further significantly mending the exploration performance of ECLPSO. ACLPSO 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. ACLPSO determines the normative interval with respect to each dimension in each generation. Based on the dimensional normative intervals, ACLPSO 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 ACLPSO successfully addresses exploration as well as exploitation and convergence as ACLPSO 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. ACLPSO 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 Province Natural Science Foundation Exceptional Young Scholar Project (2018ACB21029), the Shaanxi Province 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.-a. 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, “Multiswarm particle swarm optimization with transfer of the best particle,” *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 904713, , 2015.

[23] R. Jensi 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 swarm 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. Chaovalitwongse, "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.