Comprehensive Learning Particle Swarm Optimization Algorithm With Local Search for Multimodal Functions

Yulian Cao, Han Zhang, Wenfeng Li, *Senior Member, IEEE*, Mengchu Zhou, *Fellow, IEEE*, Yu Zhang, and Wanpracha Art Chaovalitwongse, *Senior Member, IEEE*

***Abstract*—A comprehensive learning particle swarm optimizer (CLPSO) embedded with local search (LS) is proposed to pursue higher optimization performance by taking the advantages of CLPSO's strong global search capability and LS's fast convergence ability. This paper proposes an adaptive LS starting strategy by utilizing our proposed quasi-entropy index to address its key issue, i.e., when to start LS. The changes of the index as the optimization proceeds are analyzed in theory and via numerical tests. The proposed algorithm is tested on multimodal benchmark functions. Parameter sensitivity analysis is performed to demonstrate its robustness. The comparison results reveal overall higher convergence rate and accuracy than those of CLPSO, state-of-the-art particle swarm optimization variants.**

***Index Terms*—Adaptive strategy, evolutionary algorithm, local search (LS), multimodal function, particle swarm optimization (PSO).**

I. Introduction

IN REAL world, most of practical engineering problems are multimodal functions [1]–[3] whose optimization is still one of the most challenging tasks due to many local optima. Several evolutionary algorithms, such as particle swarm optimization (PSO) [4], ant colony optimization [5], differential evolution (DE) algorithm [6], cooperative co-evolution algorithm [7], and estimation of distribution algorithm [8], have been proposed to solve the multimodal optimization problems. Among these algorithms, PSO which imitates the foraging behavior of bird flocks [4] is one of the most outstanding population-based evolutionary algorithms. It becomes popular not only because it is easy to implement, but also due to its strong optimization ability. Its efficiency in solving benchmark functions and complex optimization problems attracts numerous attentions [9]–[17].

For the optimization of multimodal functions where many local optima exist, the optimization algorithm should try to avoid trapping into local optima, which is a difficult issue for many heuristic algorithms [18]. As a typical metaheuristic algorithm, the standard PSO algorithm indeed has the drawback of premature convergence and easily trapping into local optima. Many improved PSO variants focus on solving the problem of trapping into local minima when used for minimizing multimodal functions, which aim to improve the global search ability of PSO. Much progress has been made. Some studies aim to avoid premature convergence by maintaining or increasing population diversity. There are many such means, e.g., dynamic clustering [19], [20], supervised learning [21], historical memory [22], and adopting different learning strategies [23], [24]. Researches have been done to improve the global search ability by combining PSO with other intelligent algorithms [25].

Regarding the optimization of multimodal functions, the convergence rate within a local unimodal area is important to ensure the optimization efficiency. In recent years, it has aroused people's attention. However, for most of the multimodal optimization algorithms, the local search capabilities require much improvement. If the required accuracy is high, they fail to find the desired optima even after converging near them. Many existing studies tried to balance the global and local search abilities, but it is difficult to ensure that both are high [26]. Most researches focus on the improvement of the former. However, less progress has been made to improve the latter. The traditional LS methods, such as quasi-Newton methods have fast local convergence rate [27], and they can be used to address a slow local convergence issue.

This paper integrates an LS method with an existing PSO algorithm with strong global search ability, named comprehensive learning PSO (CLPSO). As an excellent PSO variant, CLPSO adopts a comprehensive learning strategy and offers

Manuscript received November 30, 2017; revised May 18, 2018 and September 24, 2018; accepted November 8, 2018. Date of publication December 5, 2018; date of current version July 30, 2019. This work was supported by the National Natural Science Foundation of China under Grant 61571336, Grant 71372202, and Grant 71672137. *(Yulian Cao and Han Zhang contributed equally to this work.) (Corresponding author: Wenfeng Li.)*

Y. Cao, W. Li, and Y. Zhang are with the School of Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China (e-mail: yalianjingren@126.com; liwf@whut.edu.cn; sanli@whut.edu.cn).

H. Zhang is with the Institute of Nuclear and Energy Technology, Karlsruhe Institute of Technology, 76344 Eggenstein-Leopoldshafen, Germany (e-mail: han.zhang@kit.edu).

M. Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

W. A. Chaovalitwongse is with the Department of Industrial Engineering, University of Arkansas, Fayetteville, AR 72701 USA (e-mail: artchao@uark.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. This material consists of formulas of benchmark functions and their description, and tables. The total size of this PDF file is 0.078 MB.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2018.2885075

very strong global search ability [28]. All particles' pervious best (referred to as pbest) information, not just its own best, is used to guide the search of a particle. Another difference between it and standard PSO is that a particle learns from different particles' pbest for different dimensions. Further research on it has been carried and its superiority over standard PSO and several other algorithms is demonstrated in [29] and [30]. Traditional LS operators are utilized to improve its local search efficiency in this paper.

In terms of the combination of LS and metaheuristic algorithms, several studies have been performed with the help of a genetic algorithm [31] and DE [32]. The results show that LS plays a positive role in accelerating their local convergence. This combination of population-based global search and individual-based local search, called a memetic algorithm [33], can be regarded as an example mechanism. The memetic algorithm offers a pattern to enhance both global and local search abilities. In recent years, its theory and application research are increasing. It can be recognized that combining swarm intelligent algorithms and LS has become an important research direction.

Some attempts have been carried out to apply LS in PSO to accelerate the local convergence of PSO variants [34], [35]. When to switch from global search to LS is a key issue. However, it is empirical in existing researches. They lack any theoretical analysis and support. For example, LS is conducted at each arbitrary given number of PSO iterations [34] or performed for all particles after each PSO iteration [35]. In most existing approaches, the switch point is predefined and fixed for all functions, which cannot take full advantage of the PSO dynamic information in the search process and cannot capture the difference among different functions. Therefore, a more effective adaptive LS starting strategy depending on the PSO dynamic behavior is needed for combining PSO and LS. Nevertheless, research on such adaptive strategy is rarely reported.

In this paper, CLPSO with an adaptive LS starting strategy named CLPSO-LS is proposed to improve the performance of CLPSO. The particle diversity is employed to describe the dynamic behavior of PSO using a quasi-entropy index, which can be used to identify whether global optimum basin (GOB) has been found. It helps to effectively tackle the key issue about when to initiate LS.

The rest of this paper is organized as follows. Related fundamental knowledge is given in Section II. Section III describes the proposed algorithm and adaptive LS starting strategy. In Section IV, the performance of the adaptive LS starting strategy is tested and analyzed. In Section V, numerical experiments are performed to test the proposed algorithm. Finally, the conclusions are drawn in Section VI.

II. Fundamental Knowledge

A. Particle Swarm Optimization Algorithm

PSO is a swarm intelligence algorithm, which is inspired by the foraging behavior of birds. A swarm with defined population size is stochastically initialized in the search space. Each particle is a candidate solution to the problem and is represented by a velocity v and a location x in the search space [4]. Particles have memory and share information with each other. Their velocities and locations are iteratively updated as follows:

$$v_{\mathrm{id}}^{t+1} = v_{\mathrm{id}}^{t} + c_1 \cdot r_1 \cdot \left(\mathrm{pbest}_{\mathrm{id}}^{t} - x_{\mathrm{id}}^{t}\right) + c_2 \cdot r_2 \cdot \left(\mathrm{gbest}_{\mathrm{d}}^{t} - x_{\mathrm{id}}^{t}\right) \tag{1}$$

$$x_{\mathrm{id}}^{t+1} = x_{\mathrm{id}}^{t} + v_{\mathrm{id}}^{t+1} \tag{2}$$

where i means the ith particle and t is the current number of iterations. d represents the problem dimension. r_1 and r_2 are random numbers that are uniformly distributed in the range of [0,1]. c_1 and c_2 are learning factors. gbest denotes the best position in the whole swarm.

The standard PSO algorithm converges fast but is easy to trap into a local optimum. This is because of premature convergence caused by the loss of the population diversity. PSO with inertia weight (ωPSO) is proposed to balance the global and LS abilities, where a parameter named inertia weigh is added to the first part of PSO. Its velocity is updated using the following equations [4]:

$$\begin{aligned} v_{\mathrm{id}}^{t+1} = {} & \omega_i \cdot v_{\mathrm{id}}^{t} + c_1 \cdot r_1 \cdot \left(\mathrm{pbest}_{\mathrm{id}}^{t} - x_{\mathrm{id}}^{t}\right) \\ & + c_2 \cdot r_2 \cdot \left(\mathrm{gbest}_{\mathrm{d}}^{t} - x_{\mathrm{id}}^{t}\right) \end{aligned} \tag{3}$$

$$\omega_i = \omega_{\mathrm{ini}} - (\omega_{\mathrm{ini}} - \omega_{\mathrm{end}}) \cdot t / M_{\mathrm{Iter}}. \tag{4}$$

A linear decreased inertia weight (denoted by ω_i) is widely used to better balance the local and search ability, as shown in (4). ω_{ini} is the inertia weight of the first loop, ω_{end} is the inertia weight of the last loop, and M_{Iter} represents the total number of iterations. It has stronger global ability with larger value of ω_i at the early stage by exploring more space. As the algorithm goes to the end of the iteration, it focuses on local exploitation around gbest and pbest.

CLPSO [28] is proposed to deal with the premature convergence and has led to a very successful improvement over PSO. It has been widely used in many benchmark problems [28], [36] and engineering applications [37], [38]. Moreover, CLPSO is a good cornerstone for further improvement and pursuit of better performance, such as DNLPSO [29] and HCLPSO [30]. Hence, CLPSO is utilized in this paper. Its velocity update formula is as follows [28]:

$$v_{\mathrm{id}}^{t+1} = \omega_i \cdot v_{\mathrm{id}}^{t} + c \cdot r_{\mathrm{id}} \cdot \left(\mathrm{pbest}_{\mathrm{fi(d)d}}^{t} - x_{\mathrm{id}}^{t}\right) \tag{5}$$

where fi(d) defines which particle's pbest that particle i should follow. It should be noted that different dimensions of a particle can select different learning objects. Each individual learns from a different particle's pbest other than its own best only. This learning strategy effectively maintains the population diversity, such that the global search ability of CLPSO is excellent. However, its local search is not fast enough at the later iteration when it focuses on exploitation. When a particle "flies" outside the boundary of the search space, it is randomly relocated in the search space.

B. Local Search (LS)

The Nelder–Mead method and quasi-Newton method are traditional LS methods [27]. Broyden–Fletch–Goldfarb–Shanno (BFGS) is a widely used quasi-Newton method. The Nelder–Mead method is a commonly used nonlinear optimization technique, which is a well-defined numerical method for optimizing an objective function in a multidimensional space. It is applied to nonlinear derivative-free optimization problems.

The quasi-Newton method is the most efficient algorithm to perform unconstrained optimization. Its basic idea is to approximate the inverse matrix of the Hessian matrix in the Newton method with a matrix that does not contain second-order derivative, which overcomes the shortcoming of the Newton method that requires the calculation of second-order partial derivative and the Hessian matrix of the objective function may be nonpositive. BFGS employs the following method to construct approximate matrices [27]:

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k s_k s_k^T H_k^T}{s_k^T H_k s_k} \tag{6}$$

where, $s_k = x_{k+1} - x_k$, $q_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.

What needs to be explained is that BFGS employs a finite difference method to approximate the first-order derivative of their objective function, which allows it to be generalized to nondifferentiable functions. Yet its convergence rate may be reduced when dealing with discontinuous and nondifferentiable functions. The convergence speed of the Nelder–Mead method can be faster than it for such functions since it does not require derivative information of its objective function.

C. Convergence Rate Analysis

CLPSO has a strong global exploration capability [28]. Now we analyze its local convergence rate at its later stage. According to the analysis of PSO convergence rate as indicated in [39], it is generally simplified as a 1-D (abbreviate as D) problem with a single particle. The update formula of CLPSO can be simplified and written in the form of a matrix as

$$\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} v^{t+1} \\ x^{t+1} \end{bmatrix} = \begin{bmatrix} \omega & -c \cdot r \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v^{t} \\ x^{t} \end{bmatrix} + \begin{bmatrix} c \cdot r \cdot \text{pbest} \\ 0 \end{bmatrix}. \tag{7}$$

For the local exploitation at its later stage, when CLPSO converges, there is a fixed point (v^*, x^*) that satisfies

$$\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} v^{*} \\ x^{*} \end{bmatrix} = \begin{bmatrix} \omega & -c \cdot r \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v^{*} \\ x^{*} \end{bmatrix} + \begin{bmatrix} c \cdot r \cdot \text{pbest} \\ 0 \end{bmatrix}. \tag{8}$$

Define $X = [v, x]^T$ and $\Delta X^t = X^t - X^*$. According to (7) and (8), we have

$$\Delta X^{t+1} = K \cdot \Delta X^t \tag{9}$$

where $K = \begin{bmatrix} w & -c \cdot r \\ w & 1 - c \cdot r \end{bmatrix}$.

Then, according to the definition of matrix compatibility norm, we can derive

$$\frac{\left\| \Delta X^{t+1} \right\|_1}{\left\| \Delta X^{t} \right\|_1} \le \|K\|_1 \tag{10}$$

where $\|K\|_1 = \max(2 \cdot w, 1 - 2 \cdot c \cdot r)$. Without loss of generality, 1-norm is used here. It shows that the norm ratio of particle location deviation $\|\Delta X\|$ between two iterations is a nonzero constant, which indicates that CLPSO has only linear convergence rate, which agrees with analysis results of other PSO algorithms [39].

For comparison, we analyze the local convergence rate of LS methods theoretically. BFGS is used as an example. According to [40], its convergence rate can be expressed as

$$\lim_{t\to\infty} \frac{\left\| x^{t+1} - x^* \right\|}{\left\| x^{t} - x^* \right\|} \le \lim_{t\to\infty} \left(\gamma (1 + r) \left\| H^t - F'\left(x^*\right) \right\| \right) = 0 \tag{11}$$

where $\gamma \ge \|F'(x^*)^{-1}\|$, which represents the upper bound of the norm of matrix $F'(x^*)^{-1}$.

It shows that the norm ratio of particle location errors between two iterations is zero, which indicates that BFGS has a super-linear convergence rate.

To sum up, LS methods represented by BFGS have a higher local convergence performance than CLPSO. Their use is expected to accelerate the local phase of CLPSO. Therefore, we intend to combine LS methods with CLPSO to find the desired optima with fewer iterative steps in the exploitation phase.

III. Proposed Algorithm

A. CLPSO-LS

As described in Section II, CLPSO has strong global search ability due to its specially designed learning strategy, but ineffective local convergence rate due to its stochastic nature, while LS has faster local convergence speed than CLPSO. Therefore, the main idea of our proposed algorithm is to use CLPSO for global search, and the prior mentioned LS methods for local search when there are particles in GOB. The algorithm is named as CLPSO with adaptive LS starting strategy (called CLPSO-LS) with the following steps.

Step 1: Generate an initial population, specifically, initialize the position and velocity of each particle.

Step 2: Carry out the CLPSO iteration loop.

Step 2.1: Evaluate all individuals in the population.

Step 2.2: Update velocity and position for individual via (5) and (2).

Step 2.3: Update the previous best solution of individual (pbest) and global one (gbest) visited so far. Update fitness of pbest (denoted as U) and fitness of gbest (denoted as G).

Step 2.4: Judge our proposed adaptive LS starting criterion, i.e., whether GOB has been searched. If so, go to step 3. Otherwise, go to step 2.

Step 3: Carry out the LS, where BFGS quasi-Newton is taken as an example with the gbest as an input seed.

Step 4: Judge whether better solutions obtained by LS. If so, go to step 5. Otherwise, go to step 6.

Step 5: Update gbest and U.

Step 6: Output gbest and G.

CLPSO iteration is performed before the LS. The basic idea of CLPSO-LS is to improve the local convergence speed of CLPSO by using proper LS methods.

How to determine whether particles have entered GOB is thus a key issue. In other words, when to start LS is a vital problem to be answered. In the rest of this section, the necessity of starting LS adaptively is discussed first. Next, a key index named quasi-entropy for starting LS adaptively is introduced. Then, the behavior feature of quasi-entropy is

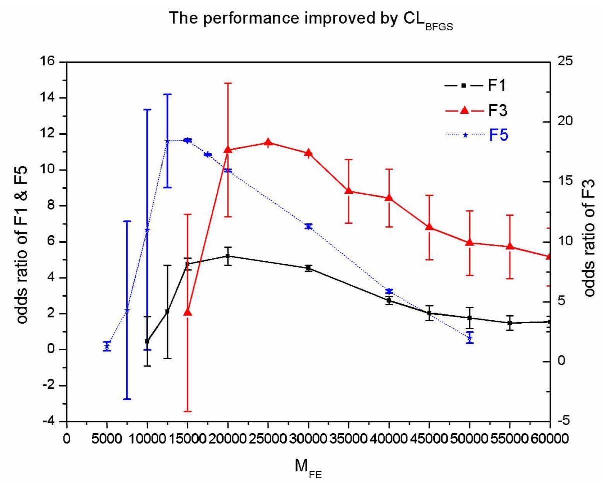

Fig. 1. Performance improved by CL_{BFGS} for F1, F3, and F5 at different M_{FE}.

analyzed theoretically. Finally, the adaptive starting criterion of LS is proposed.

B. Necessity of Starting LS Adaptively

On one hand, if LS were carried out before sufficient global search, the proposed algorithm would converge to a local optimum. This is not consistent with the final objective to find the global optimal solution. Clearly, prematurely starting LS cannot take any advantages of the proposed algorithm. On the contrary, it may fall into local minima instead, resulting in poor results. On the other hand, if LS starts when CLPSO has already done much random LS, it may waste too much computing resources. The advantage of CLPSO-LS cannot be revealed either.

The performance of CL_{BFGS}, named for CLPSO with BFGS quasi-Newton method as its LS, is analyzed with starting LS at different CLPSO iterations by testing 10-D Ackley's function (F1), Rastrigin's function (F3), and Schwefel's function (F5), respectively. The maximum number of function evaluations (referred to as M_{FE}) and a fixed number of function evaluations for LS (referred to as L_{FE}) are given. Define S_t as $M_{FE} - L_{FE}$. When CLPSO has performed S_t function evaluations, LS is launched. Each experiment runs for 25 times independently. For comparison, experiments for CLPSO under the same M_{FE} are carried out. The error values (denoted as e) of CL_{BFGS} and CLPSO are recorded. They are denoted as $e(CL_{BFGS})$ and e(CLPSO), respectively. e is calculated by

$$e = f(x) - f^* \quad (12)$$

where f^* repents the global optimum's fitness value. The odds ratio (denoted as τ) is calculated as follows:

$$\tau = \log_{10}\left(\frac{e(\text{CLPSO})}{e(\text{CL}_{\text{BFGS}})}\right). \quad (13)$$

The logarithmic operation helps to intuitively learn the difference. Fig. 1 shows the degree that the performance is enhanced by CL_{BFGS}. Take the scatter of F1 when M_{FE} equals 20 000 as an example. It means that the odds ratio of the error value from CL_{BFGS} to that from CLPSO is about 5. As can be seen from Fig. 1, the improvement curves show significant peaks, where the multiple ascends first and then descends with the increase of the M_{FE} before starting LS. Moreover, the error bars at the peak window are smaller than that in other regions, indicating the performance during the peak window is more robust than those at other time. Therefore, the peak window, where performance improved most is the best time to start LS, and it can maximize the advantages of the proposed algorithm.

It can be seen from the graph, the position of the maximum odds ratio is different for F1, F3, and F5, and the width of the window is also not the same. Here, high performance odds ratio interval of F5 could be achieved when LS is used at about 15 000 M_{FE}, while the best start point of LS for F3 is about 25 000 M_{FE}. These results indicate that the best time to start LS is different for different functions. Furthermore, even for the same function, because of the randomness nature in PSO, such as the random initialization, random learn factors, and random learn objects, the best time to start LS is different at each independent run. In short, it cannot set a predefined fixed reasonable value for every experiment. This requires an adaptive LS starting strategy based on the dynamic behavior of PSO, such that different functions can start LS at an appropriate time.

C. Quasi-Entropy

Which particles are selected to do LS should be considered in integrating LS into CLPSO. First, even if all particles are selected to do LS when the particles are not in GOB, only local optimum are obtained rather than the global one. It is clearly inadvisable. Second, if many or even all particles do LS, since these particles may be in the same single peak region, they converge to the same value, resulting in a waste of computing resources. This is clearly inadvisable too. Therefore, our proposed strategy is to perform LS with gbest as a seed to maximize efficiency.

How to determine the appropriate time to start LS is a key issue. We cannot judge whether a particle has reached GOB by using the information of a single particle itself. However, based on the pbest values, if more than one top-ranking particle gather together instead of scattered distribution, it is believed that these particles have entered GOB and they are approaching to the global optimal solution. It heralds the end of CLPSO's global search phase. This is the best time to start an LS method. How to determine the degree of aggregation of these particles becomes the key. Studies [41], [42] have shown that population diversity can determine the aggregation degree.

In PSO, commonly used physical quantities are particle position, pbest and particle velocity [38]. They can construct different diversity indices. Take the particle position for example, commonly used indicators are variance, Euclidean distance, dimensional variance, and Shannon entropy. Shannon entropy is a frequently used population diversity index in evolutionary algorithms. Take the particles' fitness for example, the entropy of the tth generation [42] is as follows:

$$E = -\sum_{i=1}^{m} p_i \log(p_i) \quad (14)$$

where m is the total number of types of fitness, and p_i represents the ratio of the number of particles contained in each type to the total number. When the fitness of all particles is the same, the entropy equals to zero. The larger difference that particle's fitness has, the larger value entropy is, which means more even distribution of particles. After the foregoing analysis and research, better particles' fitnesses of pbest are selected as a diversity index. The reasons are as follows.

1) The reason for using fitness rather than position as diversity is that even if particles' positions are close to each other, they might be in different peaks for multimodal problems. Their fitness value can vary widely and so the fitness is more appropriate to be adopted as a diversity criterion.
2) From the velocity update formula, current information of a particle is random, or the index is affected by noise. However, pbest records individual previous best location and tends to be better. Thus, this information is of strong reliability. In this case, pbest is more effective for portraying the population diversity than the current positions of particles.
3) Since only those good-performance particles but not all particles are needed to care about, only a part of U's are used to calculate the population diversity to exclude the interference of poor-performance particles to the metrics. The percentage of particles to be selected for calculating the entropy is denoted as β.

According to its original definition, population entropy's evaluation needs the maximum fitness of pbest (referred to Umax) and the minimum value (referred to Umin). Then [Umin, Umax] is divided into m equal intervals. m is usually taken as the population size N. One common way is setting the maximum fitness of pbest in initial population as Umax and Umin to be zero. The interval size is directly related to the number of particles, but the number of particles is very small in some real problems. Therefore, the interval size is selected to be very large. In later iterations, all particles are treated in the same interval once particles start to gather. Then the entropy value is calculated as zero. However, if particles have not yet entered GOB, the original definition of the population entropy no longer has the ability to portray the change of diversity. Therefore, there are some hidden defects in the original definition of population entropy.

In this paper, a quasi-entropy (denoted as Q_E) index is proposed to overcome the above limitations in the entropy index, which is used to characterize diversity

$$Q_E^t = -\sum_{i=1}^{N} P_i^t \log P_i^t \tag{15}$$

where

$$P_i^t = \frac{U_i^t}{\sum_{i=1}^{N} U_i^t}. \tag{16}$$

Its implication is that if the particles selected to calculate the Q_E are regarded as a set $\boldsymbol{S}$, P_i^t represents the proportion of each particle's U to the sum of all particles' U in set $\boldsymbol{S}$. The advantage of using Q_E is avoiding any parameters settings.

D. Theoretical Analysis of Quasi-Entropy

It is important to analyze the mathematical properties of quasi-entropy theoretically. Some assumptions are as follows.

1) The total number of particles used to compute it is N. The ratio of particles that have entered GOB is λ. Without loss of generality, assume that λN is an integer.
2) The following relationship is satisfied for the λN particles that have entered GOB. $f(\text{pbest}_1) = f(\text{pbest}_2) = \cdots = f(\text{pbest}_{\lambda N}) = O(\varepsilon)$, where ε is a small number that approaches 0. $f(\text{pbest}_i)$ represents the fitness of the ith particle's pbest.
3) For the other $(1-\lambda)N$ particles that do not enter GOB, the following relationship is satisfied:

$$f\big(\text{pbest}_{\lambda N+1}\big) = f\big(\text{pbest}_{\lambda N+2}\big) = \cdots = f\big(\text{pbest}_{N}\big) = O(1).$$

Then, it is easy to get the relationship

$$O(1) \gg O(\varepsilon).$$

Based on the above assumptions, Q_E can be expressed as the sum of two parts, denoted as Q_{E1} and Q_{E2}, respectively

$$Q_E = -\sum_{i=1}^{N} P_i \log P_i = \underbrace{\left(-\sum_{i=1}^{\lambda N} P_i \log P_i\right)}_{Q_{E1}} + \underbrace{\left(-\sum_{i=\lambda N+1}^{N} P_i \log P_i\right)}_{Q_{E2}}.$$

For the part of particles that have entered GOB

$$P_i = \frac{U_i}{\sum_{i=1}^{\lambda N} U_i} = \frac{O(\varepsilon)}{\sum_{i=1}^{\lambda N} U_i} \to 0.$$

Hence, $Q_{E1} = -\sum_{i=1}^{\lambda N} (\lim_{P_i \to 0} P_i \log P_i) = 0$.

For the portion of particles that have not yet entered GOB

$$P_i = \frac{O(1)}{\sum_{i=\lambda N+1}^{N} O(1)} = \frac{1}{(1-\lambda)N}.$$

Hence, $Q_{E2} = -\sum_{i=\lambda N+1}^{N} P_i \log P_i = -\log(1/[(1-\lambda)N])$. Thus,

$$Q_E = Q_{E1} + Q_{E2} = -\log\left(\frac{1}{(1-\lambda)N}\right). \tag{17}$$

We obtain the partial derivative of quasi-entropy with respect to λ

$$\frac{\partial(Q_E)}{\partial\lambda} = -\frac{1}{(1-\lambda)} < 0 \tag{18}$$

$$\frac{\partial^2(Q_E)}{\partial\lambda^2} = -\frac{1}{(1-\lambda)^2} < 0. \tag{19}$$

We analyze its behavior theoretically based on (17). When no particle entering GOB, λ equals 0. Thus, Q_E equals a constant $-\log(1/N)$, which indicates that Q_E fluctuates slightly over a small range around this constant. When $0 < \lambda < 1$, the partial derivative of Q_E with respect to λ is always negative, as shown in (18) and when $\lambda = 0$, $[\partial(Q_E)/\partial\lambda]_{\lambda=0} = -1$. It indicates that Q_E begins to decrease when there are particles entering GOB. Furthermore, according to (19), $\partial^2(Q_E)/\partial\lambda^2$ is always negative, thus indicating that the more particles entering GOB, the faster Q_E declines.

It should be noted that the above conclusion is also valid for the situation that CLPSO could not find the global optimum due to the extreme complexity of some optimization problems. The theoretical tool can be extended to analyze the quasi-entropy performance during the global search phase of finding the "best basin." Here, the term best basin means the best region that PSO can find rather than the basin of the global optimum. In this situation, the assumptions are modified as follows.

$O(\varepsilon)$ is the function fitness in the "best" local optimum basin, which is much smaller than that of particles do not enter the best local optimum basin $O(M)$

$$O(M) \gg O(\varepsilon). \tag{20}$$

It is well known that the function $y = P_i \log P_i$ is a monotonically increasing function with respect to P_i. Therefore, we obtain the relationship $Q_{E2} \gg Q_{E1}$. Thus, the same conclusions in (17)–(19) are also correct for the "best" local optimum basin case.

E. Adaptive LS Starting Strategy

According to the above analysis, it can be concluded that in the early stage of CLPSO, i.e., its global search stage, no particle enters GOB and Q_E basically keeps a constant. However, in its later stage, as more and more particles enter GOB, Q_E experiences a sharp declining process, which indicates that CLPSO has started its local convergence phase. This provides a possibility of judging the switch from a global search phase to a local search phase. The ideal state is that LS starts after gbest enters GOB and hopefully the start can be as soon as possible to accelerate local search.

We propose the following strategy to start LS adaptively in our research. When Q_E of iteration t decreases to a certain percentage of that at initial population, LS is triggered. The judgment criterion is as follows:

$$Q_E^t \leq \theta \cdot Q_E^0 \tag{21}$$

where θ is the reduction ratio of Q_E. Thus, the adaptive starting strategy of LS and CLPSO is when Q_E of iteration t satisfies (21). The performance of this strategy is to be tested and analyzed to verify its rationality.

IV. Performance Test of Adaptive LS Starting Strategy

A. Performance of Quasi-Entropy in Two-Dimensional Problems

The performance of quasi-entropy is analyzed by taking the two-D Schwefel function (F5) as an example. To visualize the distribution of particles, CLPSO is applied to it, and the population size is set to be 15. Q_E at different iterations and its corresponding particle positions are shown in Fig. 2, demonstrating three stages that no particles enter GOB, some enter it and most of them enter it.

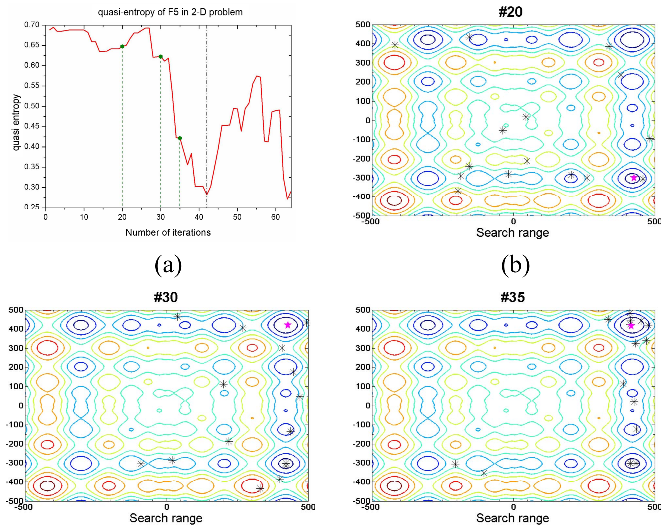

Fig. 2. Curve of quasi-entropy and particle distribution at some iterations. (a) Change of Q_E with iteration count. (b) Distribution of particles at the 20th iteration. (c) Distribution of particles at the 30th iteration. (d) Distribution of particles at the 35th iteration.

After 20 iterations, Q_E is still high. This can be reflected from Fig. 2(b), where scattered particles mean strong diversity and particles have not entered GOB. It can be seen from the curve in Fig. 2(a) that Q_E begins to sharply decline from the 30th iteration and decreases to a very small value at the 40th iteration. This is consistent with the theoretical analysis. Eight particles gather at the top right corner that is the same area of the global optimal solution, where majority of particles have started the local convergence stage. Since only partial particles are considered to calculate the quasi-entropy and the population size is 15, it means that all the 1/3 top ranking particles have gathered to the same single peak area. Therefore, if LS is triggered between the time window of the 30th and 40th iterations, the global optimum can be found very fast. The earlier LS starts in this window, the faster the proposed algorithm converges to the global optimum. If LS starts after the 40th iteration, CLPSO has made much random LS itself, which wastes computing resources. However, if an LS method is used right at the 30th iteration, this could be realized with much fewer iterations.

To sum up, it can be seen from the curve of Q_E that it is relatively stable first and begins to rapidly decline at some point, which is consistent with the theoretical analysis. Many other multimodal functions are tested. Their results are not shown due to limited space. They all show that this phenomenon widely exists. It can be concluded from the theoretical and experimental analysis, the proposed adaptive judgement criterion based on quasi-entropy can effectively judge whether particles have entered GOB. Thus, Q_E can be effectively used to find a proper time to start LS.

B. Adaptive LS Starting Strategy for High-Dimensional Problems

A quantitative index, ratio of particles in GOB (referred to as ρ_{GOB}), is proposed to further verify the rationality of the

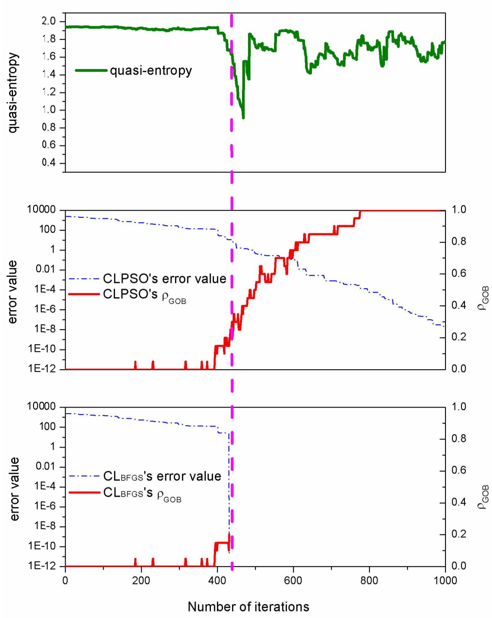

Fig. 3. Quasi-entropy, ρ_{GOB} and error value of the 10-D F5.

adaptive LS starting strategy for high dimensional multimodal functions. Taking the 10-D multimodal function Schwefel (F5) as an example, define its search range as $[-500, 500]^{10}$. Its location of the global optimum (denoted as x^*) is known, which is at $x^* = [420.96, 420.96, \ldots, 420.96]$. By derivative analysis, the continuous region in the vicinity of x^* is GOB. According to calculation, the interval of GOB of F5 is $[302.96, 558.96]^{10}$. Then ρ_{GOB} is the ratio of the number of particles entering GOB to the population size of the swarm.

For comparison, CLPSO and CLPSO-LS based on adaptive BFGS quasi-Newton (referred to as CL_{BFGS}) are tested on the 10-D F5. The population size is 20 and M_{FE} is 40 000. In order to clearly capture when to start LS, the data used for Fig. 3 is from a single run. The relationship of quasi-entropy, ρ_{GOB} and error value is show in Fig. 3. Some vital information can be concluded from it.

1) The same variation rule of quasi-entropy as that of the two-D problem is observed. Q_E is stable first and begins to decrease rapidly at a point, which is consistent with the conclusion of our theoretical analysis. The vertical dotted line indicates the point, where LS starts, which is determined by the adaptive judgment criterion of (20).
2) CLPSO is divided into the early global exploration stage and the later local exploitation stage, and entering GOB is the segmentation point of these two stages. First, when it is in the global search stage, ρ_{GOB} equals zero. Then ρ_{GOB} increases as particles enter GOB. When ρ_{GOB} increases to a certain value, error value begins to decline, and then drops rapidly. This shows that it has entered the local fast convergence stage.
3) If not using LS, even if ρ_{GOB} is already high, the fitness is still very large. It means that CLPSO has lower computational efficiency than CL_{BFGS}.
4) ρ_{GOB} has risen to a certain value when Q_E decreases rapidly. It shows that a certain number of particles have entered GOB. It is not appropriate to start the LS directly when only a very small number of particles enter GOB, because at such time the number of particles entering it is too small and the information in the basin is too limited to determine whether this basin is a true GOB. Thus, it can be seen that LS is activated only after a certain number of particles enter GOB.

V. Experimental Results and Performance Comparisons

A. Multimodal Benchmark Functions

Sixteen widely used multimodal benchmark functions are utilized to examine the performance of CLPSO-LS, which are selected from [28], [43], and [44]. See their formulas in the supplementary material. Detailed properties of each function, including the global optimum (x^*), search range, the global optimum's fitness value (f^*), and initialization range are presented in Table A in the supplementary material. The maximum numbers of function evaluations (M_{FE}) for 10-D and 30-D problems of each function are listed in Table B in the supplementary material. The population sizes for 10-D and 30-D problems are 40 and 80, respectively.

To further evaluate the performance of CLPSO-LS, the CEC2013 benchmark functions for the special session of multimodal function optimization [45] are also tested. There are 20 multimodal functions with different characteristics in CEC2013 benchmarks, and the detailed description of these functions can be found in [45]. The population size and M_{FE} of each function in CEC2013 benchmark are presented in Table C in the supplementary material.

All experiments run 50 times independently. Termination condition is that the global optimum is found or M_{FE} is reached. Experiments are run in MATLAB on a PC with the 64-bit win10 professional operating system, 8 GB RAM and 2.40 GHz processor. Two different LS methods are employed in the proposed algorithm. Our proposed algorithm is collectively called CLPSO-LS. Specifically, CL_{BFGS} is a CLPSO-LS algorithm based on adaptive starting of BFGS quasi-Newton method. Similarly, $CL_{N\text{-}M}$ is a CLPSO-LS algorithm based on adaptive starting of Nelder–Mead method.

B. Experimental Results

The mean and standard deviation values of performance on 10-D and 30-D tested multimodal functions are shown in Table I. Two CLPSO-LS algorithms and CLPSO are tested.

The proposed algorithms converge to better solutions compared with CLPSO for most of tested multimodal functions' 10-D and 30-D problems, which can be easily concluded from Table I. First, we discuss the results of 10-D problems. The best CLPSO-LS algorithm, i.e., CL_{BFGS}, improves the performance by six orders of magnitude compared to CLPSO in F1. In F3 and F4, CL_{BFGS} converges to the global optimum at each run. Their performance is significantly better than that of CLPSO. In F5, F8, F9, and F10, both CLPSO-LS algorithms perform better than CLPSO by improving the performance for at least eight orders of magnitude. For the composition function F13, CL_{BFGS} increases the performance by 12 orders of

TABLE I
MEAN FOR 10-D AND 30-D MULTIMODAL FUNCTIONS

Function	CLPSO		CL_{BFGS}			CL_{N-M}		
	mean	std.	mean	std.	*h*	mean	std.	*h*
F1-D10	1.97E-03	9.74E-04	**6.68E-09**	2.35E-08	**1**	7.81E-05	2.45E-04	**1**
F2-D10	5.50E-04	1.54E-03	**2.96E-04**	1.46E-03	**1**	3.01E-04	1.47E-03	**1**
F3-D10	3.95E-04	5.95E-04	**0.00E+00**	0.00E+00	**1**	3.39E-07	1.85E-06	**1**
F4-D10	2.00E-04	2.47E-04	**0.00E+00**	0.00E+00	**1**	1.93E-06	9.71E-06	**1**
F5-D10	1.24E-01	1.38E-01	1.18E-11	1.60E-12	**1**	**2.55E-12**	1.69E-12	**1**
F6-D10	4.08E-03	2.53E-03	1.31E-06	1.56E-06	**1**	**3.90E-08**	1.96E-07	**1**
F7-D10	6.57E-04	1.82E-03	**6.41E-04**	2.22E-03	**1**	**6.41E-04**	2.22E-03	**1**
F8-D10	1.25E-04	1.19E-04	5.89E-13	1.88E-13	**1**	**1.22E-14**	1.47E-14	**1**
F9-D10	7.15E-05	7.40E-05	6.14E-13	1.66E-13	**1**	**8.53E-15**	8.17E-15	**1**
F10-D10	1.63E-02	2.67E-02	1.24E-11	2.41E-11	**1**	**3.58E-12**	3.81E-12	**1**
F11-D10	7.44E-03	5.50E-03	**2.88E-08**	1.10E-07	**1**	5.84E-05	2.73E-04	**1**
F12-D10	2.28E-02	2.41E-02	1.49E-06	1.44E-06	**1**	**1.17E-08**	4.90E-08	**1**
F13-D10	5.98E-01	2.35E+00	**3.33E-13**	1.78E-13	**1**	1.38E-04	9.73E-04	**1**
F1-D30	1.18E-01	2.01E-02	**1.18E-07**	2.14E-07	**1**	6.04E-02	3.29E-02	**1**
F2-D30	1.44E-03	5.91E-04	**4.21E-13**	2.53E-12	**1**	5.21E-04	6.82E-04	**1**
F3-D30	4.70E-01	1.84E-01	**3.55E-17**	2.51E-16	**1**	3.57E-02	2.51E-02	**1**
F4-D30	5.61E-01	2.83E-01	**6.00E-02**	2.40E-01	**1**	1.05E-01	2.55E-01	**1**
F5-D30	1.02E+01	5.14E+00	**3.73E-11**	5.60E-12	**1**	1.23E-10	1.36E-10	**1**
F6-D30	2.53E-02	8.62E-03	**2.08E-06**	1.00E-06	**1**	9.47E-05	6.55E-04	**1**
F7-D30	6.95E-05	3.79E-05	**8.63E-12**	3.37E-12	**1**	1.20E-06	3.68E-06	**1**
F8-D30	2.71E-01	1.07E-01	**1.94E-12**	3.28E-13	**1**	1.09E-03	2.58E-03	**1**
F9-D30	1.40E-03	7.18E-04	**1.90E-12**	2.98E-13	**1**	1.23E-06	8.71E-06	**1**
F10-D30	5.60E-01	3.42E-01	**7.38E-10**	2.81E-09	**1**	7.14E-02	2.34E-01	**1**
F11-D30	1.28E-01	3.66E-02	**2.60E-07**	3.02E-07	**1**	6.14E-02	3.83E-02	**1**
F12-D30	5.05E-01	3.19E-01	**7.54E-05**	1.27E-04	**1**	1.67E-02	6.26E-02	**1**
F13-D30	1.48E-02	2.55E-02	**5.88E-13**	6.72E-13	**1**	1.92E-05	9.23E-05	**1**
F14-D30	3.53E+02	6.21E+01	**2.05E+02**	5.20E+01	**1**	2.12E+02	5.82E+01	**1**
F15-D30	3.20E+02	2.65E+01	**1.63E+02**	5.44E+01	**1**	1.67E+02	5.44E+01	**1**
F16-D30	7.09E+02	5.85E+01	6.42E+02	1.27E+02	**1**	**5.07E+02**	1.19E+01	**1**

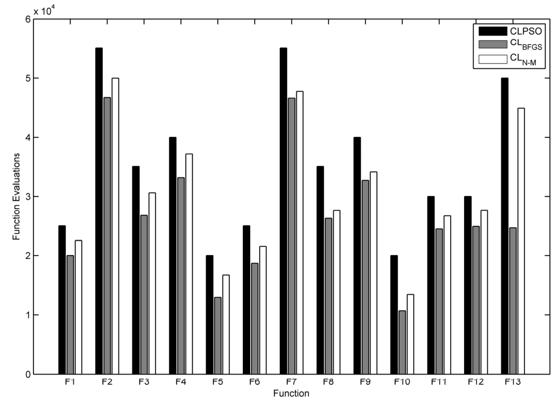

Fig. 4. Number of function evaluations on 10-D multimodal functions.

TABLE II
SUCCESS RATE OF CLPSO-LS ALGORITHMS AT DIFFERENT θ

Algorithm	0.94	0.95	0.96	0.97
CL_{BFGS}	0.9877	0.9877	0.9692	0.9385
CL_{N-M}	0.9231	0.9292	0.9415	0.9138

TABLE III
SUCCESS RATE OF CLPSO-LS ALGORITHMS AT DIFFERENT β

Algorithm	1/5	1/4	1/3	1/2
CL_{BFGS}	0.9600	0.9723	0.9877	0.9877
CL_{N-M}	0.8769	0.8954	0.9292	0.9569

magnitude. The best one improves the performance by about 5–6 orders of magnitude in F6, F11, and F12. They are shifted or rotated based on the Ackley function, and thus we have the similar results. Second, for 30-D problems, CL_{BFGS} performs the best in F1 to F15, and CL_{N-M} performs the best in F16. Especially in F2, F3, F5, and F8, the error value of CL_{BFGS} is more than 10 orders of magnitude better than that of CLPSO. It should be noted that for F14-D30, F15-D30, and F16-D30, local optima are obtained if their error values are around 10^2 during optimization. The statistical results show that the proposed algorithms converge to better solutions than CLPSO, which matches with the theoretical analysis and indicates the proposed method is also excellent in a converging-to-local-optima case.

Furthermore, the Wilcoxon rank sum test is conducted to analyze the results of 50 individual trials of CLPSO versus CL_{BFGS} and CLPSO versus CL_{N-M} for each function. Taking CLPSO versus CL_{BFGS} as an example, when CL_{BFGS} is better than CLPSO and the h value of Wilcoxon rank sum test equals 1, it means that CL_{BFGS} is significantly better than CLPSO at the 5% significance level. From the h value in Table I, CL_{BFGS} and CL_{N-M} are both significantly better than that of CLPSO for all 29 tests. So, overall, our proposed algorithm is significantly better than the original CLPSO algorithm.

From the standard deviation in Table I, we conclude that the values of CLPSO-LS are much smaller than those of CLPSO. This indicates that the performance of CLPSO-LS is more robust than that of CLPSO.

The total computational cost of LS is the sum of computation cost of LS and all objective evaluation cost. Compared with the objective function evaluation cost, its computation cost is negligible. Therefore, we use the number of objective evaluations to evaluate the computational cost of LS. It can be intuitively seen from the histogram that CLPSO-LS algorithms use fewer function evaluations than CLPSO (Fig. 4). By considering the results in Table I, it can be concluded that the proposed algorithm uses fewer computing resources to get better performance than CLPSO. Similar conclusion can be drawn for the 30-D problems.

It can be concluded that the proposed algorithms achieve better performance by using fewer computation resources, and this is applicable to problems of different dimensions.

Sensitivity analysis is performed to check how the introduced parameters affects the performance and efficiency of the proposed algorithm. One parameter is the reduction ratio of Q_E, i.e., θ. The other one is the percentage of particles to be selected, i.e., β. Two statistical indicators are utilized to describe it. One is the success rate of accelerating convergence, which refers to the degree that the parameters affect the convergence. To quantitatively assess the effects of parameter settings on the acceleration effect, the median is the second index to evaluate sensitivity.

Assume that n functions are tested, and each experiment runs k times independently. a_{ij} represents the ratio of error values of CLPSO-LS to that of CLPSO when the jth independent experiment of function i is tested. When this ratio is less than the given threshold r, CLPSO-LS is regarded to have the

TABLE IV
MEDIAN OF CLPSO-LS ALGORITHMS AT DIFFERENT θ

θ	Algorithm	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
0.94	CL_{BFGS}	6.72E-09	0.00E+00	0.00E+00	0.00E+00	1.18E-11	4.66E-07	5.22E-12	5.92E-13	6.63E-13	7.28E-12	1.48E-09	1.52E-06	2.78E-13
	CL_{N-M}	3.91E-14	1.22E-15	1.24E-14	5.33E-15	3.64E-12	5.68E-14	9.99E-16	3.55E-15	3.55E-15	1.82E-12	1.27E-08	8.72E-09	1.24E-28
0.95	CL_{BFGS}	1.01E-09	0.00E+00	0.00E+00	0.00E+00	1.27E-11	4.65E-07	6.30E-12	6.09E-13	5.81E-13	5.46E-12	5.07E-10	1.07E-06	2.76E-13
	CL_{N-M}	4.26E-14	4.44E-16	1.42E-14	5.33E-15	1.82E-12	7.82E-14	6.66E-16	5.33E-15	5.33E-15	1.82E-12	4.62E-14	4.97E-14	1.40E-28
0.96	CL_{BFGS}	7.14E-11	1.11E-16	0.00E+00	0.00E+00	1.27E-11	4.75E-07	6.39E-12	6.25E-13	5.77E-13	6.37E-12	2.87E-11	4.84E-07	2.75E-13
	CL_{N-M}	2.13E-14	3.00E-15	5.33E-15	8.88E-15	1.82E-12	4.26E-14	3.33E-16	3.55E-15	3.55E-15	2.73E-12	6.39E-14	1.07E-13	1.72E-28
0.97	CL_{BFGS}	2.15E-10	1.11E-16	0.00E+00	0.00E+00	1.27E-11	5.37E-07	1.10E-11	6.22E-13	6.73E-13	5.46E-12	5.75E-12	4.80E-07	2.76E-13
	CL_{N-M}	2.13E-14	9.99E-16	5.33E-15	1.07E-14	2.73E-12	4.26E-14	9.99E-16	7.11E-15	3.55E-15	2.73E-12	4.97E-14	9.24E-14	1.83E-28

TABLE V
MEDIAN OF CLPSO-LS ALGORITHMS AT DIFFERENT β

β	Algorithm	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
1/5	CL_{BFGS}	8.62E-10	1.11E-16	0.00E+00	0.00E+00	1.27E-11	3.94E-07	6.74E-12	6.00E-13	6.08E-13	8.19E-12	3.33E-09	1.42E-06	2.79E-13
	CL_{N-M}	2.84E-14	2.34E-07	5.33E-15	5.33E-15	2.73E-12	7.46E-14	2.22E-16	5.33E-15	5.33E-15	1.82E-12	2.45E-04	1.54E-09	1.78E-28
1/4	CL_{BFGS}	7.69E-10	1.11E-16	0.00E+00	0.00E+00	1.18E-11	4.65E-07	5.04E-12	6.64E-13	6.15E-13	6.37E-12	4.62E-10	1.57E-06	2.76E-13
	CL_{N-M}	2.84E-14	6.65E-07	7.11E-15	1.24E-14	2.73E-12	3.91E-14	2.22E-16	5.33E-15	1.78E-15	1.82E-12	2.50E-11	1.39E-08	1.54E-28
1/3	CL_{BFGS}	1.01E-09	0.00E+00	0.00E+00	0.00E+00	1.27E-11	4.65E-07	6.30E-12	6.09E-13	5.81E-13	5.46E-12	5.07E-10	1.07E-06	2.76E-13
	CL_{N-M}	4.26E-14	4.44E-16	1.42E-14	5.33E-15	1.82E-12	7.82E-14	6.66E-16	5.33E-15	5.33E-15	1.82E-12	4.62E-14	4.97E-14	1.40E-28
1/2	CL_{BFGS}	9.96E-11	0.00E+00	0.00E+00	0.00E+00	1.18E-11	1.65E-06	5.28E-12	6.15E-13	5.76E-13	5.46E-12	1.40E-09	1.74E-06	2.73E-13
	CL_{N-M}	2.84E-14	1.78E-09	3.55E-15	5.33E-15	1.82E-12	4.62E-14	4.44E-16	5.33E-15	7.11E-15	2.73E-12	4.62E-14	6.75E-14	2.08E-28

acceleration effect on the jth run of this multimodal function. Thus, the success rate s is calculated as

$$s = \frac{p}{k * n} \tag{22}$$

where p

$$p = \sum_{i=1}^{n} \sum_{j=1}^{k} (a_{ij} < r). \tag{23}$$

It calculates the sum of the tests whose a_{ij} is less than r. To ensure clear acceleration effect, we believe that the accuracy should be improved at least 100 times. So, the threshold r is assigned to be 0.01 in this paper. Four values are chosen for θ (Table II) and four values are selected for β (Table III).

First, success rate is adopted to analyze the sensitivity of parameter θ, as shown in Table II. Only slight difference exists for different θ. It means that the success rate is insensitive to parameter θ. Second, the median values (Table IV) vary from 5.92E-13 to 6.25E-13 when CL_{BFGS} solves F8 with different parameter θ, which is slightly different from each other. Similar results can be observed at other multimodal functions. CL_{N-M} obtains the median values with slight difference at different θ too. To conclude, the proposed algorithm is robust when different θ is adopted.

The success rate values at different parameter β are given in Table III. Very slight difference is observed from the two algorithms. This indicates that the success rate is insensitive to parameter β. From the analysis, the larger β, the higher success rate. This is because more particles are chosen for the calculation of Q_E when β is larger. More iterations are needed for Q_E to decline to a given value when more particles are considered. Thus, LS starts later, and the success rate is greater. The test results agree with this trend. In addition, the median values at different parameter β vary slightly for each algorithm (Table V). In closing, no obvious difference is observed, and the proposed algorithm is insensitive to θ and β.

Furthermore, the results of CLPSO-LS at different parameters are all superior to that of CLPSO. To sum up, the performance of CLPSO-LS is excellent and robust. We suggest that [0.93, 0.98] and [0.2, 0.5] are recommended for θ and β, respectively. In this paper, we let $\theta = 0.95$ and $\beta = 1/3$.

To explore the effect of an initialization range on the algorithm performance, numerical experiments for CLPSO and CLPSO-LS with a biased initialization range are carried out. Table VI presents the specific biased initialization range [28]. The results are given in Table VII. By comparing with the results in Table I, it shows that CLPSO and CLPSO-LS are insensitive to the initialization range and the performance of CLPSO-LS is better than that of CLPSO. This is consistent with the conclusion that CLPSO is robust to initialization [28].

C. Comparison With Other PSO Variants

In order to further evaluate the performance of the proposed algorithm, ten state-of-the-art PSO variants are selected for comparison. Specifically, they are ELPSO [46], DMS-L-PSO [34], DNLPSO [29], HCLPSO [30], SRPSO [47], DMeSR-PSO [23], UPSO [48], FIPS [49], FDR-PSO [50], and CPSO-H [51]. DMS-L-PSO is a dynamic multiswarm PSO with LS, whose LS starting strategy is fixed and

TABLE VI
BIASED INITIALIZATION RANGE FOR SENSITIVITY ANALYSIS

Function	Search Range	Initialization range
F1	$[-32.768, 32.768]^D$	$[-32.768, 16]^D$
F2	$[-600, 600]^D$	$[-600, 200]^D$
F3	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
F4	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
F5	$[-500, 500]^D$	$[-500, 200]^D$
F6	$[-32.768, 32.768]^D$	$[-32.768, 16]^D$
F7	$[-600, 600]^D$	$[-600, 200]^D$
F8	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
F9	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
F10	$[-500, 500]^D$	$[-500, 200]^D$
F11	$[-32.768, 32.768]^D$	$[-32.768, 16]^D$
F12	$[-32.768, 32.768]^D$	$[-32.768, 16]^D$
F13	$[-5, 5]^D$	$[-5, 2]^D$

TABLE VII
PERFORMANCE OF CLPSO AND CLPSO-LS WITH BIASED INITIALIZATION RANGE

Function	CLPSO		CL_{BFGS}		$CL_{N\text{-}M}$	
	mean	std.	mean	std.	mean	std.
F1	1.69E-03	9.39E-04	**4.68E-08**	2.14E-07	6.99E-05	1.70E-04
F2	7.43E-04	2.15E-03	**5.92E-04**	2.05E-03	6.00E-04	2.05E-03
F3	1.92E-04	2.23E-04	**0.00E+00**	0.00E+00	1.28E-05	6.37E-05
F4	1.06E-04	1.55E-04	**0.00E+00**	0.00E+00	5.28E-07	1.50E-06
F5	1.44E-02	1.64E-02	1.19E-11	1.03E-12	**3.57E-12**	4.18E-12
F6	3.98E-03	3.00E-03	1.64E-06	1.96E-06	**1.33E-08**	6.26E-08
F7	8.58E-04	2.10E-03	**5.92E-04**	2.05E-03	**5.92E-04**	2.05E-03
F8	1.27E-04	1.73E-04	6.65E-13	1.76E-13	**7.03E-15**	6.21E-15
F9	6.30E-05	4.86E-05	6.12E-13	1.52E-13	**5.26E-15**	4.43E-15
F10	2.15E-01	5.23E-01	2.04E-10	8.69E-10	**3.31E-12**	2.45E-12
F11	4.10E-02	2.26E-02	**2.57E-08**	5.51E-08	2.04E-03	4.78E-03
F12	1.54E-01	2.01E-01	1.46E-06	1.47E-06	**6.20E-07**	1.77E-06
F13	2.04E+00	7.59E+00	**4.70E-13**	3.66E-13	2.56E-03	1.28E-02

predetermined, which is different from our adaptive LS starting strategy. DNLPSO and HCLPSO are improved based on CLPSO.

The mean and standard deviation values of ten PSO variants on 30-D tested functions are presented in Table VIII. The tested results of 10-D problems are listed in Table D in supplementary material. Table IX displays the mean and standard deviation values of the CEC2013 benchmark for the special session of multimodal function optimization. It can be seen that the proposed algorithm performs basically much better than other algorithms. The standard deviation indicates that theirs are consistently smaller than those of comparison PSO variants. It means that the performance of CLPSO-LS is more robust.

Furthermore, the ranking analysis results based on the mean values of PSO variants are presented to understand our conclusions intuitively. CL_{BFGS} is top ranked and is superior to its peers. $CL_{N\text{-}M}$ is ranked the second. DMS-L-PSO and HCLPSO are next, which are better than CLPSO. From the results in Table X and Table E in the supplementary material, it is worth noting that the rankings of our proposed algorithms are relatively better than other PSO variants for most multimodal functions, implying that our adaptive LS starting strategy outperforms canonical PSO algorithms for these benchmark problems. This demonstrates the effectiveness of the CLPSO-LS in providing robust performance.

The Wilcoxon rank sum test is performed on the ranking between CLPSO-LS and other PSO variants to analyze their statistical significance. In Table F in the supplementary material, h value equaling 1 means that CL_{BFGS} is significantly better than the compared algorithm with 95% confidence. This can also be seen from the p value. The p value less than 0.05 means that the compared algorithm significantly differs from CL_{BFGS}.

The results show that CL_{BFGS} and $CL_{N\text{-}M}$ are both significantly better than all compared PSO variants, as shown in Table X and Tables F and G in the supplementary material.

Convergence curve intuitively shows an evolution process of the best solution over iterations. Because of the similarity, we select the evolution curve of 10-D F5 only as a representative

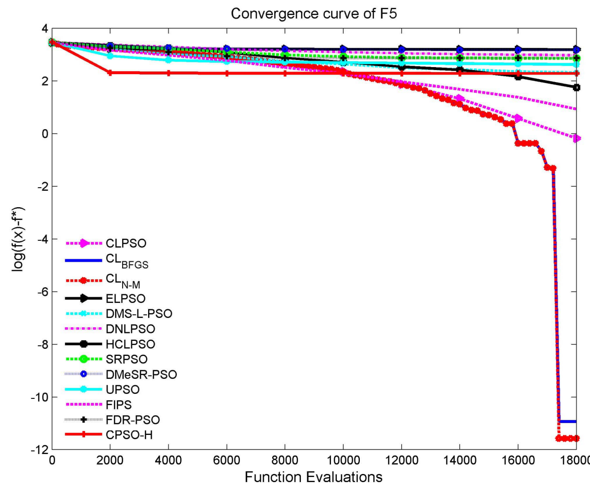

Fig. 5. Convergence curve of different PSO variants for 10-D F5.

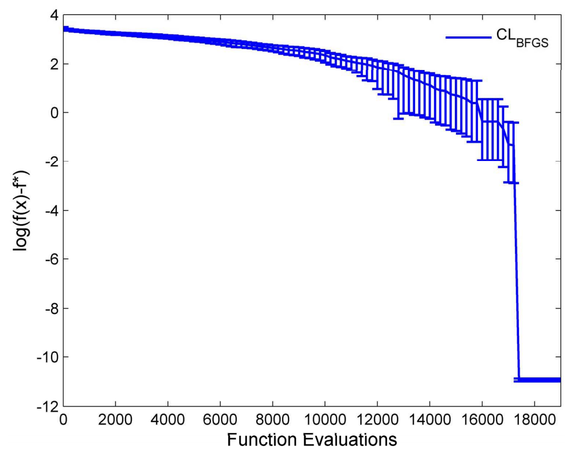

Fig. 6. Convergence curve of CL_{BFGS} with error bar for 10-D F5.

TABLE VIII
MEAN AND STANDARD DEVIATION OF 30-D MULTIMODAL FUNCTIONS OF PSO VARIANTS

Func.	Metrics	ELPSO	DMS-L-PSO	DNLPSO	HCLPSO	SRPSO	DMeSR-PSO	UPSO	FIPS	FDR-PSO	CPSO-H
F1	mean	2.71E+00	4.36E-08	2.07E-05	3.53E-11	2.24E+00	3.14E-08	4.88E-09	8.19E-01	3.12E-14	2.24E-03
	std.	5.46E-01	8.11E-08	8.84E-05	3.53E-11	2.16E+00	1.59E-08	2.50E-09	1.57E-01	7.47E-15	7.21E-04
F2	mean	1.78E-01	4.88E-03	7.06E-03	0.00E+00	9.94E-01	8.65E-03	8.88E-18	1.59E-01	7.53E-03	1.83E-02
	std.	9.56E-02	5.79E-03	7.17E-03	0.00E+00	1.33E+00	1.69E-02	6.28E-17	5.87E-02	1.15E-02	2.41E-02
F3	mean	3.41E+01	2.23E+01	2.22E+01	1.31E+01	6.88E+01	4.33E+00	6.77E+01	1.04E+02	2.73E+01	1.98E-07
	std.	8.42E+00	5.75E+00	6.96E+00	3.76E+00	4.31E+01	1.56E+00	1.17E+01	9.97E+00	7.39E+00	2.06E-07
F4	mean	3.90E+01	2.18E+01	2.61E+01	1.83E+01	5.75E+01	2.02E+00	7.74E+01	8.50E+01	1.59E+01	8.77E-09
	std.	1.61E+01	4.52E+00	6.14E+00	3.02E+00	2.17E+01	1.12E+00	1.39E+01	1.05E+01	4.83E+00	1.25E-08
F5	mean	5.47E+03	3.70E+03	6.29E+03	1.69E+03	4.26E+03	6.54E+02	3.10E+03	1.14E+03	3.27E+03	1.26E+02
	std.	7.99E+02	8.88E+02	6.94E+02	3.60E+02	1.62E+03	2.20E+02	3.86E+02	4.61E+02	7.23E+02	1.16E+02
F6	mean	3.41E+00	3.49E-08	1.84E-06	6.26E-11	4.17E+00	1.02E-09	4.20E-09	5.33E-01	1.77E-14	1.94E-03
	std.	1.14E+00	4.84E-08	3.50E-06	4.87E-11	5.36E+00	8.27E-10	1.86E-09	8.64E-02	5.63E-15	6.20E-04
F7	mean	5.19E-01	8.04E-03	3.83E-03	2.22E-18	2.47E+00	6.06E-03	1.48E-04	9.35E-02	1.02E-02	1.84E-02
	std.	1.43E+00	8.49E-03	5.56E-03	1.57E-17	8.58E+00	9.53E-03	1.05E-03	4.88E-02	1.31E-02	3.22E-02
F8	mean	7.10E+01	4.10E+01	9.79E+01	1.69E+01	7.87E+01	1.87E+00	6.15E+01	7.45E+01	2.83E+01	7.96E-02
	std.	1.59E+01	1.06E+01	2.34E+01	5.10E+00	4.77E+01	1.26E+00	1.41E+01	1.17E+01	8.49E+00	2.73E-01
F9	mean	7.59E+01	3.70E+01	8.29E+01	2.28E+01	7.56E+01	1.58E+00	6.74E+01	6.27E+01	1.82E+01	6.00E-02
	std.	1.70E+01	9.58E+00	1.98E+01	3.58E+00	3.68E+01	1.24E+00	1.21E+01	1.04E+01	6.14E+00	2.40E-01
F10	mean	3.44E+03	2.25E+03	3.77E+03	3.81E+02	3.44E+03	1.64E+02	2.60E+03	8.07E+01	2.04E+03	3.55E+01
	std.	7.32E+02	4.14E+02	6.43E+02	1.89E+02	1.32E+03	1.49E+02	5.53E+02	5.63E+01	5.41E+02	5.98E+01
F11	mean	3.04E+00	9.31E-02	9.76E-07	4.21E-14	2.24E+00	2.31E-02	1.35E-08	1.73E-01	1.20E-01	1.84E+00
	std.	8.69E-01	2.82E-01	6.00E-06	1.24E-14	1.67E+00	1.63E-01	9.54E-08	3.35E-02	3.32E-01	2.52E+00
F12	mean	3.30E+00	9.56E-02	3.73E-02	4.35E-14	2.82E+00	1.27E-12	3.29E-02	1.16E-01	1.72E-01	3.80E+00
	std.	9.60E-01	3.84E-01	1.84E-01	2.19E-14	3.46E+00	1.62E-12	2.33E-01	2.05E-02	4.44E-01	5.84E+00
F13	mean	7.75E+01	2.80E+01	7.20E+01	2.44E-21	6.91E+01	2.10E-07	0.00E+00	6.08E-04	1.80E+01	8.20E+01
	std.	9.39E+01	6.71E+01	1.09E+02	1.72E-20	9.53E+01	1.49E-06	0.00E+00	2.23E-03	4.82E+01	1.29E+02
F14	mean	5.17E+02	3.80E+02	4.29E+02	3.36E+02	3.62E+02	4.43E+02	3.39E+02	3.66E+02	3.47E+02	3.25E+02
	std.	1.59E+02	6.60E+01	1.23E+02	6.31E+01	1.33E+02	6.44E+01	6.25E+01	3.47E+01	1.20E+02	1.08E+02
F15	mean	3.31E+02	1.81E+02	2.79E+02	1.67E+02	2.67E+02	2.20E+02	1.43E+02	2.51E+02	2.26E+02	4.24E+02
	std.	1.50E+02	1.30E+02	1.39E+02	8.11E+01	1.32E+02	1.64E+02	2.73E+01	3.26E+01	1.68E+02	1.66E+02
F16	mean	9.49E+02	5.10E+02	9.47E+02	5.00E+02	8.52E+02	5.10E+02	5.00E+02	5.12E+02	6.43E+02	7.08E+02
	std.	2.82E+02	5.48E+01	3.13E+02	4.59E-11	1.25E+02	5.48E+01	3.99E-06	4.29E+00	2.24E+02	3.07E+02

to demonstrate the features of different algorithms (Fig. 5). The convergence curve of CL_{BFGS} with error bars is shown in Fig. 6. Since CLPSO-LS is an improved CLPSO based on the adaptive LS starting strategy, it is not redundant to make a deep comparison between it and two other improved CLPSO algorithms, i.e., DNLPSO and HCLPSO, and the comparison between it and LS-based PSO, i.e., DMS-L-PSO.

Two different subgroups are employed in HCLPSO and they are responsible for the exploration and exploitation separately, aiming at balancing the abilities of LS and global exploration. However, its global ability is weaker than that of CLPSO-LS. Because only particles in one subgroup are utilized to explore global regions in HCLPSO, while all particles are used for global search in CLPSO-LS. Furthermore, the LS ability of traditional LS method introduced in CLPSO-LS is stronger than that of HCLPSO. As a result, the performance of CLPSO-LS is much better than that of HCLPSO, which can be seen from the evolution curve in Fig. 5. To improve the local search ability of CLPSO, the gbest information that has been removed from the velocity update formula of CLPSO is still employed in DNLPSO. However, it leads to the loss of the population diversity. Therefore, DNLPSO visibly falls behind CLPSO and our proposed algorithms in its later iterations as shown in Fig. 5.

DMS-L-PSO starts LS in its early iterations, so its local convergence is stronger at the beginning. However, lots of early started LSs not only occupy the computation resources of the PSO global exploration, but also increase the risk of trapping into local minima. Thus, the overall performance of DMS-L-PSO is weakened. However, an adaptive LS starting strategy

TABLE IX
MEAN AND STANDARD DEVIATION OF CEC2013 BENCHMARK FUNCTIONS OF PSO VARIANTS

Algorithm	Metrics	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
CLPSO	mean	8.25E-01	3.09E-06	4.16E-03	7.89E-02	3.82E-04	3.80E+00	2.89E-04	1.12E+02	3.46E-04	8.65E-04
	std.	8.50E-01	1.07E-05	1.31E-02	1.19E-01	5.12E-04	4.30E+00	3.74E-04	9.88E+01	5.45E-04	9.48E-04
CL_{BFGS}	mean	5.27E-03	0.00E+00	3.42E-03	0.00E+00	0.00E+00	2.84E-14	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	std.	5.25E-03	0.00E+00	1.30E-02	0.00E+00	0.00E+00	3.25E-14	0.00E+00	0.00E+00	0.00E+00	0.00E+00
$CL_{N\text{-}M}$	mean	3.69E-08	0.00E+00	3.42E-03	0.00E+00	0.00E+00	1.43E-13	4.44E-17	6.04E-06	1.34E-11	2.96E-16
	std.	1.98E-07	0.00E+00	1.30E-02	0.00E+00	0.00E+00	5.36E-13	2.43E-16	2.40E-05	5.57E-11	1.62E-15
ELPSO	mean	0.00E+00	4.75E-06	2.60E-02	3.46E-04	2.08E-06	2.65E-01	7.22E-05	1.29E+02	3.85E-08	4.71E-04
	std.	0.00E+00	1.72E-05	2.58E-02	5.42E-04	2.49E-06	4.34E-01	1.76E-04	4.28E+02	1.07E-07	1.22E-03
DMS-L-PSO	mean	2.01E+01	9.66E-12	3.58E-02	4.36E-14	3.74E-15	1.96E-11	4.94E-09	3.99E-10	4.01E-12	1.36E-10
	std.	2.02E+01	2.63E-11	5.82E-02	3.05E-14	8.04E-15	8.03E-11	1.88E-08	1.02E-09	1.81E-11	3.40E-10
DNLPSO	mean	2.87E+00	1.83E-05	1.20E-02	3.32E-02	1.12E-04	1.05E+01	9.61E-04	2.54E+02	1.05E-03	3.45E-03
	std.	2.79E+00	4.02E-05	2.05E-02	8.31E-02	2.59E-04	1.29E+01	2.88E-03	4.14E+02	4.25E-03	6.17E-03
HCLPSO	mean	4.11E+00	1.77E-06	9.47E-03	3.80E-02	3.66E-04	4.84E+00	5.85E-04	1.20E+02	2.56E-04	1.51E-03
	std.	1.22E+01	4.51E-06	1.92E-02	4.58E-02	6.46E-04	6.40E+00	6.56E-04	2.05E+02	3.62E-04	2.70E-03
SRPSO	mean	2.50E+00	1.53E-06	1.83E-03	2.68E-03	2.70E-06	4.49E+00	1.88E-04	4.22E+01	8.48E-05	8.07E-05
	std.	7.21E+00	5.70E-06	9.35E-03	7.41E-03	4.29E-06	1.17E+01	3.55E-04	1.01E+02	2.12E-04	2.44E-04
DMeSR-PSO	mean	1.66E+01	1.07E-09	6.16E-03	2.14E-03	4.33E-04	8.64E+00	5.66E-04	2.37E+02	3.02E-04	9.76E-05
	std.	1.84E+01	3.12E-09	1.57E-02	6.02E-03	1.64E-03	1.32E+01	2.35E-03	3.49E+02	6.21E-04	3.57E-04
UPSO	mean	6.21E+00	1.15E-07	8.81E-03	1.18E-03	1.77E-05	4.17E+00	5.54E-05	2.83E+00	3.23E-07	8.57E-06
	std.	1.17E+01	2.43E-07	1.67E-02	2.21E-03	1.88E-05	1.16E+01	5.89E-05	9.97E+00	1.02E-06	1.40E-05
FIPS	mean	1.78E+01	1.31E-05	2.15E-02	6.75E-02	6.03E-04	1.06E+01	1.56E-03	3.25E+02	1.12E-03	7.26E-03
	std.	1.86E+01	2.14E-05	4.39E-02	7.09E-02	6.36E-04	7.99E+00	1.54E-03	2.98E+02	1.55E-03	9.94E-03
FDR-PSO	mean	1.81E+01	2.74E-08	3.70E-02	2.14E-03	8.31E-06	1.12E+00	1.19E-04	3.13E-01	2.79E-05	2.61E-03
	std.	1.95E+01	8.54E-08	1.05E-01	5.81E-03	2.03E-05	2.18E+00	3.61E-04	1.07E+00	1.51E-04	8.01E-03
CPSO-H	mean	1.20E+01	7.63E-12	2.73E-02	1.22E-01	1.01E-03	1.67E+01	6.17E-03	7.54E+02	4.51E-03	6.10E-03
	std.	1.86E+01	2.53E-11	5.91E-02	2.29E-01	1.88E-03	2.38E+01	1.34E-02	8.39E+02	8.59E-03	1.15E-02

Algorithm	Metrics	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
CLPSO	mean	4.08E-01	7.28E-01	1.08E+00	2.24E+00	5.58E+00	9.46E-01	1.86E+00	2.18E-02	1.43E+00	2.54E-01
	std.	6.44E-01	1.55E+00	2.26E+00	3.94E+00	5.82E+00	1.14E+00	2.13E+00	4.89E-02	1.01E+00	1.81E-01
CL_{BFGS}	mean	2.36E-16	3.94E-18	0.00E+00	1.07E+00	1.55E+00	3.93E-01	9.55E-01	5.19E-14	9.26E-02	1.43E-08
	std.	4.50E-16	3.69E-18	0.00E+00	2.48E+00	3.93E+00	1.31E+00	1.26E+00	2.84E-13	3.53E-01	6.30E-08
$CL_{N\text{-}M}$	mean	4.94E-22	1.08E-21	0.00E+00	0.00E+00	4.90E-01	8.00E-10	2.35E-01	1.52E-02	8.62E-02	1.03E-04
	std.	1.88E-21	5.85E-21	0.00E+00	0.00E+00	2.23E+00	3.33E-09	8.05E-01	8.31E-02	3.29E-01	2.55E-04
ELPSO	mean	6.72E-05	4.61E+00	3.31E-04	6.39E-07	2.30E+00	5.17E+00	3.27E+00	1.02E+02	3.54E+01	7.45E+01
	std.	1.40E-04	2.51E+01	4.91E-04	1.49E-06	2.92E+00	1.96E+01	4.94E+00	1.64E+02	5.20E+01	7.61E+01
DMS-L-PSO	mean	2.23E-02	3.84E-02	8.37E-03	7.59E-02	4.41E-01	1.57E-11	3.20E-01	0.00E+00	0.00E+00	0.00E+00
	std.	9.18E-02	2.10E-01	3.63E-02	4.09E-01	1.73E+00	6.25E-11	8.42E-01	0.00E+00	0.00E+00	0.00E+00
DNLPSO	mean	9.08E-02	1.71E-01	8.55E-01	1.03E+00	8.52E+00	7.62E+00	1.42E+01	0.00E+00	0.00E+00	1.43E+02
	std.	1.09E-01	3.77E-01	2.78E+00	2.37E+00	1.13E+01	2.93E+01	1.68E+01	0.00E+00	0.00E+00	1.22E+02
HCLPSO	mean	5.56E-02	3.52E-02	2.71E-01	2.67E-02	1.39E+00	5.53E-05	8.08E-01	1.21E-12	1.34E+00	7.66E-02
	std.	8.54E-02	7.32E-02	7.43E-01	9.83E-02	2.72E+00	2.78E-04	2.07E+00	7.32E-13	1.25E+00	1.98E-01
SRPSO	mean	2.11E-03	1.73E-01	1.63E-03	6.53E-07	9.35E-01	0.00E+00	1.31E+00	0.00E+00	0.00E+00	8.32E-01
	std.	1.15E-02	6.43E-01	5.44E-03	1.88E-06	2.06E+00	0.00E+00	2.01E+00	0.00E+00	0.00E+00	2.56E+00
DMeSR-PSO	mean	3.40E-02	9.77E+00	4.91E-02	7.97E+00	5.17E+00	9.39E+00	1.02E+00	0.00E+00	0.00E+00	1.02E+02
	std.	1.46E-01	2.61E+01	1.86E-01	3.71E+01	9.63E+00	5.14E+01	1.50E+00	0.00E+00	0.00E+00	2.63E+02
UPSO	mean	5.16E-04	5.34E-04	3.67E-03	2.66E-05	1.18E+00	2.35E-07	1.81E+00	2.08E-13	5.53E+00	1.53E+01
	std.	9.37E-04	7.63E-04	4.51E-03	5.64E-05	2.64E+00	1.28E-06	3.10E+00	5.38E-13	2.92E+00	1.48E+01
FIPS	mean	1.93E-01	9.53E-01	4.79E-01	4.66E-01	2.98E+00	3.98E-02	5.28E-01	1.44E-04	3.29E+00	8.47E+00
	std.	3.29E-01	2.37E+00	7.15E-01	6.07E-01	4.24E+00	1.58E-01	2.84E+00	5.65E-04	1.54E+00	2.85E+00
FDR-PSO	mean	1.05E-04	1.83E-01	3.14E-03	2.66E-05	1.97E+00	0.00E+00	1.59E+00	1.04E-13	1.50E+00	4.29E+00
	std.	2.19E-04	4.94E-01	1.34E-02	1.41E-04	4.02E+00	0.00E+00	2.63E+00	3.95E-13	1.57E+00	2.15E+00
CPSO-H	mean	7.97E-02	1.34E+01	1.37E+00	5.10E+00	7.16E+00	7.74E+01	2.59E+01	3.92E+02	1.30E+02	2.09E+02
	std.	1.77E-01	3.39E+01	2.44E+00	1.94E+01	7.01E+00	7.93E+01	3.40E+01	1.95E+02	7.97E+01	6.93E+01

is employed in our proposed algorithm, and it triggers the LS only after GOB is found. Therefore, the strong global ability of CLPSO is maintained and the advantages of LS are fully utilized. To sum up, the proposed algorithms represented by the CL_{BFGS} and $CL_{N\text{-}M}$ are superior to other compared PSO variants.

TABLE X
RANK OF PSO VARIANTS BASED ON THE MEAN VALUES OF TESTED MULTIMODAL FUNCTIONS

Metrics	CLPSO	CL_{BFGS}	CL_{N-M}	ELPSO	DMS-L-PSO	DNLPSO	HCLPSO	SRPSO	DMeSR-PSO	UPSO	FIPS	FDR-PSO	CPSO-H
Ave. Rank	7.2857	3	3.2041	9.9388	5.2449	9.9388	5.3878	7.4082	7.5306	5.9796	9.4694	6.7143	9.3469
Overall Rank	7	1	2	12	3	12	4	8	9	5	11	6	10

VI. CONCLUSION

A new variant of PSO, CLPSO-LS, is proposed in this paper. It takes advantages of the fast convergence feature of traditional LS and CLPSO's strong global search ability. The proposed adaptive LS starting strategy effectively addresses the key issue of finding out the proper time to start LS as optimization proceeds. A rapid decline of quasi-entropy calculated from the fitness of pbest indicates the beginning of a sharp decrease of population diversity. It helps us judge whether a good number of particles have entered the GOB, which is a sign of the end of global exploration as well as the start of local exploitation. The proposed quasi-entropy is used to determine the best LS start time in CLPSO-LS.

Two canonical LS methods are applied in CLPSO-LS. Experiments on 10-D and 30-D problems of multimodal benchmark functions, as well as the CEC2013 benchmark for the special session of multimodal function optimization are carried out to test their performance. The statistical results indicate that CLPSO-LS performs drastically better than CLPSO, which means that the traditional LS methods can greatly improve the local convergence rate of CLPSO after adopting our proposed adaptive LS starting strategy.

Parameter sensitivity analysis is conducted to figure out the influence of two newly introduced parameters on the performance of our proposed algorithm. The results indicate that it is insensitive to them. Comparative experiments with ten state-of-the-art PSO variants are carried out to further validate its superiority.

We will apply this hybrid strategy to a combination of other LS methods and other advanced PSO algorithms in our future research. The simulated annealing method may be a good LS candidate, especially for the extreme complex functions, due to its global search ability.

REFERENCES

[1] Q. Kang, X. Y. Song, M. C. Zhou, and L. Li, "A collaborative resource allocation strategy for decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2018.2818175.

[2] S. Gao, S. Song, J. Cheng, Y. Todo, and M. Zhou, "Incorporation of solvent effect into multi-objective evolutionary algorithm for improved protein structure prediction," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 4, pp. 1365–1378, Jul./Aug. 2018.

[3] B. Wang, X. Xia, H. Meng, and T. Li, "Bad-scenario-set robust optimization framework with two objectives for uncertain scheduling systems," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 1, pp. 143–153, Jan. 2017.

[4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, 1995, pp. 1942–1948.

[5] Q. Yang *et al.*, "Adaptive multimodal continuous ant colony optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 191–205, Apr. 2017.

[6] Z.-J. Wang *et al.*, "Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 894–908, Dec. 2017.

[7] M. Yang *et al.*, "Efficient resource allocation in cooperative co-evolution for large-scale global optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 493–505, Aug. 2017.

[8] Q. Yang *et al.*, "Multimodal estimation of distribution algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 636–650, Mar. 2017.

[9] T. Xiong, Y. Bao, Z. Hu, and R. Chiong, "Forecasting interval time series using a fully complex-valued RBF neural network with DPSO and PSO algorithms," *Inf. Sci.*, vol. 305, pp. 77–92, Jun. 2015.

[10] G. Tian, Y. Ren, and M. C. Zhou, "Dual-objective scheduling of rescue vehicles to distinguish forest fires via differential evolution and particle swarm optimization combined algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3009–3021, Nov. 2016.

[11] H. Duan, P. Li, and Y. Yu, "A predator-prey particle swarm optimization approach to multiple UCAV air combat modeled by dynamic game theory," *IEEE/CAA J. Automatica Sinica*, vol. 2, no. 1, pp. 11–18, Jan. 2015.

[12] M. Dadgar, S. Jafari, and A. Hamzeh, "A PSO-based multi-robot cooperation method for target searching in unknown environments," *Neurocomputing*, vol. 177, pp. 62–74, Feb. 2016.

[13] A. A. Aburomman and M. B. I. Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Appl. Soft Comput.*, vol. 38, pp. 360–372, Jan. 2016.

[14] S. Zhang *et al.*, "Optimal computing budget allocation for particle swarm optimization in stochastic optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 206–219, Apr. 2017.

[15] Q. Yang *et al.*, "A level-based learning swarm optimizer for large-scale optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 578–594, Aug. 2018.

[16] M. R. Bonyadi and Z. Michalewicz, "Impacts of coefficients on movement patterns in the particle swarm optimization algorithm," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 378–390, Jun. 2017.

[17] Q. Lin *et al.*, "Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 32–46, Feb. 2018.

[18] Z. Beheshti and S. M. H. Shamsuddin, "A review of population-based meta-heuristic algorithm," *Int. J. Adv. Soft Comput. Appl.*, vol. 5, no. 1, pp. 1–35, Mar. 2013.

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

[20] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *Proc. IEEE Congr. Evol. Comput.*, 2005, pp. 522–528.

[21] W. Dong and M. C. Zhou, "A supervised learning and control method to improve particle swarm optimization algorithms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1149–1159, Jul. 2017.

[22] J. Li, J. Zhang, C. Jiang, and M. C. Zhou, "Composite particle swarm optimizer with historical memory for function optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2350–2363, Oct. 2015.

[23] M. R. Tanweer, S. Suresh, and N. Sundararajan, "Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems," *Inf. Sci.*, vol. 326, pp. 1–24, Jan. 2016.

[24] X. Liang, W. Li, P. P. Liu, Y. Zhang, and A. A. Agbo, "Social network-based swarm optimization algorithm," in *Proc. IEEE Int. Conf. Netw. Sens. Control*, 2015, pp. 360–365.

[25] W. Ma, M. Wang, and X. Zhu, "Hybrid particle swarm optimization and differential evolution algorithm for bi-level programming problem and its application to pricing and lot-sizing decisions," *J. Intell. Manuf.*, vol. 26, no. 3, pp. 471–483, Jun. 2015.

[26] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Inf. Sci.*, vol. 237, pp. 82–117, Jul. 2013.

[27] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 4th ed. New York, NY, USA: Springer, 2015.

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

[29] M. Nasir *et al.*, "A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization," *Inf. Sci.*, vol. 209, no. 5, pp. 16–36, Nov. 2012.

[30] N. Lynn and P. N. Suganthan, "Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation," *Swarm Evol. Comput.*, vol. 24, pp. 11–24, Oct. 2015.

[31] W. E. Hart, "Adaptive global optimization with local search," Ph.D. dissertation, Dept. Comput. Sci. Eng., Univ. California, San Diego, CA, USA, 1994.

[32] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 107–125, Feb. 2008.

[33] P. Moscato, “On evolution, search, optimization, genetic algorithms and martial arts—Towards memetic algorithms,” Caltech Concurrent Comput. Program, California Inst. Technol., Pasadena, CA, USA, Rep. 826, 1989.
[34] S. Z. Zhao, J. J. Liang, P. N. Suganthan, and M. F. Tasgetiren, “Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization,” in *Proc. IEEE Congr. Evol. Comput.*, 2008, pp. 3845–3852.
[35] F. Han and Q. Liu, “An improved hybrid PSO based on ARPSO and the Quasi-Newton method,” in *Proc. ICSI-CCI*, vol. 9140, 2015, pp. 460–467.
[36] X. Yu and X. Zhang, “Multiswarm comprehensive learning particle swarm optimization for solving multiobjective optimization problems,” *PLoS ONE*, vol. 12, no. 2, Feb. 2017, Art. no. e0172033.
[37] A. Borji, M. Hamidi, and A. M. E. Moghadam, “CLPSO-based fuzzy color image segmentation,” in *Proc. North Amer. Fuzzy Inf. Process. Soc.*, 2007, pp. 508–513.
[38] S. Malhotra, S. R. Sahoo, and S. Anand, “CLPSO based droop optimization technique for DC microgrid,” in *Proc. Indian Control Conf. (ICC)*, Jan. 2018, pp. 172–177.
[39] I. C. Trelea, “The particle swarm optimization algorithm: Convergence analysis and parameter selection,” *Inf. Process. Lett.*, vol. 85, no. 6, pp. 317–325, Mar. 2003.
[40] C. G. Broyden, J. E. Dennis, and J. J. Moré, “On the local and superlinear convergence of Quasi-Newton methods,” *J. Inst. Math. Appl.*, vol. 12, no. 3, pp. 223–245, Mar. 1973.
[41] Y. Shi and R. C. Eberhart, “Population diversity of particle swarms,” in *Proc. Evol. Comput.*, 2008, pp. 1063–1067.
[42] E. J. S. Pires, J. A. T. Machado, and P. B. D. M. Oliveira, “Entropy diversity in multi-objective particle swarm optimization,” *Entropy*, vol. 15, no. 12, pp. 5475–5491, Dec. 2013.
[43] J. J. Liang, P. N. Suganthan, and K. Deb, “Novel composition test functions for numerical global optimization,” in *Proc. Swarm Intell. Symp.*, 2005, pp. 68–75.
[44] P. N. Suganthan *et al.*, “Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,” Nanyang Technol. Univ., Singapore, Rep. 2005005, 2005.
[45] X. Li, A. Engelbrecht, and M. G. Epitropakis, “Benchmark functions for CEC’2013 special session and competition on niching methods for multimodal function optimization,” Evol. Comput. Mach. Learn. Group, RMIT Univ., Melbourne VIC, Australia, Tech. Rep., 2013.
[46] A. R. Jordehi, “Enhanced leader PSO (ELPSO): A new PSO variant for solving global optimization problems,” *Appl. Soft Comput.*, vol. 26, pp. 401–417, Jan. 2015.
[47] M. R. Tanweer, S. Suresh, and N. Sundararajan, “Self regulating particle swarm optimization algorithm,” *Inf. Sci.*, vol. 294, no. 10, pp. 182–202, Feb. 2015.
[48] K. E. Parsopoulos and M. N. Vrahatis, “Unified particle swarm optimization in dynamic environments,” in *Proc. EvoWorkshops*, vol. 3449, 2005, pp. 590–599.
[49] R. Mendes, J. Kennedy, and J. Neves, “The fully informed particle swarm: Simpler, maybe better,” *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 204–210, Jun. 2004.
[50] T. Peram, K. Veeramachaneni, and C. K. Mohan, “Fitness-distance-ratio based particle swarm optimization,” in *Proc. Swarm Intell. Symp.*, 2003, pp. 174–181.
[51] F. van den Bergh and A. P. Engelbrecht, “A cooperative approach to particle swarm optimization,” *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, Jun. 2004.

Yulian Cao received the B.S., M.S., and Ph.D. degrees in logistics engineering from the Wuhan University of Technology, Wuhan, China, in 2009, 2012, and 2018 respectively.

From 2012 to 2014, she was a Visiting Scholar with the University of Washington, Seattle, WA, USA. Since 2018, she has been a Post-Doctoral Fellow with the University of New South Wales, Sydney, Australia. Her current research interests include evolutionary computing, PSO algorithm, memetic algorithm, and logistics system modeling and optimization.

Han Zhang received the B.S. degree from Harbin Engineering University, Harbin, China, in 2010 and the Ph.D. degree from Tsinghua University, Beijing, China, in 2015.

From 2013 to 2014, he was a Visiting Scholar with the University of Michigan, Ann Arbor, MI, USA. He has been a Scientific Fellow with the Karlsruhe Institute of Technology, Karlsruhe, Germany, since 2015. His current research interests include heuristic optimization algorithm, nonlinear/linear iteration algorithm, and numerical partial differential equations solution.

Wenfeng Li (SM’12) received the B.S. degree from the Zhengzhou Institute of Light Industry, Zhengzhou, China, the M.S. degree from the Huazhong University of Science and Technology, Wuhan, China, in 1990, and the Ph.D. degree from the Wuhan University of Technology, Wuhan, China, in 2000,

He is a Full Professor with the School of Logistics Engineering, Wuhan University of Technology. He was a Visiting Scholar with the Royal Institute of Technology of Sweden, Stockholm, Sweden, in 2003. He was a Visiting Professor with the New Jersey Institute of Technology, Newark, NJ, USA, in 2012, and a Visiting Professor with the Center for Urban Science And Progress (CUSP) of New York University, Brooklyn, NY, USA, in 2015. He set up the Institute of Logistics and Robotics, the Center of Internet of Things (IoT) and Logistics Technologies, and the International Joint Laboratory of IoT. He and his research group have accomplished over 25 scientific research projects, with the fund from National Natural Science Foundation of China, Research and Development Project of MOST, and Research and Development Projects of Chinese provinces. His current research Interests include IoT and its application, robotics, logistics information and automation, simulation and planning of logistics, and supply chain.

Mengchu Zhou (S’88–M’90–SM’93–F’03) received the B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

He joined the New Jersey Institute of Technology, Newark, NJ, USA, in 1990, where he is currently a Distinguished Professor of electrical and computer engineering. He has over 700 publications including 12 books, over 400 journal papers (over 300 in IEEE TRANSACTIONS), 11 patents, and 28 book chapters. He is the Founding Editor of the IEEE Press Book Series on Systems Science and Engineering. His current research interests include Petri nets, intelligent automation, Internet of Things, big data, Web services, and intelligent transportation.

Dr. Zhou was a recipient of the Humboldt Research Award for U.S. Senior Scientists from the Alexander von Humboldt Foundation, the Franklin V. Taylor Memorial Award, and the Norbert Wiener Award from the IEEE Systems, Man and Cybernetics Society. He is a life member of the Chinese Association for Science and Technology, USA, and served as its President in 1999. He is a fellow of the International Federation of Automatic Control and the American Association for the Advancement of Science.

Yu Zhang received the B.S., M.S., and Ph.D. degrees in logistics engineering from the Wuhan University of Technology, Wuhan, China, in 1996, 2001, and 2007, respectively.

He is a Full Professor with the School of Logistics Engineering, Wuhan University of Technology. He was a Visiting Scholar with the Lehigh University, Bethlehem, PA, USA, in 2010. His current research interests include modeling and simulating analysis of logistic system, fleet assignment and routing, vehicle routing, and machine scheduling of port.

Wanpracha Art Chaovalitwongse (M’05–SM’11) received the Ph.D. degree in industrial and systems engineering from the University of Florida, Gainesville, FL, USA, in 2003.

He is currently 21st Century Leadership Chair in engineering, a Professor of industrial engineering, and the Co-Director of the Institute of Advanced Data Analytics at the University of Arkansas, Fayetteville, AR, USA. He has held faculty positions with Rutgers University, New Brunswick, NJ, USA, and Princeton University, Princeton, NJ, USA. From 2011 to 2016, he was a Professor of industrial and systems engineering, radiology (Joint), and bioengineering (Adjunct) with the University of Washington (UW), Seattle, WA, USA, where he also served as an Associate Director of the Integrated Brain Imaging Center, UW Medical Center. His research group conducts basic computational science, applied and translational research at the interface of engineering, medicine, and other emerging disciplines.