

Article

A Multi-Strategy Adaptive Comprehensive Learning PSO Algorithm and Its Application

Ye'e Zhang and Xiaoxia Song *

College of Computer and Network Engineering, Shanxi Datong University, Datong 037009, China; sxd_t_zye@sxdtdx.edu.cn

* Correspondence: sjsongxiaoxia@sxdtdx.edu.cn

Abstract: In this paper, a multi-strategy adaptive comprehensive learning particle swarm optimization algorithm is proposed by introducing the comprehensive learning, multi-population parallel, and parameter adaptation. In the proposed algorithm, a multi-population parallel strategy is designed to improve population diversity and accelerate convergence. The population particle exchange and mutation are realized to ensure information sharing among the particles. Then, the global optimal value is added to velocity update to design a new velocity update strategy for improving the local search ability. The comprehensive learning strategy is employed to construct learning samples, so as to effectively promote the information exchange and avoid falling into local extrema. By linearly changing the learning factors, a new factor adjustment strategy is developed to enhance the global search ability, and a new adaptive inertia weight-adjustment strategy based on an S-shaped decreasing function is developed to balance the search ability. Finally, some benchmark functions and the parameter optimization of photovoltaics are selected. The proposed algorithm obtains the best performance on 6 out of 10 functions. The results show that the proposed algorithm has greatly improved diversity, solution accuracy, and search ability compared with some variants of particle swarm optimization and other algorithms. It provides a more effective parameter combination for the complex engineering problem of photovoltaics, so as to improve the energy conversion efficiency.

Keywords: CLPSO; multi-strategy; search ability; photovoltaic optimization



Citation: Zhang, Y.; Song, X. A Multi-Strategy Adaptive Comprehensive Learning PSO Algorithm and Its Application. *Entropy* **2022**, *24*, 890. <https://doi.org/10.3390/e24070890>

Academic Editors: Ben Niu, Shuang Geng and Rong Qu

Received: 17 May 2022

Accepted: 23 June 2022

Published: 28 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Many problems in reality can be transformed into optimization problems. These optimization problems have complex characteristics, such as multiple constraints, high dimensionality, nonlinearity, and uncertainty, making them difficult to solve by the traditional optimization methods [1,2]. Therefore, an efficient new method is sought to solve these complex problems. Swarm intelligence optimization algorithms are a new evolutionary computing technology, which refers to some intelligent optimization algorithms with distributed intelligent behavior characteristics inspired by the swarm behavior of insects, herds, birds, fish, etc. [3–5]. This has become the research focus of more and more researchers. It has a special relationship with artificial life, and includes Harris hawk optimization (HHO), slime mold algorithm (SMA), artificial bee colony (ABC), firefly optimization, cuckoo search, and brainstorming optimization algorithm [6–9] for engineering scheduling, image processing, the traveling salesman problem, cluster analysis, and logistics location.

PSO is a swarm intelligence optimization technology developed by Kennedy and Eberhart [10]. The main idea is to solve the optimization problem through individual cooperation and information sharing. The PSO takes on a simple, strong parallel structure. Therefore, it has been used in multi-objective optimization, scheduling optimization, vehicle routing problems, etc. Although the PSO shows good optimization performance, it has slow convergence in solving complex optimization problems. Thus, a variety of improvement strategies for PSO are presented. Nickabadi et al. [11] presented a new adaptive

inertia weight approach. Wang et al. [12] presented a self-adaptive learning model based on PSO for solving application problems. Zhan et al. [13] presented an orthogonal learning strategy for PSO. Li and Yao [14] presented a cooperative PSO. Xu [15] presented an adaptive tuning for the parameters of PSO based on a velocity and inertia weight strategy to avoid the velocity close to zero in the early stages. Wang et al. [16] presented a hybrid PSO using a diversity mechanism and neighborhood search. Chen et al. [17] presented an aging leader and challenger PSO. Qu et al. [18] presented a distance-based PSO. Cheng and Jin [19] presented a social learning PSO based on controlling dimension-dependent parameters. Tanweer et al. [20] presented a self-regulating PSO with the best human learning. Taherkhani et al. [21] presented an adaptive PSO approach. Moradi and Gholampour [22] presented a hybrid PSO based on a local search strategy. Gong et al. [23] developed a new hybridized PSO framework with another optimization method for “learning”. Nouri et al. [24] presented an effective and distributed PSO. Wang et al. [25] presented a hybrid PSO with adaptive learning to guarantee exploitation. Aydilek [26] presented a hybrid PSO with a firefly algorithm mechanism. Xue et al. [27] presented a self-adaptive PSO. Song et al. [28] presented a variable-size cooperative co-evolutionary PSO with the idea of “divide and conquer”. Song et al. [29] presented a bare-bones PSO with mutual information.

Sources	Results and Contribution to PSO
Nickabadi et al. [11]	Designed an adaptive inertia weight strategy for PSO
Zhan et al. [13]	Designed an orthogonal learning strategy for PSO
Xu [15]	Designed an adaptive tuning strategy for the parameters for PSO
Wang et al. [16]	Developed a hybrid PSO
Cheng and Jin [19]	Developed a social learning PSO
Tanweer et al. [20]	Developed a self-regulating PSO
Moradi and Gholampour [22]	Designed a local search strategy for PSO
Gong et al. [23]	Developed a new hybridized PSO
Xue et al. [27]	Developed a self-adaptive PSO
Song et al. [28]	Developed a variable-size cooperative co-evolutionary PSO
Song et al. [29]	Developed a bare-bones PSO

The comprehensive learning PSO (CLPSO) algorithm is a variant of PSO, and has good application in multimodal problems. However, because the CLPSO algorithm uses the current search velocity and individual optimal value to update the search velocity, the search velocity value in the later iteration is very small, resulting in slow convergence and reducing the computational efficiency. In order to improve the CLPSO algorithm, researchers have conducted some useful works. Liang et al. [30] presented a variant of PSO (CLPSO) using a new learning strategy. Maltra et al. [31] presented a hybrid cooperative CLPSO by cloning fitter particles. Mahadevan and Kannan [32] presented a learning strategy for PSO to develop a CLPSO to overcome premature convergence. Ali and Khan [33] presented an attributed multi-objective CLPSO for solving well-known benchmark problems. Hu et al. [34] presented a CLPSO-based memetic algorithm. Zhong et al. [35] presented a discrete CLPSO with an acceptance criterion of SA. Lin and Sun [36] presented a multi-leader CLPSO based on adaptive mutation. Zhang et al. [37] presented a local optima topology (LOT) structure with the CLPSO for solving various functions. Lin et al. [38] presented an adaptive mechanism to adjust the comprehensive learning probability of CLPSO. Wang and Liu [39] presented a novel saturated control method for a quadrotor to achieve three-dimensional spatial trajectory tracking with heterogeneous CLPSO. Cao et al. [40] presented a CLPSO with local search. Chen et al. [41] presented a grey-wolf-enhanced CLPSO based on the

elite-based dominance scheme. Wang et al. [42] presented a heterogeneous CLPSO with a mutation operator and dynamic multi-swarm. Zhang et al. [43] presented a novel CLPSO using the Bayesian iteration method. Zhou et al. [44] presented an adaptive hierarchical update CLPSO based on the strategies of weighted synthesis. Tao et al. [45] presented an enhanced CLPSO with dynamic multi-swarm.

Sources	Results and Contribution to PSO
Maltra et al. [31]	Developed a hybrid cooperative CLPSO
Ali and Khan [33]	Developed an attributed multi-objective CLPSO
Hu et al. [34]	Presented a CLPSO with local search
Zhong et al. [35]	Presented a discrete CLPSO
Lin and Sun [36]	Designed an adaptive mutation for multi-leader CLPSO
Lin et al. [38]	Designed an adaptive mechanism for CLPSO
Cao et al. [40]	Developed a CLPSO with local search
Chen et al. [41]	Developed a grey-wolf-enhanced CLPSO
Wang et al. [42]	Developed a heterogeneous CLPSO
Zhou et al. [44]	Developed an adaptive hierarchical update CLPSO
Tao et al. [45]	Developed an enhanced CLPSO with dynamic multi-swarm

These improved CLPSO algorithms use the individual optimal information of particles to guide the whole iterative process, have better diversity and search range, and can solve complex multimodal problems. However, because the global optimal value does not participate in the particle velocity and position, the particle velocity is too small in the later search, and the convergence speed is slow. At the same time, due to the lack of measures for avoiding the local optimization, once the optimal values of most particles fall into the local optimization, the convergence is unable to find the global optimal value, and the performance is unstable. Therefore, to improve the optimization performance of CLPSO, a novel multi-strategy adaptive CLPSO (MSACLPSO) based on making use of comprehensive learning, multi-population parallel, and parameter adaptation was designed for this paper. The MSACLPSO effectively promotes information exchange in different dimensions, ensures information sharing in the population, enhances the convergence and stability, and balances the search ability compared with the other related algorithms.

The main contributions and novelties of this paper are described as follows.

- (1) A novel multi-strategy adaptive CLPSO (MSACLPSO) based on comprehensive learning, multi-population parallel, and parameter adaptation is presented.
- (2) A multi-population parallel strategy is designed to improve population diversity and accelerate convergence.
- (3) A new velocity update strategy is designed by adding the global optimal value in the population to the velocity update.
- (4) A new adaptive adjustment strategy of learning factors is developed by linearly changing the learning factors.
- (5) A parameter optimization method of photovoltaics is designed to prove the actual application ability.

2. PSO

PSO is a population-based search algorithm that simulates the social behavior of birds within a range. In PSO, all individuals are referred to as particles, which are flown through the search space to delete the success of other individuals. The position of particles changes according to the individual's social and psychological tendencies. The change of one particle is influenced by knowledge or experience. As a modeling result of the social

behavior, the search is processed to return to previously successful areas in the search space. The particle's velocity (v) and position (x) are changed by the particle best value ($pBest$) and global best value ($gBest$). The formula for updating velocity and position is given as follows:

$$v_{ij}^{t+1} = \omega v_{ij}^t + c_1 r_1 (pBest_{ij}^t - x_{ij}^t) + c_2 r_2 (gBest_{ij}^t - x_{ij}^t) \quad (1)$$

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

where v_{ij}^{t+1} is the velocity of the i^{th} particle at the j^{th} iteration, x_{ij}^{t+1} is the position of particle i^{th} at the j^{th} iteration, and the position of the particle is related to its velocity. w is an inertia weight factor, which is used to reflect the motion habits of particles and represent the tendency of particles to maintain their previous speed. c_1 is a self-cognition factor, which reflects the particle's memory of its own historical experience, and represents that the particle has a tendency to approach its best position. c_2 is a social cognition factor, which reflects the population's historical experience of collaboration and knowledge sharing among particles, and represents that particles tend to approach the best position in the population or field history. r_1 and r_2 represent random numbers in $[0, 1]$, which denote the remembrance ability for the research. Generally, the value in the V can be clamped to the range $[-V_{max}, V_{max}]$ in order to control the excessive roaming of particles outside the search space. The PSO is terminated until the maximal number of iterations is reached or the best particle position cannot be further improved. The PSO achieves better robustness and effectiveness in solving optimization problems.

The basic flow of the PSO is shown in Figure 1.

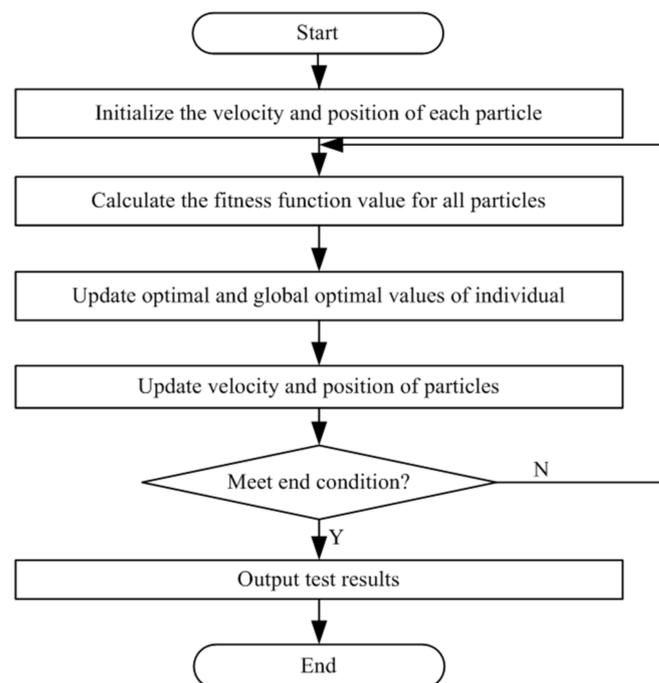


Figure 1. The basic flow of the PSO.

3. CLPSO

PSO can easily fall into local extrema, which leads to premature convergence. Thus, a new update strategy is presented to develop a CLPSO algorithm. In the PSO, each particle learns from its own optimal value and the global optimal value. Therefore, in the velocity update formula of CLPSO, the social part of the global optimal solution of particle learning is not used. In addition, in the velocity update formula of the traditional PSO algorithm, each particle learns from all dimensions of its own optimal value, but its own optimal value is not optimal in all dimensions. Therefore, the CLPSO algorithm introduces

a comprehensive learning strategy to construct learning samples using the $pBest$ of all particles to promote the information exchange, improve population diversity, and avoid falling into local extrema. The comprehensive learning strategy is to use the individual historical optimal solution of all particles in the population to update the particle position in order to effectively enhance the exploration ability of the PSO and achieve excellent optimization performance in solving multimodal optimization problems. The velocity update of particle and position is described as follows:

$$\begin{cases} v_{ij}^{t+1} = \omega v_{ij}^t + cr_{ij}^t (pBest_{f_i(j)}^t - x_{ij}^t) \\ v_{ij}^{t+1} = \min(v_{ijmax}, \max(v_{ijmin}, v_{ij}^{t+1})) \\ x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \\ x_{ij}^{t+1} = \min(x_{ijmax}, \max(x_{ijmin}, x_{ij}^{t+1})) \end{cases} \quad (3)$$

where $i = 1, 2, 3, \dots, P$ and $j = 1, 2, 3, \dots, D$. P is the size of the population and D is the search space dimension. $x_i^t = [x_{i1}^t, x_{i2}^t, \dots, x_{ij}^t, \dots, x_{iD}^t]$ is the particle position, $v_i^t = [v_{i1}^t, v_{i2}^t, \dots, v_{ij}^t, \dots, v_{iD}^t]$ is the velocity of particle i , $[x_{ijmin}, x_{ijmax}]$ is the search range of particle i , $[v_{ijmin}, v_{ijmax}]$ is the velocity range, ω is the inertia weight, c is the learning factor, r_{ij}^t is a randomly distributed number on $(0, 1)$, $f_i(j)$ refers to other particles that particle i needs to learn in the D -dimension, and $pbest_{f_i(j)}^t$ can be the optimal position of any particle.

The determination method of $f_i(j)$ is described as follows: For each particle dimension, a random probability is produced. If the random probability is greater than the learning probability P_{c_i} , then this particle dimension learns from the corresponding dimension of its own individual optimal value. On the other hand, two particles are randomly selected to learn the better optimal value. To ensure the population's polymorphism, the CLPSO also sets an update interval number m ; that is, when the individual optimal value of particle i has not been updated for m iterations, it is regenerated.

4. MSACLPSO

PSO has simplicity, practicality, and fixed parameters, but it has the disadvantage of easily falling into local optima, as well as weak local search ability. The CLPSO has slow velocity in the later search, low convergence speed, and unstable performance. To solve these problems, a multi-strategy adaptive CLPSO (MSACLPSO) algorithm is proposed by introducing a comprehensive learning strategy, multi-population parallel strategy, velocity update strategy, and parameter adaptive strategy. In MSACLPSO, a comprehensive learning strategy is introduced to construct learning samples using the $pBest$ of all particles to promote information exchange, improve population diversity, and avoid falling into local extrema. To overcome the lack of local search ability in the later stage, the global optimal value of the population is used for the velocity update, and a new update strategy is proposed to enhance the local search ability. The multi-population parallel strategy is employed to divide the population into N subpopulations, and then iterative evolution is carried out appropriately to achieve particle exchange and mutation, enhance the population diversity, accelerate the convergence, and ensure information sharing between the particles. The linearly changing strategy of the learning factors is employed to realize the iterative evolution in different stages and the adaptive adjustment strategy of learning factors, which can enhance the global search ability and improve the local search ability. The S -shaped decreasing function is adopted to realize the adaptive adjustment of inertia weight to ensure that the population has high speed in the initial stage, reduce the search speed in the middle stage—so that the particles will more easily converge to the global optimum—and maintain a certain speed for the final convergence in the later stage.

4.1. Multi-Population Parallel Strategy

The idea of multi-population parallel is based on the natural phenomenon of the evolution of the same species in different regions. It divides the population into multiple subpopulations, and then each subpopulation searches for the optimal value in parallel to improve the search ability. The indirect exchange of the optimal value and dynamic recombination of the population can enhance the population diversity and accelerate the convergence. A multi-population parallel strategy is proposed here. The main ideas of the multi-population parallel strategy are described as follows: The population is divided into N subpopulations in the process of evolution. For each subpopulation, the particle carries out iterative evolution, and the particle exchange and particle mutation under appropriate conditions are executed according to certain rules, so as to ensure information sharing between the particles of the population through the exchange of particles between subpopulations. Therefore, to enhance the local search ability of the CLSPO algorithm in the later stage, a new update strategy is presented after the g_0 generation is completed. That is, the global optimal value $gBest$ of the population is added to the velocity update, as shown in Equation (4):

$$\begin{cases} v_{ij}^{t+1} = \omega v_{ij}^t + c_1 r_{1ij}^t (pBest_{f_i(j)}^t - x_{ij}^t) + c_2 r_{2ij}^t (gBest_{f_i(j)}^t - x_{ij}^t) \\ v_{ij}^{t+1} = \min(v_{ijmax}, \max(v_{ijmin}^t, v_{ij}^{t+1})) \\ x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \\ x_{ij}^{t+1} = \min(x_{ijmax}, \max(x_{ijmin}^t, x_{ij}^{t+1})) \end{cases} \quad (4)$$

where c_1 and c_2 are learning factors, $pBest_{f_i(j)}^t$ is the optimal value of the particle in each subpopulation $\{pBest_1^t, pBest_2^t, \dots, pBest_p^t, \dots, pBest_p^t\}$, $gBest_{f_i(j)}^t$ is the optimal value of each subpopulation $\{gBest_1^t, gpBest_2^t, \dots, gBest_p^t, \dots, gBest_p^t\}$, r_{1ij}^t and r_{2ij}^t are randomly distributed numbers on $(0, 1)$.

4.2. Adaptive Learning Factor Strategy

In PSO, the values of c_1 and c_2 are set in advance according to experiences, reducing the self-learning ability. Therefore, the linearly changing strategy of the learning factors is developed for c_1 and c_2 . In the early evolution stage, the self-cognition item is reduced and the social cognition item is increased to improve the global search ability. In the later evolution stage, the local search ability is guaranteed by encouraging particles to converge towards the global optimum. Therefore, the adaptive learning factor strategy is described as follows:

$$c_1 = c_{min} + (c_{max} - c_{min})(T - t)/T \quad (5)$$

$$c_2 = c_{min} + (c_{max} - c_{min})t/T \quad (6)$$

where c_{max} and c_{min} are the maximum value and minimum value, respectively.

4.3. Adaptive Inertia Weight Strategy

In PSO, when the particles in the population tend to be the same, the last two terms in the particle velocity update formula—namely, the social cognition part, and the individual's own cognition part—will gradually tend towards 0. If the inertia weight ω is less than 1, the particle speed will gradually decrease, or even stop moving, which result in premature convergence. When the optimal fitness of the population has not changed (i.e., has stagnated) for a long time, the inertia weight ω should be adjusted adaptively according to the degree of premature convergence. If the same adaptive operation is adopted for the population, when the population has converged to the global optimum, the probability of destroyed excellent particles will increase with the increase in their inertia weight, which will degrade the performance of the PSO algorithm. To better balance the search ability, an S-shaped decreasing function is adopted to ensure that the population has high speed in

the initial stage, and the search speed decreases in the middle stage, so that the particles can easily converge to the global optimum value and, finally, converge at a certain speed in the later stage. The S-shaped decreasing function for the inertia weight ω is described as follows:

$$\omega = (\omega_{max} - \omega_{min}) / (1 + \exp(2 * a * t / T - a)) + \omega_{min} \tag{7}$$

where ω_{max} and ω_{min} are the maximum and minimum values, respectively— $\omega_{max} = 0.9$ and $\omega_{min} = 0.2$ —and a is the control coefficient to adjust the speed change, where $a = 13$.

4.4. Model of MSACLPSO

The flow of MSACLPSO is shown in Figure 2.

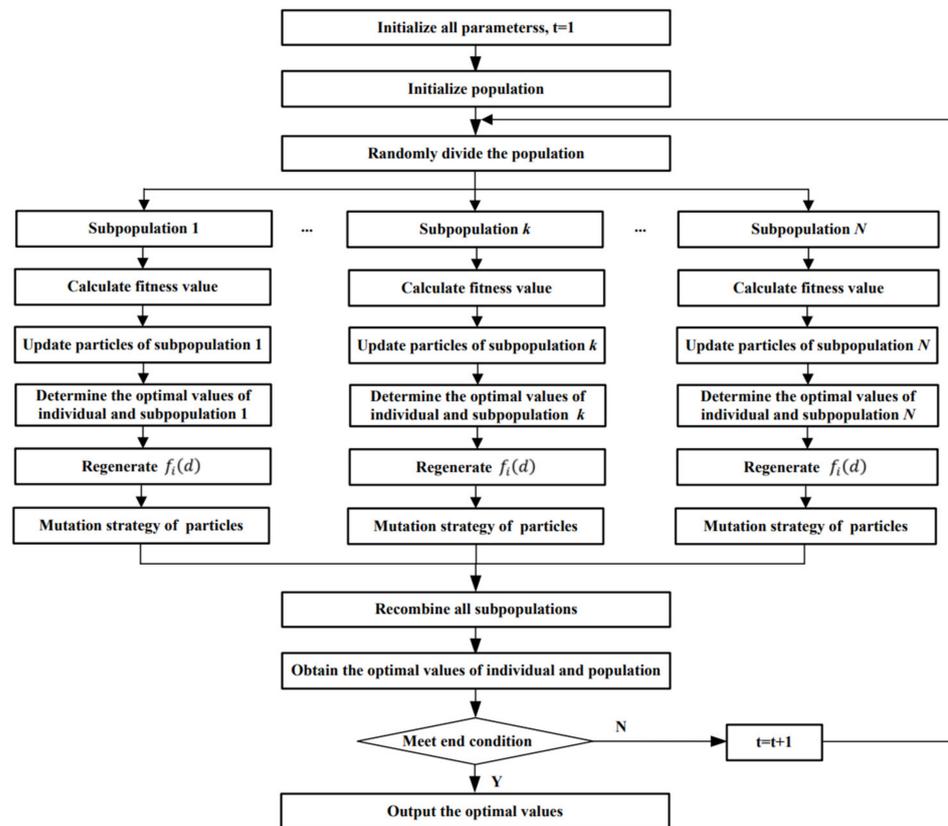


Figure 2. The flow of MSACLPSO.

The steps of MSACLPSO are described as follows:

Step 1: Divide the population into N subpopulations, and initialize all parameters.

Step 2: Execute the CLPSO algorithm for each subpopulation. The objective function is used to find out the individual optimal value of the particle, the optimal value of the subpopulation, and the global optimal value of the population. To ensure the high global search ability in the early stage, T_0 is set for the early stage, and each subpopulation updates all particle states according to Equation (3). To enhance the local search ability of CLSPO in the later stage, after the T_0 iteration is completed, each subpopulation updates all particle states according to Equation (4).

Step 3: If the optimal value of one subpopulation does not update for successive $R1$ iterations, the population may fall into local optimization. To avoid falling into the local optimum for the subpopulation, the mutation strategy is used here. Each dimension of each particle in the subpopulation is mutated with the probability P_m . The mutation mode is described as follows:

$$x_{id}^t = x_{id}^t + randn(x_{idmax} - x_{idmin})(T - t) / T \tag{8}$$

where $randn$ is the random number on $(-1, 1)$.

Step 4: After T_0 iterations are executed, to enhance population diversity, the particles are randomly exchanged between populations every interval R iteration to recombine subpopulations. The recombination of subpopulations is described as follows: All subpopulations randomly select 50% of the particles, which are randomly exchanged with the particles of other populations. Then, according to the fitness values of all particles in all subpopulations, $1/N$ particles with the best fitness values in each subpopulation are selected to construct a new population. It is worth noting that the exchanged particle can be any particle in any other population.

Step 5: Determine whether the end conditions are met. If they are met, the optimal result is output; otherwise, return to Step 2.

5. Experiment Simulation and Analysis

5.1. Test Functions

To verify the performance of MSACLPSO, 10 famous benchmark functions were selected. The detailed description is shown in Table 1.

Table 1. The detailed description.

Function Name	Function Expression	S	F_{min}	f_bias
Sphere	$F_1 = \sum_{i=1}^D x_i$	$[-100, 100]^D$	0	-450
Schwefel 1.2	$F_2 = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^D$	0	-450
High Conditioned Elliptic	$F_3 = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2$	$[-100, 100]^D$	0	-450
Schwefel 1.2 with Noise	$F_4 = \left(\sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2 \right) * (1 + 0.4 N(0,1))$	$[-100, 100]^D$	0	-450
Schwefel 2.6	$F_5 = \max\{ x_1 + 2x_2 - 7 , 2x_1 + x_2 - 5 \}$	$[-100, 100]^D$	0	-310
Rosenbrock	$F_6 = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	$[-100, 100]^D$	0	390
Griewank	$F_7 = \sum_{i=1}^{D-1} \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-100, 100]^D$	0	390
Ackley	$F_8 = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20$	$[-32, 32]^D$	0	-140
Rastrigin	$F_9 = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5, 5]^D$	0	-330
Expanded Schaffer	$F_{10} = 0.5 + \frac{(\sin^2(\sqrt{x^2+y^2}) - 0.5)}{(1+0.001(x^2+y^2))^2}$	$[-100, 100]^D$	0	-300

5.2. Experimental Environment and Parameter Setting

The experimental environment mainly included Core I5-4200H, Win10, RAM-16GB, and MATLAB R2018b. The optimization performance of MSACLPSO was compared with other state-of-the-art algorithms, including the basic version of PSO (PSO) [46], self-organizing hierarchical PSO (HPSO) [47], fully-informed PSO (FIPS) [48], unified PSO (UPSO) [49], CLPSO [30], and static heterogeneous particle swarm optimization (sH-PSO) [50]. In MSACLPSO, the population is divided into two subpopulations, and four main parameters are adjusted to balance exploration and exploitation. These parameters

include population size, acceleration coefficients, iteration number, and dimensions. In our experiment, a large number of alternative values were tested, and some classical values were selected from other literature, and then these parameter values were experimentally modified until the most reasonable parameter values were selected. These selected parameter values attained the optimal solution, so that they could accurately and efficiently verify the effectiveness of MSACLPSO in solving optimization problems. Some parameters that were tuned included the population size $N_P = 40$, the number of subpopulations $N = 2$, $c_{min} = 0.5$ and $c_{max} = 2.5$, the dimension $D = 30$, run times $T = 30$, the maximum number of iterations $G = 200$, and function evaluations FEs = 300,000. The specific settings are shown in Table 2.

Table 2. The parameter settings.

Algorithms	ω	c	c_1	c_2	P_{c_i}	N_P	FES
PSO	0.9~0.4	—	2.0	2.0	—	60	300,000
HPSO	—	—	2.5~0.5	0.5~2.5	—	40	300,000
FIPS	—	2	—	—	—	40	300,000
UPSO	—	1.49445	—	—	—	40	300,000
OLPSO	0.9~0.4	2	—	—	—	40	300,000
CLPSO	0.9~0.4	1.49445	—	—	0.5	40	300,000
sHPSO	0.72	—	2.5~0.5	0.5~2.5	—	40	300,000
MSACLPSO	0.95~0.3	3.0~1.5	2.5~0.5	0.5~2.5	0.5	40	300,000

5.3. Experimental Results and Analysis

The population was divided into two subpopulations, and different numbers of individuals were set. The error mean (mean) value and standard deviation (Std) value were applied to evaluate the optimization performance of MSACLPSO. The obtained experimental results with the different numbers of individuals for 30-dimensional problems are shown in Table 3. The best results are the bold.

As can be seen from Table 3, the subpopulation size $P_1 = 10$ and $P_2 = 30$ obtained the best optimization performance for the 10 test benchmark functions compared with other subpopulation sizes. However, for the functions F_5 , F_6 , and F_9 , MSACLPSO did not obtain satisfactory optimization performance. Therefore, the subpopulation size $P_1 = 10$ and $P_2 = 30$ was selected for performance evaluation of MSACLPSO.

MSACLPSO was compared with some variants of PSO algorithms. The optimization performance was obtained according to the mean and Std of the 20 obtained results. The obtained experimental results using different algorithms for test functions with 30 dimensions are shown in Table 4. The obtained best results are highlighted in **bold**.

Table 3. The different numbers of individuals (P_1 and P_2) in two subpopulations for MSACLPSO.

Functions	Indices	10 + 30	15 + 25	20 + 20	25 + 15	30 + 10	40 + 0
F_1	Mean	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Std	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
F_2	Mean	1.9862×10^{-8}	1.6547×10^{-6}	2.2056×10^{-4}	1.6547×10^{-2}	3.3274×10^{-1}	3.7543×10^{-1}
	Std	3.5974×10^{-8}	1.7430×10^{-6}	3.3469×10^{-4}	1.3275×10^{-2}	4.5401×10^{-1}	4.7341×10^{-1}
F_3	Mean	5.5673×10^{-5}	6.1432×10^5	8.4102×10^5	1.3610×10^6	2.1977×10^6	2.3560×10^6
	Std	1.7703×10^{-5}	2.4205×10^5	3.2359×10^5	6.6034×10^5	9.2605×10^5	6.6496×10^5
F_4	Mean	4.2485×10^2	5.0328×10^2	6.0462×10^2	8.1743×10^2	1.4058×10^3	1.4135×10^3
	Std	2.4452×10^2	2.8874×10^2	4.4718×10^2	4.0569×10^2	7.1673×10^2	6.4532×10^2
F_5	Mean	2.7830×10^3	2.5065×10^3	2.8913×10^3	3.1673×10^3	3.2137×10^3	3.5478×10^3
	Std	5.5702×10^2	4.1407×10^2	4.0531×10^2	5.9613×10^2	4.2715×10^2	5.5379×10^2
F_6	Mean	3.1637	2.2637	2.1975	7.1762×10^{-1}	2.9757×10^{-1}	3.3405×10^{-1}
	Std	3.3643	4.0623	3.4537	9.2546×10^{-1}	5.2504×10^{-1}	6.6492×10^{-1}
F_7	Mean	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Std	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
F_8	Mean	1.9745×10^1	1.9805×10^1	1.9832×10^1	1.9867×10^1	1.9835×10^1	2.0645×10^1
	Std	8.3746×10^{-2}	7.1485×10^{-2}	5.8043×10^{-2}	5.7903×10^{-2}	8.8562×10^{-2}	7.8530×10^{-2}
F_9	Mean	1.0673	1.0245×10^{-1}	0.0000	0.0000	0.0000	1.2473
	Std	1.0305	3.8672×10^{-1}	0.0000	0.0000	0.0000	1.2865
F_{10}	Mean	1.0782×10^1	1.0954×10^1	1.1065×10^1	1.1438×10^1	1.1714×10^1	1.1904×10^1
	Std	4.0645×10^{-1}	5.4680×10^{-1}	4.3591×10^{-1}	5.4613×10^{-1}	4.1527×10^{-1}	4.3681×10^{-1}

Table 4. The obtained experimental results using different algorithms.

Functions	Indices	PSO	HPSO	FIPS	OLPSO	UPSO	sHPSO	CLPSO	HCLPSO	MSACLPSO
F_1	Mean	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
F_2	Mean	3.70×10^{-1}	3.79×10^{-6}	7.79×10^1	1.38×10^1	2.65E-07	1.44×10^{-2}	1.14×10^3	1.70×10^{-6}	1.99×10^{-8}
	Std	3.20×10^{-1}	2.82×10^{-6}	2.71×10^1	8.33	2.42E-07	7.10×10^{-2}	2.53×10^2	1.71×10^{-6}	3.60×10^{-8}
F_3	Mean	6.53×10^6	7.72×10^5	2.45×10^7	1.60×10^7	1.54×10^6	8.75×10^5	1.22×10^7	6.42×10^5	5.57×10^5
	Std	4.17×10^6	2.96×10^5	6.29×10^6	7.04×10^6	4.75×10^5	5.34×10^5	3.34×10^6	2.61×10^5	1.77×10^5
F_4	Mean	3.81×10^2	2.48×10^4	1.15×10^3	2.18×10^3	7.28×10^3	2.02×10^4	8.77×10^3	5.22×10^2	4.25×10^2
	Std	3.31×10^2	5.71×10^3	3.73×10^2	1.09×10^3	2.79×10^3	9.94×10^3	1.85×10^3	3.09×10^2	2.45×10^2
F_5	Mean	3.85×10^3	9.20×10^3	2.22×10^3	3.30×10^3	6.32×10^3	6.94×10^3	4.47×10^3	2.97×10^3	2.78×10^3
	Std	8.00×10^2	1.81×10^3	5.14×10^2	3.75×10^2	1.63×10^3	1.43×10^3	4.26×10^2	4.55×10^2	5.57×10^2
F_6	Mean	7.02×10^1	5.04×10^1	3.77×10^1	2.07×10^1	6.82×10^1	1.15×10^2	2.39	2.39	3.16
	Std	9.51×10^1	5.05×10^1	3.50×10^1	2.50×10^1	9.64×10^1	2.29×10^2	3.84	4.27	3.36
F_7	Mean	7.60×10^{-1}	1.00×10^{-2}	3.00×10^{-2}	1.00×10^{-2}	2.00×10^{-2}	4.00×10^2	7.00×10^{-1}	2.00×10^{-2}	0.00
	Std	1.41	1.00×10^{-2}	2.00×10^{-2}	1.00×10^{-2}	1.00×10^{-2}	4.00×10^2	1.50×10^{-1}	2.00×10^{-2}	0.00
F_8	Mean	2.09×10^1	2.07×10^1	2.09×10^1	2.10×10^1	2.10×10^1	2.02×10^1	2.10×10^1	2.09×10^1	1.97×10^1
	Std	7.00×10^{-2}	1.50×10^{-1}	6.00×10^{-2}	8.00×10^{-2}	5.00×10^{-2}	1.90×10^{-1}	6.00×10^{-2}	9.00×10^{-2}	8.37×10^{-2}
F_9	Mean	1.90×10^1	1.07×10^1	5.71×10^1	0.00	8.52×10^1	8.25×10^1	1.00×10^2	0.00	1.07
	Std	5.37	4.96	1.46×10^1	0.00	1.69×10^1	2.44×10^1	1.25×10^1	0.00	1.03
F_{10}	Mean	1.27×10^1	1.23×10^1	1.31×10^1	1.31×10^1	1.28×10^1	1.31×10^1	1.26×10^1	1.19×10^1	1.09×10^1
	Std	4.30×10^{-1}	3.70×10^{-1}	2.10×10^{-1}	2.00×10^{-1}	3.30×10^{-1}	3.90×10^{-1}	2.20×10^{-1}	5.80×10^{-1}	4.06×10^{-1}

As shown in Table 4, all algorithms performed equally on test function F_1 , and PSO obtained the best solution on test function F_4 . FIPS obtained the best solution on test function F_5 . MSACLPSO performed well on test functions $F_1 \sim F_5$. For multimodal functions, MSACLPSO performed well on all functions, and obtained the best performance on test functions F_7 , F_8 , and F_{10} , and the second-best performance on test functions F_6 and F_9 . On the other hand, CLPSO and HCLPSO obtained the best solution on test function F_6 , and OLPSO and HCLPSO obtained the best solution on test function F_9 . Overall, MSACLPSO obtained the best performance on 6 out of 10 test functions. Therefore, MSACLPSO performs well, and obtains the best optimization performance for multimodal problems. In our experiment, MSACLPSO used several strategies of comprehensive learning, multi-population parallel, and parameter adaptation. Although the strategies of comprehensive learning and parameter adaptation need more running time, the multi-population parallel strategy can reduce the running time. Therefore, the time complexity of MSACLPSO is similar to that of the other compared algorithms.

To test the statistical difference between MSACLPSO and the other variants of PSO algorithms, the non-parametric Wilcoxon signed-rank test was used to compare the results of MSACLPSO and the results of the other variants of PSO. The obtained results of MSACLPSO against other algorithms are shown in Table 5.

Table 5. The test results under $\alpha = 0.05$.

Functions	PSO	HPSO	FIPS	OLPSO	UPSO	sHPSO	CLPSO	HCLPSO
F_1	=	=	=	=	=	=	=	=
F_2	+	+	+	+	+	+	+	+
F_3	+	+	+	+	+	+	+	+
F_4	−	+	+	+	+	+	+	+
F_5	+	+	−	+	+	+	+	+
F_6	+	+	+	+	+	+	−	−
F_7	+	+	+	+	+	+	+	+
F_8	+	+	+	+	+	+	+	+
F_9	+	+	+	−	+	+	+	−
F_{10}	+	+	+	+	+	+	+	+
+ / = / −	8/1/1	9/1/0	8/1/1	8/1/1	9/1/0	9/1/0	8/1/1	7/1/2

As shown in Table 5, MSACLPSO performs better than the other variants of PSO algorithms through the number of (+ / = / −) in the last row of the Wilcoxon signed-rank test results under $\alpha = 0.05$.

To sum up, it can be seen that the optimized values of parameters for MSACLPSO are $\omega = 0.43$, $c = 2.1$, $c_1 = 1.8$, $c_2 = 2.1$, and $P_{c_i} = 0.5$ for solving these complex optimization problems.

6. Case Analysis

Renewable energy has always been the focus of dealing with the key issues of traditional energy consumption, which uses nonrenewable energy. Solar energy is an up-and-coming resource, in which PV plays a vital role. However, the PV device is usually placed in an exposed environment, which leads to its degradation. This seriously affects the efficiency of PV. Therefore, MSACLPSO was employed to effectively and accurately optimize the PV parameters to establish an optimized PV model. The values of parameters for MSACLPSO were the same as given in Section 5.3.

6.1. Modeling for PV

A lot of PV models have been designed, and were applied to illustrate the I–V characteristics. The SDM and DDM are the most widely used [51]. The PV model is described in Table 6.

Table 6. The modelling for PV.

PV	I_L
SDM	$I_L = I_{ph} - I_{sd} \left[\exp\left(\frac{q(V_L + I_L R_s)}{nkT}\right) - 1 \right] - \frac{V_L + I_L R_s}{R_{sh}}$
DDM	$I_L = I_{ph} - I_{sd1} \left[\exp\left(\frac{q(V_L + I_L R_s)}{n_1 kT}\right) - 1 \right] - I_{sd2} \left[\exp\left(\frac{q(V_L + I_L R_s)}{n_2 kT}\right) - 1 \right] - \frac{V_L + I_L R_s}{R_{sh}}$

It is crucial to search for the optimal parameter values in order to minimize the error of the PV models. The error functions are described as follows: For the SDM,

$$\begin{cases} f(V_L, I_L, x) = I_{ph} - I_{sd} \left[\exp\left(\frac{q(V_L + I_L R_s)}{nkT}\right) - 1 \right] - \frac{V_L + I_L R_s}{R_{sh}} - I_L \\ x = \{I_{ph}, I_{sd}, R_s, R_{sh}, n\} \end{cases} \tag{9}$$

For the DDM,

$$\begin{cases} f(V_L, I_L, x) = I_{ph} - I_{sd1} \left[\exp\left(\frac{q(V_L + I_L R_s)}{n_1 kT}\right) - 1 \right] - I_{sd2} \left[\exp\left(\frac{q(V_L + I_L R_s)}{n_2 kT}\right) - 1 \right] - \frac{V_L + I_L R_s}{R_{sh}} - I_L \\ x = \{I_{ph}, I_{sd1}, I_{sd2}, R_s, R_{sh}, n_1, n_2\} \end{cases} \tag{10}$$

To evaluate the PV model, the RMSE is described as follows:

$$RMSE(x) = \sqrt{\frac{1}{N} \sum_{k=1}^N f(V_L, I_L, x)^2} \tag{11}$$

6.2. Modeling for PV

To validate the performance of MSACLPSO, the PSO, BLPSO, CLPSO, CPMPPO, IJAYA, GOTLBO, SATLBO, DE/BBO, DBBO, STLBO, WOA, CWOA, LWOA, GWO, EGWO, WDO, DE, JADE, and MPPCEDE [52–64] algorithms were used for comparison. The parameter values of MSACLPSO were the same as given in Section 5.2. The parameter values of the other compared algorithms were the same as in the literature. The maximum number of iterations was $G = 200$, and these algorithms were executed for 20 runs. Therefore, the statistical results of the SRE, LRE, MRE, and Std were obtained. The value of RMSE was employed to quantify the solution accuracy, while the Std of the RMSE described the reliability. The statistical results of the experiment with the SDM and DDM are shown in Tables 7 and 8, respectively. The obtained best results are highlighted in **bold**.

As can be seen from Table 7, CPMPPO, MPPCEDE, and MSACLPSO obtained the SRE, LRE, and MRE values. For the Std of RMSE, MSACLPSO performed well. Therefore, the optimization performance of MSACLPSO was better than that of the compared algorithms for SDM. As can be seen from Table 8, MSACLPSO obtained the best results for the SRE, LRE, MRE, and Std of RMSE. For the Std of RMSE, MSACLPSO obtained the best Std. Therefore, MSACLPSO is the best algorithm for DDM.

To sum up, it can be seen that the performance of MSACLPSO was demonstrated by optimizing the PV model parameters. All of the compared results containing the optimized parameters, along with the SRE, LRE, MRE, and Std values, show that MSACLPSO can obtain the optimal parameters. This provides a more effective parameter combination for the complex engineering problems of photovoltaics, so as to improve the energy conversion efficiency.

Table 7. The obtained results of RMSE for the SDM.

Algorithms	SRE	LRE	MRE	Std	Symbol
PSO	2.44805×10^{-3}	9.86022×10^{-4}	1.31844×10^{-3}	5.24500×10^{-4}	+
BLPSO	1.74592×10^{-3}	1.03122×10^{-3}	1.31377×10^{-3}	1.90400×10^{-4}	+
CLPSO	1.25274×10^{-3}	9.92075×10^{-4}	1.06081×10^{-3}	7.04200×10^{-5}	+
CPMPSO	9.86022×10^{-4}	9.86022×10^{-4}	9.86022×10^{-4}	2.17556×10^{-17}	+
IJAYA	9.86841×10^{-4}	9.86022×10^{-4}	9.86051×10^{-4}	1.49300×10^{-7}	+
GOTLBO	1.39559×10^{-3}	9.86608×10^{-4}	1.08300×10^{-3}	9.70900×10^{-5}	+
SATLBO	1.00674×10^{-3}	9.86025×10^{-4}	9.88799×10^{-4}	4.81300×10^{-6}	+
DE/BBO	1.84123×10^{-3}	9.86022×10^{-4}	1.25173×10^{-3}	2.08225×10^{-4}	+
DBBO	2.36083×10^{-3}	9.86820×10^{-4}	1.38755×10^{-3}	2.70008×10^{-4}	+
STLBO	1.02033×10^{-3}	9.86022×10^{-4}	9.87207×10^{-4}	6.25700×10^{-6}	+
WOA	1.00397×10^{-2}	1.10759×10^{-3}	3.25587×10^{-3}	2.16463×10^{-3}	+
CWOA	3.28588×10^{-2}	9.98677×10^{-4}	5.44921×10^{-3}	6.33831×10^{-3}	+
LWOA	1.92042×10^{-2}	9.99621×10^{-4}	3.44545×10^{-3}	3.33774×10^{-3}	+
GWO	4.43070×10^{-2}	1.28030×10^{-3}	1.13440×10^{-2}	1.48470×10^{-2}	+
EGWO	5.24900×10^{-3}	2.11210×10^{-3}	3.50150×10^{-3}	1.59880×10^{-3}	+
WDO	4.42600×10^{-3}	1.22101×10^{-3}	2.18020×10^{-3}	7.63880×10^{-4}	+
DE	1.81059×10^{-3}	9.86022×10^{-4}	1.02116×10^{-3}	1.44688×10^{-4}	+
JADE	1.41030×10^{-3}	9.86060×10^{-4}	1.08330×10^{-3}	1.09000×10^{-4}	+
MPPCEDE	9.86022×10^{-4}	9.86022×10^{-4}	9.86022×10^{-4}	0.00000	+
MSACLPSO	9.86022×10^{-4}	9.864574×10^{-4}	9.83758×10^{-4}	7.52967×10^{-18}	+

Table 8. The obtained results of the RMSE for the DDM.

Algorithms	SRE	LRE	MRE	Std	Symbol
PSO	4.34952×10^{-2}	9.82485×10^{-4}	4.37645×10^{-3}	1.01270×10^{-2}	+
BLPSO	1.93654×10^{-3}	1.08218×10^{-3}	1.53462×10^{-3}	2.45890×10^{-4}	+
CLPSO	1.38835×10^{-3}	9.94316×10^{-4}	1.13959×10^{-3}	9.39950×10^{-5}	+
CPMPSO	9.86022×10^{-4}	9.82485×10^{-4}	9.83137×10^{-4}	1.33980×10^{-6}	+
IJAYA	9.99410×10^{-4}	9.82494×10^{-4}	9.86860×10^{-4}	3.22120×10^{-6}	+
GOTLBO	1.53359×10^{-3}	9.85097×10^{-4}	1.16335×10^{-3}	1.51770×10^{-4}	+
SATLBO	1.23062×10^{-3}	9.82824×10^{-4}	1.00544×10^{-3}	5.02710×10^{-5}	+
DE/BBO	1.63508×10^{-3}	9.87990×10^{-4}	1.19281×10^{-3}	2.03849×10^{-4}	+
DBBO	9.84995×10^{-4}	2.29052×10^{-3}	1.22395×10^{-3}	3.08780×10^{-4}	+
STLBO	1.52433×10^{-3}	9.82561×10^{-4}	1.03435×10^{-3}	1.41980×10^{-4}	+
WOA	1.15633×10^{-3}	1.16011×10^{-2}	3.42961×10^{-3}	2.23226×10^{-3}	+
CWOA	8.86567×10^{-3}	1.13004×10^{-3}	3.50587×10^{-3}	2.15341×10^{-3}	+
LWOA	1.04935×10^{-3}	1.11900×10^{-2}	3.12337×10^{-3}	1.81559×10^{-3}	+
GWO	4.07970×10^{-2}	1.02742×10^{-3}	9.90850×10^{-4}	1.29040×10^{-2}	+
EGWO	5.00690×10^{-3}	1.80620×10^{-3}	3.06700×10^{-3}	1.70500×10^{-3}	+
WDO	4.93450×10^{-3}	1.68120×10^{-3}	3.29180×10^{-3}	8.41370×10^{-4}	+
DE	2.00941×10^{-3}	9.82936×10^{-4}	1.06862×10^{-3}	2.23325×10^{-4}	+
JADE	2.23830×10^{-3}	9.83510×10^{-4}	1.46570×10^{-3}	3.81000×10^{-4}	+
MPPCEDE	9.82908×10^{-4}	9.82485×10^{-4}	9.82504×10^{-4}	8.02951×10^{-8}	+
MSACLPSO	9.82743×10^{-4}	9.82368×10^{-4}	9.81463×10^{-4}	9.68924×10^{-9}	+

7. Conclusions

In this paper, a multi-strategy adaptive CLPSO with comprehensive learning, multi-population parallel, and parameter adaptation is proposed. A multi-population parallel strategy was designed to improve population diversity and accelerate convergence. Then, a new velocity update strategy was designed for the velocity update, and a new adaptive adjustment strategy of learning factors was developed. Additionally, a parameter optimization method for photovoltaics was designed to prove the actual application ability. Ten benchmark functions were used to prove the effectiveness of MSACLPSO in comparison with different variants of PSO. On 6 out of 10 functions, MSACLPSO obtained the best

performance. MSACLPSO performed well and obtained the best optimization performance for multimodal problems. In addition, the actual SDM and DDM were selected for parameter optimization. The experimental results show that the actual application ability of the MSACLPSO was confirmed in comparison with the other algorithms. MSACLPSO is an alternative optimization technique for solving complex problems and actual engineering problems.

However, MSACLPSO is still insufficient in solving large-scale parameter optimization problems, such as time complexity and easy stagnation, among others. In the future, these applications should be considered [65–72]. The algorithm should be deeply studied, and the parameter adaptability of MSACLPSO in different stages and scales should also be further explored in future works.

Author Contributions: Conceptualization, Y.Z. and X.S.; methodology, Y.Z.; software, Y.Z.; validation, Y.Z. and X.S.; investigation, X.S.; resources, X.S.; data curation, Y.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, X.S.; visualization, X.S.; supervision, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Thank you very much for contribution of Deng.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

PSO	Particle swarm optimization
HPSO	Hierarchical PSO
FIPS	Fully-informed PSO
UPSO	Unified PSO
sHPSO	Static heterogeneous PSO
CLPSO	Comprehensive learning PSO
MSACLPSO	Multi-strategy adaptive CLPSO
ABC	Artificial bee colony
SMA	Slime mold algorithm
HHO	Harris hawk optimization
w	Inertia weight factor
c_1	Self-cognition factor
c_2	Social cognition factor
V_{\max}	Max velocity
RMSE	Root-mean-square error
SRE	Smallest RMSE
LRE	Largest RMSE
MRE	Mean RMSE
Std	Standard deviation
PV	Photovoltaics
SDM	Single-diode model
DDM	Double-diode model
I–V	Current–voltage
P–V	Power–voltage
r_1, r_2	Random numbers
v_{ij}^{t+1}	Velocity
P	Size of population
D	Search space dimension

References

1. Diep, Q.B.; Truong, T.C.; Das, S.; Zelinkada, I. Self-organizing migrating algorithm with narrowing search space strategy for robot path planning. *Appl. Soft Comput.* **2022**, *116*, 108270. [[CrossRef](#)]
2. Li, G.; Li, Y.; Chen, H.; Deng, W. Fractional-order controller for course-keeping of underactuated surface vessels based on frequency domain specification and improved particle swarm optimization algorithm. *Appl. Sci.* **2022**, *12*, 3139. [[CrossRef](#)]
3. Deng, W.; Xu, J.J.; Gao, X.Z.; Zhao, H.M. An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 1578–1587. [[CrossRef](#)]
4. Sarkar, B.; Biswas, A. Multicriteria decision making approach for strategy formulation using Pythagorean fuzzy. *Expert Syst.* **2022**, *39*, e12802. [[CrossRef](#)]
5. Gupta, A.; Datta, S.; Das, S. Fuzzy clustering to identify clusters at different levels of fuzziness: An evolutionary multiobjective optimization approach. *IEEE Trans. Cybern.* **2021**, *51*, 2601–2611. [[CrossRef](#)] [[PubMed](#)]
6. Li, X.; Zhao, H.; Yu, L.; Chen, H.; Deng, W.; Deng, W. Feature extraction using parameterized multi-synchrosqueezing transform. *IEEE Sens. J.* **2022**. [[CrossRef](#)]
7. Halim, A.H.; Ismail, I.; Das, S. Performance assessment of the metaheuristic optimization algorithms: An exhaustive review. *Artif. Intell. Rev.* **2021**, *54*, 2323–2409. [[CrossRef](#)]
8. An, Z.; Wang, X.; Li, B.; Xiang, Z.; Zhang, B. Robust visual tracking for UAVs with dynamic feature weight selection. *Appl. Intell.* **2022**. [[CrossRef](#)]
9. Deng, W.; Zhang, X.; Zhou, Y.; Liu, Y.; Zhou, X.; Chen, H.; Zhao, H. An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. *Inf. Sci.* **2022**, *585*, 441–453. [[CrossRef](#)]
10. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
11. Nickabadi, A.; Ebadzadeh, M.M.; Safabakhsh, R. A novel particle swarm optimization algorithm with adaptive inertia weight. *Appl. Soft Comput.* **2011**, *11*, 3658–3670. [[CrossRef](#)]
12. Wang, Y.; Li, B.; Weise, T.; Wang, J.; Yuan, B.; Tian, Q. Self-adaptive learning based particle swarm optimization. *Inf. Sci.* **2011**, *181*, 4515–4538. [[CrossRef](#)]
13. Zhan, Z.H.; Zhang, J.; Li, Y.; Shi, Y.H. Orthogonal learning particle swarm optimization. *IEEE Trans. Evol. Comput.* **2011**, *15*, 832–847. [[CrossRef](#)]
14. Li, X.D.; Yao, X. Cooperatively coevolving particle swarms for large scale optimization. *IEEE Trans. Evol. Comput.* **2012**, *16*, 210–224.
15. Xu, G. An adaptive parameter tuning of particle swarm optimization algorithm. *Appl. Math. Comput.* **2013**, *219*, 4560–4569. [[CrossRef](#)]
16. Wang, H.; Sun, H.; Li, C.H.; Rahnamayan, S.; Pan, J.S. Diversity enhanced particle swarm optimization with neighborhood search. *Inf. Sci.* **2013**, *223*, 119–135. [[CrossRef](#)]
17. Chen, W.N.; Zhang, J.; Lin, Y.N.; Zhan, Z.H.; Chung, H.S.H.; Shi, Y.H. Particle swarm optimization with an aging leader and challengers. *IEEE Trans. Evol. Comput.* **2013**, *17*, 241–258. [[CrossRef](#)]
18. Qu, B.Y.; Suganthan, P.N.; Das, S. A distance-based locally informed particle swarm model for multimodal optimization. *IEEE Trans. Evol. Comput.* **2013**, *17*, 387–402. [[CrossRef](#)]
19. Cheng, R.; Jin, Y.C. A social learning particle swarm optimization algorithm for scalable optimization. *Inf. Sci.* **2015**, *291*, 43–60. [[CrossRef](#)]
20. Tanweer, M.R.; Suresh, S.; Sundararajan, N. Self-regulating particle swarm optimization algorithm. *Inf. Sci.* **2015**, *294*, 182–202. [[CrossRef](#)]
21. Taherkhani, M.; Safabakhsh, R. A novel stability-based adaptive inertia weight for particle swarm optimization. *Appl. Soft Comput.* **2016**, *38*, 281–295. [[CrossRef](#)]
22. Moradi, P.; Gholampour, M. A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Appl. Soft Comput.* **2016**, *43*, 117–130. [[CrossRef](#)]
23. Gong, Y.J.; Li, J.J.; Zhou, Y.; Yun, L.; Chung, S.H.; Shi, Y.H.; Zhang, J. Genetic learning particle swarm optimization. *IEEE Trans. Cybern.* **2016**, *46*, 2277–2290. [[CrossRef](#)]
24. Nouiri, M.; Bekrar, A.; Jemai, A.; Niar, S.; Ammari, A.C. An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. *J. Intell. Manuf.* **2018**, *29*, 603–615. [[CrossRef](#)]
25. Wang, F.; Zhang, H.; Li, K.; Lin, Z.; Yang, J.; Shen, X. A hybrid particle swarm optimization algorithm using adaptive learning strategy. *Inf. Sci.* **2018**, *436*, 162–177. [[CrossRef](#)]
26. Aydilek, I.B. A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. *Appl. Soft Comput.* **2018**, *66*, 232–249. [[CrossRef](#)]
27. Xue, Y.; Xue, B.; Zhang, M.J. Self-adaptive particle swarm optimization for large-scale feature selection in classification. *ACM Trans. Knowl. Discov. Data* **2019**, *13*, 50. [[CrossRef](#)]
28. Song, X.F.; Zhang, Y.; Zhang, Y.; Guo, Y.N.; Sun, X.Y.; Wang, Y.L. Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data. *IEEE Trans. Evol. Comput.* **2020**, *24*, 882–895. [[CrossRef](#)]
29. Song, X.F.; Zhang, Y.; Gong, D.W.; Sun, X.Y. Feature selection using bare-bones particle swarm optimization with mutual information. *Pattern Recognit.* **2021**, *112*, 107804. [[CrossRef](#)]

30. Liang, J.J.; Qin, A.K.; Suganthan, P.N.; Baskar, S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.* **2006**, *10*, 281–295. [[CrossRef](#)]
31. Maltra, M.; Chatterjee, A. A hybrid cooperative-comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. *Expert Syst. Appl.* **2008**, *34*, 1341–1350.
32. Mahadevan, K.; Kannan, P.S. Comprehensive learning particle swarm optimization for reactive power dispatch. *Appl. Soft Comput.* **2010**, *10*, 641–652. [[CrossRef](#)]
33. Ali, H.; Khan, F.A. Attributed multi-objective comprehensive learning particle swarm optimization for optimal security of networks. *Appl. Soft Comput.* **2013**, *13*, 3903–3921. [[CrossRef](#)]
34. Hu, Z.Y.; Bao, Y.K.; Xiong, T. Comprehensive learning particle swarm optimization based memetic algorithm for model selection in short-term load forecasting using support vector regression. *Appl. Soft Comput.* **2014**, *25*, 15–25. [[CrossRef](#)]
35. Zhong, Y.W.; Lin, J.; Wang, L.; Zhang, H. Discrete comprehensive learning particle swarm optimization algorithm with Metropolis acceptance criterion for traveling salesman problem. *Swarm Evol. Comput.* **2018**, *42*, 77–88. [[CrossRef](#)]
36. Lin, A.P.; Sun, W. Multi-leader comprehensive learning particle swarm optimization with adaptive mutation for economic load dispatch problems. *Energies* **2019**, *12*, 116. [[CrossRef](#)]
37. Zhang, K.; Huang, Q.J.; Zhang, Y.M. Enhancing comprehensive learning particle swarm optimization with local optima topology. *Inf. Sci.* **2019**, *471*, 1–18. [[CrossRef](#)]
38. Lin, A.P.; Sun, W.; Yu, H.; Wu, G.; Tang, O.E. Adaptive comprehensive learning particle swarm optimization with cooperative archive. *Appl. Soft Comput.* **2019**, *77*, 533–546. [[CrossRef](#)]
39. Wang, J.J.; Liu, G.Y. Saturated control design of a quadrotor with heterogeneous comprehensive learning particle swarm optimization. *Swarm Evol. Comput.* **2019**, *46*, 84–96. [[CrossRef](#)]
40. Cao, Y.; Zhang, H.; Li, W.; Zhou, M.; Zhang, Y.; Chaovalitwongse, W.A. Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions. *IEEE Trans. Evol. Comput.* **2019**, *23*, 718–731. [[CrossRef](#)]
41. Chen, C.C.; Wang, X.C.; Yu, H.L.; Zhao, N.N.; Wang, M.J.; Chen, H.L. An enhanced comprehensive learning particle swarm optimizer with the elite-based dominance scheme. *Complexity* **2020**, *2020*, 4968063. [[CrossRef](#)]
42. Wang, S.L.; Liu, G.Y.; Gao, M.; Cao, S.; Wang, J.C. Heterogeneous comprehensive learning and dynamic multi-swarm particle swarm optimizer with two mutation operators. *Inf. Sci.* **2020**, *540*, 175–201. [[CrossRef](#)]
43. Zhang, X.; Sun, W.; Xue, M.; Lin, A.P. Probability-optimal leader comprehensive learning particle swarm optimization with Bayesian iteration. *Appl. Soft Comput.* **2021**, *103*, 107132. [[CrossRef](#)]
44. Zhou, S.B.; Sha, L.; Zhu, S.; Wang, L.M. Adaptive hierarchical update particle swarm optimization algorithm with a multi-choice comprehensive learning strategy. *Appl. Intell.* **2022**, *52*, 1853–1877. [[CrossRef](#)]
45. Tao, X.M.; Guo, W.J.; Li, X.K.; He, Q.; Liu, R.; Zou, J.R. Fitness peak clustering based dynamic multi-swarm particle swarm optimization with enhanced learning strategy. *Expert Syst. Appl.* **2022**, *191*, 116301. [[CrossRef](#)]
46. Shi, Y.H.; Eberhart, R.C. A modified particle swarm optimizer. In Proceedings of the IEEE Congress on Evolutionary Computation, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
47. Ratnaweera, A.; Halgamuge, S.K.; Watson, H.C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans. Evol. Comput.* **2004**, *8*, 240–255. [[CrossRef](#)]
48. Mendes, R.; Kennedy, J.; Neves, J. The fully informed particle swarm: Simpler, maybe better. *IEEE Trans. Evol. Comput.* **2004**, *8*, 204–210. [[CrossRef](#)]
49. Parsopoulos, K.E.; Vrahatis, M.N. UPSO: A Unified particle swarm optimization scheme. In Proceedings of the International Conference of Computational Methods in Sciences and Engineering 2004 (ICCMSE 2004), Attica, Greece, 19–23 November 2004; CRC Press: Boca Raton, FL, USA, 2004; pp. 868–873.
50. Engelbrecht, A.P. Heterogeneous particle swarm optimization. In Proceedings of the 7th International Conference on Swarm Intelligence (ANTS 2010), Brussels, Belgium, 8–10 September 2010; pp. 191–202.
51. Li, S.; Gu, Q.; Gong, W.Y.; Ning, B. An enhanced adaptive differential evolution algorithm for parameter extraction of photovoltaic models. *Energy Convers. Manag.* **2020**, *205*, 112443. [[CrossRef](#)]
52. Song, Y.; Wu, D.; Deng, W.; Gao, X.-Z.; Li, T.; Zhang, B.; Li, Y. MPPCEDE: Multi-population parallel co-evolutionary differential evolution for parameter optimization. *Energy Convers. Manag.* **2021**, *228*, 113661. [[CrossRef](#)]
53. Cui, H.; Guan, Y.; Chen, H. Rolling element fault diagnosis based on VMD and sensitivity MCKD. *IEEE Access* **2021**, *9*, 120297–120308. [[CrossRef](#)]
54. Zhou, X.B.; Ma, H.J.; Gu, J.G.; Chen, H.L.; Deng, W. Parameter adaptation-based ant colony optimization with dynamic hybrid mechanism. *Eng. Appl. Artif. Intel.* **2022**. [[CrossRef](#)]
55. Li, T.Y.; Shi, J.Y.; Deng, W.; Hu, Z.D. Pyramid particle swarm optimization with novel strategies of competition and cooperation. *Appl. Soft Comput.* **2022**, *121*, 108731. [[CrossRef](#)]
56. Ran, X.; Zhou, X.; Lei, M.; Tepsan, W.; Deng, W. A novel k-means clustering algorithm with a noise algorithm for capturing urban hotspots. *Appl. Sci.* **2021**, *11*, 11202. [[CrossRef](#)]
57. Zhang, Z.H.; Min, F.; Chen, G.S.; Shen, S.P.; Wen, Z.C.; Zhou, X.B. Tri-partition state alphabet-based sequential pattern for multivariate time series. *Cogn. Comput.* **2021**, 1–19. [[CrossRef](#)]
58. Tian, C.; Jin, T.; Yang, X.; Liu, Q. Reliability analysis of the uncertain heat conduction model. *Comput. Math. Appl.* **2022**, *119*, 131–140. [[CrossRef](#)]

59. Xu, G.; Bai, H.; Xing, J.; Luo, T.; Xiong, N.N.; Cheng, X.; Liu, S.; Zheng, X. SG-PBFT: A secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent Internet of vehicles. *J. Parallel Distrib. Comput.* **2022**, *164*, 1–11. [[CrossRef](#)]
60. Ahmadianfar, I.; Heidari, A.; Noshadian, S.; Chen, H.; Gandomi, A.H. INFO: An efficient optimization algorithm based on weighted mean of vectors. *Expert Syst. Appl.* **2022**, *195*, 116516. [[CrossRef](#)]
61. Wei, Y.Y.; Zhou, Y.Q.; Luo, Q.F.; Deng, W. Optimal reactive power dispatch using an improved slime Mould algorithm. *Energy Rep.* **2021**, *7*, 8742–8759. [[CrossRef](#)]
62. Zhang, X.; Wang, H.; Du, C.; Fan, X.; Cui, L.; Chen, H.; Deng, F.; Tong, Q.; He, M.; Yang, M.; et al. Custom-molded offloading footwear effectively prevents recurrence and amputation, and lowers mortality rates in high-risk diabetic foot patients: A multicenter, prospective observational study. *Diabetes Metab. Syndr. Obes. Targets Ther.* **2022**, *15*, 103–109. [[CrossRef](#)]
63. Chen, H.; Zhang, Q.; Luo, J. An enhanced Bacterial Foraging Optimization and its application for training kernel extreme learning machine. *Appl. Soft Comput.* **2020**, *86*, 105884. [[CrossRef](#)]
64. Wu, X.; Wang, Z.; Wu, T.; Bao, X. Solving the family traveling salesperson problem in the adleman–lipton model based on DNA computing. *IEEE Trans. Nanobiosci.* **2022**, *21*, 75–85. [[CrossRef](#)]
65. Sarkar, B.; Tayyab, M.; Kim, N.; Habib, M.S. Optimal production delivery policies for supplier and manufacturer in a constrained closed-loop supply chain for returnable transport packaging through metaheuristic approach. *Comput. Ind. Eng.* **2019**, *135*, 987–1003. [[CrossRef](#)]
66. Wu, E.Q.; Zhou, M.C.; Hu, D.E.; Zhu, L.; Tang, Z.; Qiu, X.-Y.; Deng, P.-Y.; Zhu, L.-M.; Ren, H. Self-paced dynamic infinite mixture model for fatigue evaluation of pilots’ brains. *IEEE Trans. Cybern.* **2020**. [[CrossRef](#)] [[PubMed](#)]
67. Sarkar, A.; Guchhait, R.; Sarkar, B. Application of the Artificial Neural Network with Multithreading Within an Inventory Model under Uncertainty and Inflation. *Int. J. Fuzzy Syst.* **2022**. [[CrossRef](#)]
68. Gupta, S.; Haq, A.; Ali, I.; Sarkar, B. Significance of multi-objective optimization in logistics problem for multi-product supply chain network under the intuitionistic fuzzy environment. *Complex Intell. Syst.* **2021**, *7*, 2119–2139. [[CrossRef](#)]
69. Sepehri, A.; Mishra, U.; Sarkar, B. A sustainable production-inventory model with imperfect quality under preservation technology and quality improvement investment. *J. Clean. Prod.* **2021**, *310*, 127332. [[CrossRef](#)]
70. Wu, J.; Wang, Z. A hybrid model for water quality prediction based on an artificial neural network, wavelet transform, and long short-term memory. *Water* **2022**, *14*, 610. [[CrossRef](#)]
71. Shao, H.D.; Lin, J.; Zhang, L.W.; Galar, D.; Kumar, U. A novel approach of multisensory fusion to collaborative fault diagnosis in maintenance. *Inf. Fusion* **2021**, *74*, 65–76. [[CrossRef](#)]
72. Zhao, H.; Liu, J.; Chen, H.; Li, Y.; Xu, J.; Deng, W. Intelligent diagnosis using continuous wavelet transform and gauss convolutional deep belief network. *IEEE Trans. Reliab.* **2022**. [[CrossRef](#)]