# *Predominant Cognitive Learning Particle Swarm Optimization for Global Numerical Optimization*

*mathematics*

MDPI

*Article*

# Predominant Cognitive Learning Particle Swarm Optimization for Global Numerical Optimization

**Qiang Yang [1], Yufei Jing [1], Xudong Gao [1], Dongdong Xu [1,*], Zhenyu Lu [1], Sang-Woon Jeon [2] and Jun Zhang [2,3]**

[1] School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 210044, China; qiang_yang@nuist.edu.cn (Q.Y.); 201983460037@nuist.edu.cn (Y.J.); 003103@nuist.edu.cn (X.G.); 001114@nuist.edu.cn (Z.L.)

[2] Department of Electrical and Electronic Engineering, Hanyang University, Ansan 15588, Korea; sangwoonjeon@hanyang.ac.kr (S.-W.J.); junzhang@ieee.org (J.Z.)

[3] Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung 413310, Taiwan

* Correspondence: 001601@nuist.edu.cn

**Abstract:** Particle swarm optimization (PSO) has witnessed giant success in problem optimization. Nevertheless, its optimization performance seriously degrades when coping with optimization problems with a lot of local optima. To alleviate this issue, this paper designs a predominant cognitive learning particle swarm optimization (PCLPSO) method to effectively tackle complicated optimization problems. Specifically, for each particle, a new promising exemplar is constructed by letting its personal best position cognitively learn from a better personal experience randomly selected from those of others based on a novel predominant cognitive learning strategy. As a result, different particles preserve different guiding exemplars. In this way, the learning effectiveness and the learning diversity of particles are expectedly improved. To eliminate the dilemma that PCLPSO is sensitive to the involved parameters, we propose dynamic adjustment strategies, so that different particles preserve different parameter settings, which is further beneficial to promote the learning diversity of particles. With the above techniques, the proposed PCLPSO could expectedly compromise the search intensification and diversification in a good way to search the complex solution space properly to achieve satisfactory performance. Comprehensive experiments are conducted on the commonly adopted CEC 2017 benchmark function set to testify the effectiveness of the devised PCLPSO. Experimental results show that PCLPSO obtains considerably competitive or even much more promising performance than several representative and state-of-the-art peer methods.

**Keywords:** predominant cognitive learning; multimodal problems; particle swarm optimization; global numerical optimization; black-box optimization

**MSC:** 68-04; 65-04

check for updates

**Citation:** Yang, Q.; Jing, Y.; Gao, X.; Xu, D.; Lu, Z.; Jeon, S.-W.; Zhang, J. Predominant Cognitive Learning Particle Swarm Optimization for Global Numerical Optimization. *Mathematics* **2022**, *10*, 1620. https://doi.org/10.3390/math10101620

Academic Editors: Dragos Sburlan, Elena Bautu and Jüri Majak

Received: 2 March 2022
Accepted: 7 May 2022
Published: 10 May 2022

## 1. Introduction

Optimization problems emerge commonly and become more and more complicated in many research fields and industrial engineering [1,2], such as object detection and tracking [3,4], automatic design of algorithms for visual attention [5,6], path planning optimization [7,8], and control of pollutant spreading on social networks [9]. In particular, these complicated optimization problems usually are non-differentiable, discontinuous, non-convex, non-linear, or multimodal [10–12]. Confronted with such kinds of complicated optimization problems, the optimization effectiveness of traditional optimization methods, such as conjugate gradient methods [13,14], space-filling curve methods [15,16], quasi-Newton methods [17,18], line search methods [19–21], and trust-region methods [22,23], deteriorates rapidly. In extreme cases, they are even infeasible for solving these complex problems. As a consequence, effective optimization algorithms are increasingly demanded to solve increasingly emerging complex optimization problems, such that the development of related fields could be boosted.

In recent years, evolutionary algorithms (EAs), such as particle swarm optimization (PSO) [24,25] and differential evolution (DE) [26,27], have presented good optimization ability in problem optimization, especially in solving those problems that traditional optimization methods cannot tackle, such as locating multiple global optima of optimization problems [28–31], and simultaneously optimizing more than one objective [32,33], etc. Different from traditional mathematical optimization methods [34–36], which usually adopt only one feasible solution to iteratively search the solution space, EAs generally employ a population of candidate solutions to undergo iterative evolution to seek the global optimum. In this manner, compared with traditional mathematical approaches [16,36,37], EAs own many unique merits. (1) EAs have no requirements on the mathematical properties of the problem to be optimized [38,39], or even can deal with problems without mathematical models. However, most traditional optimization methods [16–18], especially gradient-based approaches [17,18], have critical requirements on the properties of optimization problems, such as continuous, differentiable, and convex. Theoretically, EAs can be adopted to optimize any kinds of problems. However, in the literature [40,41], EAs are mainly employed to tackle optimization problems that traditional optimization techniques cannot cope with. (2) EAs usually have strong global search ability since they maintain a population of individuals to explore the search space in different directions [38,41]. Therefore, falling into local regions could be avoided with a high probability. Nevertheless, traditional mathematical optimization methods [42–44] usually employ only one feasible candidate, and thus search the solution space in only one direction to seek the global optimum. As a consequence, it is likely for them to fall into local areas, especially when dealing with optimization problems with a lot of wide and flat local basins.

As a kind of EA, PSO [45,46] has been successfully employed to cope with different kinds of optimization problems since its first introduction in 1995 by Kennedy and Eberhart [47,48]. During the optimization, PSO maintains a population of candidate feasible solutions to search the solution space iteratively. By means of its great advantages, such as independence of the mathematic properties of problems to be optimized, fast convergence, and inherent parallelism [49], PSO has been researched by a lot of researchers. As a result, PSO has been not only widely applied to solve complex problems, such as multimodal optimization [28–31], and multi-objective optimization [32,33], but also has been commonly adopted to tackle real-world optimization problems, such as vehicle routing problems [45,50], neural networks [51,52], and task assignment [53,54].

In the literature [55–59], it is widely accepted that the learning strategy in updating the velocity of particles has a significant influence on assisting PSO to obtain good performance because it determines the way of information diffusion within the swarm. As a result, researchers have designed a lot of novel learning schemes for PSO to promote its optimization performance [11,60], such as comprehensive learning strategies [56,61,62], orthogonal learning mechanisms [57,63,64], and hybrid algorithm learning methods [55,65,66]. Roughly, existing learning mechanisms for PSO could be classified into two categories: exemplar construction-based learning methods [55–57] and topology-based learning strategies [58,59,67].

Exemplar construction-based learning strategies aim to construct new learning exemplars for particles to learn from [55–57] that may not be visited by particles. In most existing studies, the constructed exemplar is generated by some dimension recombination methods based on historically best positions of particles. By this method, it is expected that the constructed exemplars could provide guidance for the evolution of the swarm, so that particles could move to more promising areas. Along this line, the most representative approach is the comprehensive learning PSO (CLPSO) [56]. To improve the construction efficiency of exemplars, researchers have devised many other exemplar-construction methods, such as the orthogonal learning PSO (OLPSO) [57] and the genetic learning PSO (GL-PSO) [55].

Different from exemplar construction-based learning strategies, topology-based learning strategies mainly adopt certain kinds of topologies to select guiding exemplars to update particles [58,59,67,68]. Different topology structures affect the way of information exchange between particles and the speed of information circulation, thereby affecting the performance of PSO. Specifically, in most topology-based learning strategies, each particle cognitively learns from its own historically best position and socially learns from the historically best position among the neighbors connected by the associated topologies. In the classical PSO [47,48], a global topology connecting all particles was utilized to select the best position among the personal best positions of all particles as the learning exemplar for each particle. This global best position brings in overly greedy attraction, such that when dealing with multimodal problems [10,48], the swarm often falls into local regions. To alleviate this dilemma, many different neighborhood topologies have been designed [58], such as ring topology, four-cluster topology, pyramid topology, and square topology. Some researchers even proposed dynamic topologies [59,68] and composited different topologies [67] to select promising exemplars to direct the update of particles, so that the learning abilities of particles are further promoted.

Though PSO has been advanced significantly, and a lot of remarkable PSO variants [55,56,59,62,69–71] have shown their great feasibility in coping with optimization problems, their optimization ability encounters great challenges when dealing with complicated problems with a number of interacting variables and a lot of wide and flat local basins. Unfortunately, these complicated problems are ubiquitous in the era of big data and Internet of Things (IoT) [72]. As a result, there is still an increasing and urgent demand for effective and efficient PSOs to tackle the increasingly emerging complex optimization problems.

To further promote the optimization performance of PSO in dealing with complicated optimization problems, this paper designs a predominant cognitive learning particle swarm optimization (PCLPSO) algorithm, which utilizes a predominant cognitive learning strategy to construct guiding exemplars for particles. Specifically, the main components of PCLPSO are summarized as follows:

(1) A predominant cognitive learning strategy (PCL) is devised to construct guiding exemplars to update particles. Different from existing exemplar construction-based learning PSOs [55–57] that construct the guiding exemplars in an elementwise way, the proposed PCL constructs a promising exemplar to guide the update of each particle by letting its personal best position cognitively learn from a predominant one randomly selected from those better than the personal best position of the updated particle. On the one hand, the personal best position of this particle learns from a better one, and thus it is expected that the constructed exemplar is more promising. As a result, the learning effectiveness of particles is expectedly promoted. On the other hand, due to the random selection in PCL, different particles generally preserve different guiding exemplars, and thus the learning diversity of particles is expectedly improved. In this way, the proposed PCLPSO could expectedly compromise the search diversity and the search convergence well to find satisfactory solutions.

(2) Dynamic parameter adjustment strategies are further designed to alleviate the predicament that PCLPSO is sensitive to involved parameters. With these dynamic strategies, different particles usually preserve different parameter settings, which is beneficial for further improving the learning diversity of particles.

To validate the optimization effectiveness and efficiency of PCLPSO, comprehensive experiments were carried out on the commonly adopted CEC 2017 benchmark function set [73] with three dimensionality (namely 30, 50, and 100) by comparing PCLPSO with seven representative and state-of-the-art PSO variants. At the same time, deep investigations on PCLPSO were also executed to determine what contributes to its promising performance.

The remainder of this paper is arranged as follows. Closely related works on PSOs are briefly reviewed in Section 2. Then, the developed PCLPSO is described in Section 3.

In Section 4, comparative experiments are executed to testify the effectiveness of PCLPSO. Finally, Section 5 concludes this paper.

## 2. Related Work

Without loss of generality, this paper aims to find the global minima of the following defined problems:

$$\text{minimize } f(\boldsymbol{x}) \quad \boldsymbol{x} = [x_1, \ldots, x_d, \ldots x_D] \in R^{\mathrm{D}} \tag{1}$$

where $\boldsymbol{x}$ is the decision variable vector composed of $D$ variables. In this paper, the objective value of the problem is used as the fitness of one particle.

In the literature, there are an ocean of optimization methods, including traditional mathematical optimization methods [15,18,20,21,36,44] and heuristic algorithms, such as evolutionary algorithms [25,27,30,38,55,56]. However, since this paper mainly proposes a PSO variant to solve complicated optimization problems, we mainly review closely related studies on PSO in the following.

### *2.1. Canonical PSO*

Typically, each particle in the classical PSO [47,48] is updated by learning from its own experience and the social experience of the entire swarm. Specifically, in PSO, each particle is represented by one position vector and one velocity vector. Based on these two vectors, each particle is updated in the following way:

$$\boldsymbol{v}_i^{t+1} = \omega \times \boldsymbol{v}_i^t + c_1 \times r_1 \times (\boldsymbol{pbest}_i^t - \boldsymbol{x}_i^t) + c_2 \times r_2 \times (\boldsymbol{gbest}^t - \boldsymbol{x}_i^t) \tag{2}$$

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1} \tag{3}$$

where $\boldsymbol{x}_i$ is the position vector of the $i$th particle, $\boldsymbol{v}_i$ is its velocity vector, and $\boldsymbol{pbest}_i$ is its personal best position found so far, while $\boldsymbol{gbest}$ is the global best position of the entire swarm found so far. $t$ denotes the generation index. $\omega$ denotes the inertia weight. $c_1$ and $c_2$ represent two acceleration coefficients in charge of the effect of $\boldsymbol{pbest}_i$ and $\boldsymbol{gbest}$ on the updated particle. $r_1$ and $r_2$ are uniformly and randomly sampled within [0, 1].

Equation (2) shows that in the canonical PSO, all particles learn from the global best position $\boldsymbol{gbest}$ of the whole swarm found so far. Such an exemplar is too greedy and thus easily leads to falling into local basins of the swarm when coping with optimization problems with many local regions [49,74,75].

To improve the effectiveness of PSO in exploring and exploiting multimodal space, many researchers have developed a lot of novel and effective learning schemes [25,55–59] to guide the update of particles. In a broad sense, existing learning methods are roughly separated into two types: constructive learning strategies [55–57] and topology-based learning strategies [58,59,67].

### *2.2. Constructive Learning Strategies for PSO*

Constructive learning strategies [55–57] mainly adopt some dimension recombination methods to randomly recombine dimensions of multiple personal best positions of particles to construct promising guiding exemplars to update particles. In the literature, the most representative constructive learning PSOs are the comprehensive learning PSO (CLPSO) [56], the orthogonal learning PSO (OLPSO) [57], and the genetic learning PSO (GLPSO) [55], which will be described in detail next.

#### 2.2.1. CLPSO

CLPSO [56] utilizes the devised comprehensive learning (CL) strategy to recombine dimensions of $\boldsymbol{pbest}$s of different particles to construct new exemplars dimension by dimension. Specifically, the velocity of each particle is updated in the following way:

$$v_{i,d}^{t+1} = \omega \times v_{i,d}^t + c \times r_d \times (pbest_{f_i(d),d}^t - x_{i,d}^t) \tag{4}$$

where $x_{i,d}$ is the $d$th dimension in the position vector of the $i$th particle, while $v_{i,d}$ is the $d$th dimension in the velocity vector of the $i$th particle. $pbest_{fi(d),d}$ is the $d$th dimension in the personal best position $pbest_{fi(d)}$, where $f_i(d)$ denotes the index of the selected personal best position for the $d$th dimension in the updated particle; $t$ denotes the generation index; $\omega$ denotes the inertia weight; $c$ represents the acceleration coefficient and $r_d$ is uniformly and randomly sampled within [0, 1].

With the above CL strategy, each particle learns from multiple historical experiences of particles. Since the personal best positions are all randomly selected, the constructed exemplars for different particles are likely different and thus the learning diversity of particles is expectedly improved largely. On the other hand, by learning from multiple personal best positions, the potentially useful information embedded in the multiple personal best positions could be integrated to direct the update of particles. As a result, the learning effectiveness of particles is expectedly promoted as well.

To further promote the optimization performance of CLPSO, some researchers have attempted to introduce or design some additional mechanisms [61,62,69,76] to assist the evolution of the swarm. For example, Lynn et al. [62] designed a heterogeneous CLPSO (HCLPSO) by partitioning the swarm into two subpopulations with one responsible for exploration and another in charge of exploitation. With respect to the subpopulation for exploration, the guiding exemplars are constructed by using ***pbest***s of particles in this subpopulation. While regarding the subpopulation for exploitation, the guiding exemplars are constructed by ***pbest***s of all particles in the entire swarm. In [76], a heterogeneous comprehensive learning and dynamic multi-swarm particle swarm optimizer (HCLDMS-PSO) was devised by introducing an improved dynamic multi-swarm (DMS) scheme and two mutation operators (a non-uniform mutation operator and a Gaussian mutation operator) into HCLPSO to further promote its optimization performance. In [77], an adaptive strategy for regulating the probability of conducting the CL strategy and a cooperative archive (CA) were devised and then embedded into CLPSO, leading to an adaptive CLPSO with collaborative archiving (ACLPSO-CA). In [69], a novel local search method was developed and then introduced into CLPSO, leading to a variant of CLPSO, named CLPSO-LS. With the help of this local search operator, the accuracy of the obtained solution is improved evidently. In [61], a local optima topology (LOT) structure was devised and then introduced into CLPSO, leading to CLPSO-LOT. Specifically, such a topology structure comprises the local optima, so that the search space of particles could be enlarged and the convergence of the swarm could be boosted with a certain probability.

### 2.2.2. OLPSO

Although CLPSO has shown promising performance on complicated problems, such as multimodal problems, it is inefficient to construct a new valid exemplar because the reorganization of dimensions is completely random and directionless. To afford effective recombination of dimensions, in [57], OLPSO was devised through the orthogonal experimental design to roughly seek useful recombination of the historical positions of particles. To be specific, the algorithm selects the dimension combinations with the best average fitness value by constructing an orthogonal matrix and then constructs a potentially useful exemplar for each particle. In particular, the velocity of each particle is updated in the following way:

$$v_{i,d}^{t+1} = w \times v_{i,d}^{t} + c \times r_d \times (pbest_{o,d}^{t} - x_{i,d}^{t}) \quad (5)$$

where $x_{i,d}$ and $v_{i,d}$ are the $d$th dimension in the position and the velocity vectors of the $i$th particle, respectively. $pbest_{o,d}$ is the determined personal best position by the orthogonal matrix for the $d$th dimension; $t$ denotes the generation index; $\omega$ denotes the inertia weight; $c$ represents the acceleration coefficient; $r_d$ is uniformly and randomly sampled from [0, 1].

With the orthogonal matrix, the dimension recombination of different personal best positions is effective to generate a good exemplar for each particle with a high probability. Therefore, OLPSO achieves much better performance than CLPSO when dealing with multimodal problems [57,63,78].

After the emergence of OLPSO, researchers have introduced a lot of other techniques [63,64,78] into OLPSO to further promote its optimization performance. For example, in [63], a variable relocation strategy (VRS) is combined with the OL strategy to develop a variant of OLPSO, named OLPSO-VRS to effectively solve dynamic optimization problems. In [64], the metropolis-based probabilistic acceptance criterion was incorporated into OLPSO, leading to a hybrid OLPSO (HOLPSO). The searching capability of the algorithm is improved by selecting guiding particles based on the probabilistic acceptance mechanism. In [78], a quadratic interpolation based OLPSO-G (QIOLPSO-G) was proposed, where a quadratic interpolation-based construction mechanism was applied to ***pbest***s of all particles to construct effective exemplars.

Although the above OLPSO variants can generate more effective learning exemplars with high probability, they consume a lot of fitness evaluations, because the OL strategy requires extensive fitness evaluations in the orthogonal experimental design.

### 2.2.3. GL-PSO

Different from the above two construction methods, GL-PSO [55] utilized the selection strategy, the crossover scheme, and the mutation mechanism in genetic algorithms (GA) to construct learning exemplars for each particle. At first, GL-PSO constructs an exemplar based on the classical update strategy of PSO as follows:

$$e_{i,d}^{t+1} = \frac{c_1 \times r_{1,d} \times pbest_{i,d}^t + c_2 \times r_{2,d} \times gbest_d^t}{c_1 \times r_{1,d} + c_2 \times r_{2,d}} \tag{6}$$

where $pbest_{i,d}$ is the $d$th dimension in $\boldsymbol{pbest_i}$ of the $i$th particle; $gbest_d$ is the $d$th dimension in $\boldsymbol{gbest}$ of the entire swarm; $e_{i,d}$ is the $d$th dimension of the constructed learning exemplar. $t$ denotes the generation index; $c_1$ and $c_2$ represent two acceleration coefficients, and $r_{1,d}$ and $r_{2,d}$ are randomly and uniformly sampled from [0, 1].

Then, GL-PSO constructs another learning exemplar for each particle via utilizing the operators in GA as follows:

(1) *Crossover:* First, the crossover operation is used to recombine ***pbest***s and ***gbest*** to build an offspring $\boldsymbol{o}_i$ in the following way:

$$o_{i,d}^{t+1} = \begin{cases} r_d \times pbest_{i,d}^t + (1 - r_d) \times gbest_d^t, & \text{if } f(pbest_i^t) < f(pbest_{k_d}^t) \\ pbest_{k_d,d}^t, & \text{otherwise} \end{cases} \tag{7}$$

where $r_d$ is randomly and uniformly sampled in [0, 1], $k_d$ is the index of a randomly selected ***pbest*** for the $d$th dimension; $f(\bullet)$ is the fitness function.

From Equation (7), it is found that this crossover operator actually generates the offspring dimension by dimension. Specifically, for each dimension, one ***pbest*** is first randomly chosen from all ***pbest***s in the current generation, and then the selected ***pbest*** is compared with $\boldsymbol{pbest_i}$. If the selected ***pbest*** is better, the dimension of the offspring inherits the value from the selected ***pbest***; otherwise, it inherits the value from the linear composition of $\boldsymbol{pbest_i}$ and ***gbest***.

(2) *Mutation*: After crossover, the generated offspring $\boldsymbol{o}_i$ performs the uniform mutation operation with a probability ($pm$) as follows:

$$o_{i,d}^{t+1} = rand(lb_d,\ ub_d), \text{ if } r_d < pm \tag{8}$$

where $lb_d$ and $ub_d$ are the lower bound and the upper bound of the $d$th dimension, respectively; $r_d$ is randomly and uniformly sampled from [0, 1]; $pm$ is the mutation probability.

Such a mutation operation is actually the typical uniform mutation. Specifically, for each dimension, a real number $r_d$ is first uniformly sampled within [0, 1]. Then, if the generated $r_d$ is lower than $pm$, the $d$th dimension of $\boldsymbol{o}_i$ is resampled in the search range [55].

(3) *Selection*: At last, the selection operation is performed to determine the final learning exemplar for each particle to update. Specifically, the selection operator is conducted between the first learning exemplar constructed by Equation (6) and the one constructed by Equations (7) and (8) as follows:

$$e_i^{t+1} = \begin{cases} o_i^{t+1}, \text{ if } f(o_i^{t+1}) < f(e_i^{t+1}) \\ e_i^{t+1}, \text{ otherwise.} \end{cases} \tag{9}$$

After the selection, the velocity of each particle is updated in the following way:

$$v_{i,d}^{t+1} = \omega \times v_{i,d}^{t} + c \times r_d \times (e_{i,d}^{t+1} - x_{i,d}^{t}) \tag{10}$$

In GL-PSO, due to the selection operator and the random construction of the learning exemplars, the finally constructed exemplars for different particles are expectedly different. With this exemplar construction strategy, the global search capability and the optimization performance of PSO are expectedly enhanced. Inspired from GL-PSO, many researchers have attempted to employ operators in other evolutionary algorithms to construct learning exemplars based on ***pbest***s of particles and ***gbest*** of the whole swarm [65,66,79]. For instance, in [65], a global GL-PSO with diversity enhancement (GGL-PSOD) was developed by changing the global topology with a local ring topology during the construction of exemplars to further enhance the search diversity of the algorithm. In [79], a triple archive PSO (TAPSO) maintains three archives to generate a promising guiding exemplar for each particle in the dimension-wise way. Specifically, this algorithm constructs a new exemplar based on the operators of GA by randomly selecting two particles the first archive and the second archive. In [66], a PSO with two differential mutation operators (PSOTD) was devised by designing a topology structure composed of two swarms and two layers. In this algorithm, two different differential mutation operations with two different control parameters are utilized to generate learning exemplars for particles.

The above variants of GL-PSO have shown promising performance in solving optimization problems. However, during the exemplar construction, they all consume many function evaluations to evaluate the constructed exemplars. This definitely reduces the number of fitness evaluations used for evolving the swarm, which may lead to insufficient evolution of the swarm to find high-quality solutions.

### 2.3. Topology-Based Learning Strategies for PSO

Different from the constructive learning strategies, the topology-based learning strategies [58,80] mainly select learning exemplars from ***pbest***s in the current generation based on some certain kinds of topologies. Specifically, the topologies are used to connect particles via information exchange to seek proper learning exemplars to guide the update of particles [81,82].

In [58], Mendes and Kennedy have designed many different neighborhood topologies, such as ring topology, four-cluster topology, pyramid topology, and square topology, where each particle is connected to a small number of individuals from a local community. Unlike the greedy global topology, these local topologies can select less greedy exemplars and thus high diversity could be maintained, which is profitable for PSO to avoid falling into local basins. In [80], Zou et al. devised a PSO based on a ring neighborhood topology, which is formed on the basis of the calculated Euclidean distance between particles, to solve multimodal optimization problems. Specifically, each particle builds its own topology and evolves in the associated local range. Instead of using fixed topologies, some researchers have employed dynamic topologies [59,83] to further enhance the search diversity of particles. Since different topology-based learning strategies preserve different properties and advantages, some researchers have attempted to use different topology-based learning strategies to update particles, so that different particles may interact with others with different topologies [67].

Though the abovementioned PSO variants have shown good performance on certain kinds of optimization problems, issues such as premature convergence and stagnation still occur when they deal with complicated problems, such as problems with a lot of interacting variables and many wide and flat local basins. With these limitations, the abovementioned algorithms cannot be adopted to tackle multimodal problems that increasingly emerge in real-world applications. To promote the optimization performance of PSO in coping with complicated problems, we design a predominant cognitive learning particle swarm optimization (PCLPSO) in this paper by constructing promising exemplars to improve the learning diversity and the learning effectiveness of particles.

## 3. Proposed PCLPSO

To elevate the search effectiveness and the search diversity of particles in tackling complicated optimization problems, this paper devises a predominant cognitive learning particle swarm optimization (PCLPSO) via utilizing the devised predominant cognitive learning strategy to construct promising learning exemplars for particles to update. Therefore, the proposed PCLPSO is a constructive-learning-based PSO variant.

### 3.1. Predominant Cognitive Learning Strategy

In the three classical constructive learning PSOs (namely CLPSO [56], OLPSO [57], and GL-PSO [55]), the construction efficiency for promising guiding exemplars cannot be guaranteed in CLPSO on account of its random selection of $\boldsymbol{pbest}$s dimension by dimension. Although OLPSO and GL-PSO construct promising exemplars more efficiently than CLPSO, they usually consume a great number of fitness evaluations during the construction of promising exemplars, which leads to the number of fitness evaluations used for the swarm evolution being reduced and consequently it is not beneficial for the algorithm to seek high-accuracy solutions.

To alleviate the above issues, this paper proposes a predominant cognitive learning strategy (PCL) to construct promising exemplars for particles. Specifically, given that the number of particles maintained in the swarm is $NP$, then, we sort the personal best positions $\boldsymbol{pbest}$s of all particles from the best to the worst. Subsequently, for each particle ($\boldsymbol{x_i}$, $0 \le i \le NP$), we construct a promising guiding exemplar as follows:

$$\boldsymbol{e}_i^{t+1} = \begin{cases} \boldsymbol{gbest}^t & \text{if } \boldsymbol{pbest}_i^t = \boldsymbol{gbest}^t \\ \boldsymbol{pbest}_i^t + F_i \times (\boldsymbol{pbest}_{rb}^t - \boldsymbol{pbest}_i^t) & \text{otherwise} \end{cases} \tag{11}$$

where $\boldsymbol{pbest_i}$ is the personal best position found by the $i$th particle so far; $\boldsymbol{gbest}$ is the global best position found by the entire swarm so far; $\boldsymbol{pbest_{rb}}$ is the personal best position randomly selected from those which are better than $\boldsymbol{pbest_i}$; $\boldsymbol{e_i}$ is the constructed exemplar for the $i$th particle; $F_i$ is a control parameter within [0, 1], which can be seen as a learning step of $\boldsymbol{pbest_i}$ of the $i$th particle, and $t$ denotes the generation index.

From Equation (11), we can see that the learning exemplar ($\boldsymbol{e_i}$) for each particle ($\boldsymbol{x_i}$) is constructed by letting its cognitive experience (namely $\boldsymbol{pbest_i}$) learn from a predominant cognitive experience of other particles (namely a better personal best position $\boldsymbol{pbest_{rb}}$), which is randomly selected from those $\boldsymbol{pbest}$s with better fitness than $\boldsymbol{pbest_i}$. In this way, the constructed exemplar is expectedly close to promising areas. It means that if the personal best position ($\boldsymbol{pbest_i}$) of one particle is just $\boldsymbol{gbest}$, we do not construct a guiding exemplar for this particle since there is no better one to learn from. In this situation, we directly use $\boldsymbol{pbest_i}$ (also $\boldsymbol{gbest}$) to direct the update of this particle.

After the construction of the guiding exemplar for the $i$th particle, it is updated in the following way:

$$\boldsymbol{v}_i^{t+1} = \omega \times \boldsymbol{v}_i^t + c_i \times r \times (\boldsymbol{e}_i^{t+1} - \boldsymbol{x}_i^t) \tag{12}$$

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1} \tag{13}$$

where $\boldsymbol{x_i}$ and $\boldsymbol{v_i}$ denote the position and the velocity vectors of the *i*th particle, respectively; $\boldsymbol{e_i}$ is the constructed guiding exemplar for the *i*th particle; $t$ denotes the generation index; $\omega$ denotes the inertia weight; $c_i$ represents the acceleration coefficient for the *i*th particle; $r$ is randomly and uniformly sampled from [0, 1].

In-depth observation of Equations (11) and (12) shows that the proposed PCL strategy preserves the following merits:

(1) The constructed guiding exemplar for each particle is hopefully more promising than its personal best position because the exemplar is generated by letting its personal best position cognitively learn from a randomly selected better one. Therefore, the learning effectiveness of particles is hopefully promoted, which helps particles locate optimal areas fast.

(2) Due to the random selection of the learning candidates in PCL, the constructed exemplars to guide the update of different particles are likely different. Hence, the learning diversity of particles is also expectedly promoted, which is helpful for the swarm to escape from local basins.

(3) In particular, we find that different particles have different numbers of learning candidates in PCL to construct exemplars, which results in ***pbest***s of different particles having different ranges to learn. Specifically, the better the ***pbest*** is, the fewer candidates this ***pbest*** has to learn from, and thus the narrower range this position moves in. Implicitly, it is found that particles with worse personal best positions prefer to explore the solution space, while particles with better personal best positions prefer to exploit the solution space.

(4) By means of the above merits, the devised PCL is expected to compromise exploration and exploitation well to search the solution space appropriately. Therefore, it is likely that the proposed PCLPSO could achieve good performance in coping with different kinds of optimization problems.

Remark

Compared with the three classical constructive learning PSOs, namely CLPSO [56], OLPSO [57], and GL-PSO [55], the proposed PCLPSO differs from them in the following aspects:

(1) As shown in Equations (11) and (12), the proposed PCLPSO constructs the guiding exemplars in a whole way, namely taking all dimensions together to construct the guiding exemplars, while CLPSO (Equation (4)), OLPSO (Equation (5)), and GL-PSO (Equation (7)) all construct the guiding exemplars in an elementwise way, namely constructing the exemplars dimension by dimension. By learning from predominant experience of other particles as a whole, the proposed PCL could implicitly take the variable correlations into consideration to effectively construct promising guiding exemplars. At the same time, it can also reduce the computational complexity and time cost during the exemplar construction.

(2) The proposed PCLPSO constructs a learning exemplar to guide the update of each particle by letting the personal best position of this particle cognitively learn from a randomly selected better personal best position. While CLPSO constructs a guiding exemplar for each particle dimension by dimension based on the personal best positions of all particles. Since the dimension recombination of the personal best positions are random, the quality of the constructed exemplar in CLPSO is uncertain. However, in PCLPSO, by learning from a better position, the quality of the constructed exemplar is expectedly improved. Therefore, compared with CLPSO, the proposed PCLPSO preserves higher efficiency in constructing more promising exemplars.

(3) There is no additional consumption of fitness evaluations in the exemplar construction in PCLPSO, while in both OLPSO and GL-PSO, a lot of fitness evaluations are consumed during the construction of the guiding exemplars. In OLPSO, a lot of fitness evaluations are used to construct the orthogonal matrix to seek the effective recombination of dimensions to generate a more promising exemplar. In GL-PSO, in the selection operation, for each particle, two fitness evaluations are taken to evaluate the

constructed exemplars, so that a better one can be determined as the guiding exemplar of this particle. Different from these two constructive learning PSOs, PCLPSO directly utilizes the constructed exemplars to direct the update of particles. Though it cannot guarantee that the constructed exemplar for each particle is definitely better than its personal best positions, the constructed exemplars are hopefully better than the associated personal best positions since they are constructed by letting the associated personal best positions learn from predominant ones. Therefore, it is expected that the proposed PCLPSO achieves more promising performance than OLPSO and GL-PSO.

### *3.2. Dynamic Strategies for Control Parameters*

From Equations (11) and (12), it is found that PCLPSO has three key parameters, namely inertia weight $\omega$, acceleration coefficient $c$, and control parameter $F$. With respect to the inertia weight $\omega$, we directly utilize the following linear decay strategy, which is commonly employed in the literature [56,62,67,69,84]:

$$\omega = 0.9 - 0.7 \times \frac{t}{T_{max}} \tag{14}$$

where $t$ is the current generation, while $T_{max}$ denotes the preset maximum number of generations.

Taking observation of Equation (14), we can see that the inertia weight $\omega$ linearly reduces from 0.9 to 0.2 as the evolution progresses. Therefore, in the early stage, a large $\omega$ is maintained to keep the moving inertia of particles, which is profitable for particles searching the solution space with high diversity. In the late stage, a small $\omega$ is maintained to decrease the influence of the inertia part. As a result, the swarm expectedly tends to exploit the found promising areas to obtain high-accuracy solutions.

With respect to the control parameter $\boldsymbol{F}$ and the acceleration coefficient $\boldsymbol{c}$, we devise the following dynamic strategies.

#### 3.2.1. Adaptive Strategy for $F$

In Equation (11), the control parameter $\boldsymbol{F}$ controls the learning step that the personal best position ($\boldsymbol{pbest_i}$) of the updated particle takes learning from the randomly selected predominant one ($\boldsymbol{pbest_{rb}}$). Therefore, it has a great effect on the quality of the constructed exemplars. In particular, a too large $F_i$ leads to too greedy learning, which results in the constructed exemplar being too close to the randomly selected $\boldsymbol{pbest_{rb}}$. In this situation, the updated particle may approach promising areas too fast, which may lead to the risk of falling into local basins and premature convergence. By contrast, a too small $F_i$ results in insufficient learning, which results in the constructed exemplar for each particle being close to its ***pbest***. In this situation, the learning effectiveness of particles is improved very limitedly, which may slow down the convergence. Moreover, since the learning ranges of different particles are different, the settings of $F_i$ should be different for different particles as well.

Bearing the above considerations into mind, we devise the following adaptive strategy for $F$:

$$F_i = Gaussian(\frac{rank(i)}{NP}, 0.1) \tag{15}$$

where $rank(i)$ is the ranking of the personal best position ($\boldsymbol{pbest_i}$) of the $i$th particle after all ***pbest***s are sorted from the best to the worst. $F_i$ is the setting of the control parameter $\boldsymbol{F}$ of the $i$th particle. $NP$ represents the number of particles in the swarm. $Gaussian$ $(\frac{rank(i)}{NP}, 0.1)$ randomly samples a real random number according to the Gaussian distribution with the mean value set as $\frac{rank(i)}{NP}$ and the standard deviation set as 0.1. Here, it should be mentioned that the Gaussian distribution with a small variance (0.1) is utilized because such a Gaussian distribution has a narrow sampling range and thus could generate diverse values around the mean value. This offers slight diversity in the exemplar construction for each particle without damaging the construction efficiency.

From Equation (15), we can see that the control parameter $F_i$ for each particle is randomly generated by the Gaussian distribution with the mean value set as the division between the rank of its personal best position and the population size *NP* and the variance set as a small value, namely 0.1. Such an adaptive strategy brings the following benefits to the proposed PCLPSO:

(1) Different particles have different settings of $F_i$. Because of the rank of the personal best position of each particle being different from each other, the mean value of the Gaussian distribution is different for different particles and thus, for different particles, the learning step $F_i$ is different during the exemplar construction. This is beneficial for further improving the learning diversity of particles and thus profitable for assisting the swarm to get out of local basins.

(2) Particles with better ***pbest***s preserve small $F_i$, while those with worse ***pbest***s have large $F_i$ during the exemplar construction. Specifically, better ***pbest***s usually have small ranks, and thus the mean values of the Gaussian distribution are small. Therefore, the learning step $F_i$ for particles with better ***pbest***s is expectedly small. This just matches the expectation that the constructed exemplars for those particles with better ***pbest***s should not be too close to the randomly selected better positions. This is because the learning range of those better ***pbest***s is narrow due to the small number of learning candidates. By contrast, worse ***pbest***s usually have large ranks, leading to the mean value of the Gaussian distribution being large. Therefore, the learning step $F_i$ is expectedly large during the construction of guiding exemplars for those particles with worse ***pbest***s. This also matches the expectation that particles with worse ***pbest***s should learn more from better ones to accelerate their moving to promising areas.

(3) As a whole, we can see that the devised adaptive scheme for ***F*** could implicitly help PCLPSO compromise the search diversity and the search effectiveness of particles well to search the solution space properly to obtain high-accuracy solutions.

Experiments conducted in Section 4.3 validate the usefulness of the designed adaptive strategy for ***F*** in helping PCLPSO achieve good performance.

### 3.2.2. Dynamic Acceleration Coefficient *c* Strategy

As for the acceleration coefficient ***c***, instead of using fixed values in the literature [55,56,59,67,69], we develop the following dynamic strategy to generate different values for different particles

$$c_i = Cauchy(1.6, 0.2) \tag{16}$$

where *Cauchy*(1.6, 0.2) generates a real number based the Cauchy distribution with the position factor set as 1.6 and the scaling factor set as 0.2. $c_i$ is the setting of the acceleration coefficient of the *i*th particle. It deserves attention that instead of using the Gaussian distribution, the Cauchy distribution is employed here because the Cauchy distribution has a long fat tail and thus can generate more diversified values than the Gaussian distribution. Moreover, we set the position parameter and the scaling factor of the Cauchy distribution as 1.6 and 0.2, respectively, because in the literature [55,56,59,67,69], researchers have investigated that the setting of the acceleration coefficient ***c*** for PSOs is usually in the range of [1.0, 2.2] and with these parameter settings, the Cauchy distribution could generate diversified value in such a range.

With this dynamic strategy, different particles have different settings of $c_i$ and the difference among the values of different particles is relatively large. This is beneficial for further improving the learning diversity of particles, which is very valuable in solving complicated optimization problems with many local basins.

Experiments conducted in Section 4.3 demonstrate the usefulness of the designed dynamic strategy for ***c*** in assisting PCLPSO to obtain promising optimization performance.

### *3.3. Complete Procedure of PCLPSO*

Integrating the abovementioned techniques together, we develop the complete PCLPSO with the overall procedure presented in Algorithm 1. Specifically, the algorithm first randomly initializes $NP$ particles and then evaluates their fitness as shown in Line 1. After the initialization, the algorithm proceeds to the main loops of the optimization (Lines 2~18). Before the update of the swarm (Lines 5~17), $\boldsymbol{pbest}$s of all particles are first sorted from the best to the worst (Line 3) and then the inertia weight $\omega$ is computed based on Equation (14) (Line 4). Then, for each particle, the control parameter $F_i$ is first calculated according to Equation (15) (Line 6) and then a promising guiding exemplar is constructed (Lines 7~12). After the construction of the guiding exemplar, the acceleration coefficient $c_i$ is sampled from the Cauchy distribution (Line 13) and then the particle is updated (Line 14). Subsequently, the updated particle is reevaluated (Line 15) and its $\boldsymbol{pbest}$ is updated accordingly (Line 16). The main loop (Lines 2~18) continuously iterates until the termination condition is satisfied. Finally, when the algorithm terminates, the best solution among all $\boldsymbol{pbest}$s is obtained as the final output (Line 19).

From Algorithm 1, we can that except for the time used for evaluating the fitness of particles, at each generation, it takes $O(NP\times\log_2 NP)$ to sort all $\boldsymbol{pbest}$s, $O(NP)$ to select random better $\boldsymbol{pbest}$s for all particles, and O($NP\times D$) to construct new guiding exemplars for all particles. Then, it takes $O(NP\times D)$ to update all particles and $O(NP\times D)$ to update $\boldsymbol{pbest}$s. On the whole, the time complexity of PCLPSO is $O(NP\times D)$. With respect to the space complexity, the same as the classical PSO, PCLPSO needs $O(NP\times D)$, $O(NP\times D)$, and $O(NP\times D)$ to store the velocity vectors, the position vectors, and the personal best position vectors of all particles, respectively.

In summary, it is concluded that PCLPSO remains as efficient as the classical PSO regarding the time complexity and the space occupation.

**Algorithm 1: The Complete Procedure of PCLPSO**

```
Input: Population size $NP$, Total fitness evaluations $FE_{max}$
 1:  Randomly initialize $NP$ particles and compute their fitness, and $fes = NP$;
 2:  While ($fes \le FE_{max}$) do
 3:    Sort $\boldsymbol{pbest}$s from the best to the worst;
 4:    Calculate inertia weight $\omega$ based on Equation (14);
 5:    For $i$ = 1:$NP$ do
 6:      Calculate the control parameter $F_i$ according to Equation (15);
 7:      If $\boldsymbol{pbest_i}$ == $\boldsymbol{gbest}$ then
 8:        Use $gbest$ as the learning exemplar $e_i$;
 9:      Else
10:        Select a better $\boldsymbol{pbest_{rb}}$ randomly from those which are better than $\boldsymbol{pbest_i}$;
11:        Construct the learning exemplar $e_i$ according to Equation (11);
12:      End If
13:      Obtain the acceleration coefficient $c_i$ according to Equation (16);
14:      Update particle $x_i$ based on Equation (12) and Equation (13);
15:      Compute the fitness of $x_i$: $f(x_i)$, and $fes$++;
16:      Update its $\boldsymbol{pbest_i}$;
17:    End For
18:  End While
19:  Find the global best solution $\boldsymbol{gbest}$ among all $\boldsymbol{pbest}$s;
Output: $f(\boldsymbol{gbest})$ and $\boldsymbol{gbest}$
```

## 4. Numerical Analysis

This section mainly presents extensive experiments to comprehensively validate the effective optimization of PCLPSO. To be specific, Section 4.1 first briefly introduces the used benchmark functions and the compared methods. Then, extensive comparisons between PCLPSO and the compared approaches are displayed in Section 4.2. At last, to perform a deep analysis on the proposed algorithm, investigative experiments are executed to testify the influence of each component on the developed PCLPSO, so that readers have a better view on why the developed method could achieve good performance.

### *4.1. Experimental Settings*

In the experiments, we utilize the CEC 2017 benchmark problem set [73], which has been commonly used to validate evolutionary algorithms in the literature [75,85,86], to testify the optimization performance of the proposed PCLPSO. As shown in Table 1, there are 29 optimization problems including 2 unimodal problems, 7 simple multimodal problems, 10 hybrid problems, and 10 composition problems in this benchmark set. More detailed information can be found in [73]. It should be mentioned that to more visually understand the optimization results, for each particle, we utilize the error computed by the subtraction between its function value and the true global optimum as its fitness on each optimization problem.

**Table 1.** The summarized characteristics of the CEC 2017 benchmark problems.

| Category | $F$ | Functions | $F_i{}^* = F_i(x^*)$ |
|---|---|---|---|
| Unimodal Functions | $F_1$ | Shifted and Rotated Bent Cigar Function | 100 |
| | $F_3$ | Shifted and Rotated Zakharov Function | 300 |
| Simple Multimodal Functions | $F_4$ | Shifted and Rotated Rosenbrock’s Function | 400 |
| | $F_5$ | Shifted and Rotated Rastrigin’s Function | 500 |
| | $F_6$ | Shifted and Rotated Expanded Scaffer’s F6 Function | 600 |
| | $F_7$ | Shifted and Rotated Lunacek Bi_Rastrigin Function | 700 |
| | $F_8$ | Shifted and Rotated Non-Continuous Rastrigin’s Function | 800 |
| | $F_9$ | Shifted and Rotated Levy Function | 900 |
| | $F_{10}$ | Shifted and Rotated Schwefel’s Function | 1000 |
| Hybrid Functions | $F_{11}$ | Hybrid Function 1 ($N$ = 3) | 1100 |
| | $F_{12}$ | Hybrid Function 2 ($N$ = 3) | 1200 |
| | $F_{13}$ | Hybrid Function 3 ($N$ = 3) | 1300 |
| | $F_{14}$ | Hybrid Function 4 ($N$ = 4) | 1400 |
| | $F_{15}$ | Hybrid Function 5 ($N$ = 4) | 1500 |
| | $F_{16}$ | Hybrid Function 6 ($N$ = 4) | 1600 |
| | $F_{17}$ | Hybrid Function 6 ($N$ = 5) | 1700 |
| | $F_{18}$ | Hybrid Function 6 ($N$ = 5) | 1800 |
| | $F_{19}$ | Hybrid Function 6 ($N$ = 5) | 1900 |
| | $F_{20}$ | Hybrid Function 6 ($N$ = 6) | 2000 |
| Composition Functions | $F_{21}$ | Composition Function 1 ($N$ = 3) | 2100 |
| | $F_{22}$ | Composition Function 2 ($N$ = 3) | 2200 |
| | $F_{23}$ | Composition Function 3 ($N$ = 4) | 2300 |
| | $F_{24}$ | Composition Function 4 ($N$ = 4) | 2400 |
| | $F_{25}$ | Composition Function 5 ($N$ = 5) | 2500 |
| | $F_{26}$ | Composition Function 6 ($N$ = 5) | 2600 |
| | $F_{27}$ | Composition Function 7 ($N$ = 6) | 2700 |
| | $F_{28}$ | Composition Function 8 ($N$ = 6) | 2800 |
| | $F_{29}$ | Composition Function 9 ($N$ = 3) | 2900 |
| | $F_{30}$ | Composition Function 10 ($N$ = 3) | 3000 |
| Search Range: $[-100, 100]^{\mathrm{D}}$ | | | |

Firstly, to comprehensively testify the effective optimization ability of the proposed PCLPSO, seven state-of-the-art and representative PSOs were selected for comparison with it. Specifically, the selected seven algorithms are TCSPSO [59], AWPSO [84], PSO-DLP [67], GL-PSO [29], CLPSO [56], HCLPSO [62], and CLPSO-LS [69]. The former three algorithms are topology-based learning PSOs, while the latter four methods are exemplar construction-based learning PSOs.

Secondly, to comprehensively compare PCLPSO with the selected PSO variants, extensive comparison experiments were conducted on the CEC 2017 benchmark set with three dimensionality settings, namely 30-*D*, 50-*D*, and 100-*D*. To make fair comparisons, we set the total number of fitness evaluations (*FEmax*) as 10,000*$D$ for all algorithms.

Thirdly, for fairness, except for the swarm size, we set the key parameter settings of the selected seven algorithms as suggested in the related papers. With respect to the swarm size, since it is usually problem-dependent, we tuned its settings on the CEC 2017 benchmark set with three settings of the dimension size for all algorithms. After the preliminary experiments on fine-tuning, the settings of the swarm size along with the settings of the other key parameters of all algorithms are presented in Table 2.

**Table 2.** Parameter settings of PCLPSO and the seven compared PSO variants.

| Algorithms | *D* | Parameter Settings | |
|---|---|---|---|
| PCLPSO | 30 | $NP = 80$ | $\omega = 0.9\text{~}0.2$, $c_i = Cauchy(1.6, 0.2)$ |
| | 50 | $NP = 80$ | |
| | 100 | $NP = 150$ | |
| TCSPSO | 30 | $NP = 150$ | $\omega = 0.9\text{~}0.2$, $c_1 = c_2 = 2$ |
| | 50 | $NP = 150$ | |
| | 100 | $NP = 80$ | |
| AWPSO | 30 | $NP = 200$ | $\omega = 0.9\text{~}0.4,\ a = 0.000035\ m$, $b = 0.5$, $c = 0$, $d = 1.5$ |
| | 50 | $NP = 200$ | |
| | 100 | $NP = 200$ | |
| CLPSO-LS | 30 | $NP = 50$ | $\omega = 0.9\text{~}0.4,\ c = 1.49445,\ \beta = \frac{1}{3},\ \theta = 0.94$ |
| | 50 | $NP = 50$ | |
| | 100 | $NP = 50$ | |
| PSO-DLP | 30 | $NP = 200$ | $\omega = 0.9\text{~}0.3,\ c_1^s = c_2^s = c_1^m = c_2^m = 2.0$, $L = 50$ |
| | 50 | $NP = 200$ | |
| | 100 | $NP = 200$ | |
| GL-PSO | 30 | $NP = 100$ | $\omega = 0.7298$, $c = 1.49618$, $pm = 0.01$, $sg = 7$ |
| | 50 | $NP = 30$ | |
| | 100 | $NP = 50$ | |
| HCLPSO | 30 | $NP = 200$ | $\omega = 0.99\text{~}0.2$, $c_1 = 2.5\text{~}0.5$, $c_2 = 0.5\text{~}2.5$, $c = 3\text{~}1.5$ |
| | 50 | $NP = 160$ | |
| | 100 | $NP = 160$ | |
| CLPSO | 30 | $NP = 40$ | $\omega = 0.9\text{~}0.4$, $c = 1.49445$, $m = 7$ |

Fourthly, to comprehensively measure the optimization performance of each algorithm, the median, the mean, and the standard deviation (Std) values in terms of the fitness of the global best solutions found at the end of the associated algorithms over 30 independent runs were employed as the measurements. Furthermore, to tell whether there is a significant difference between the proposed PCLPSO and the compared methods, the Wilcoxon rank sum test at the significance level of $\alpha = 0.05$ was conducted to compare PCLPSO with each of the compared methods on each problem. To investigate the overall optimization performance of all algorithms on one whole benchmark set, we carried out the Friedman test at the significance level of $\alpha = 0.05$ to get the average rank of each algorithm. It should be mentioned that the above two tests were executed by directing using the API in Matlab.

At last, it should be noticed that all algorithms were coded under PyCharm CE and run on a server with Inter(R) Core (TM) i7-10700T CPU @ 2.90 GHz and 8G RAM.

### 4.2. Comparison with State-of-the-Art PSOs

Tables 3–5 exhibit the comparison results in terms of the global best fitness between PCLPSO and the seven selected PSO variants on the CEC 2017 benchmark sets with the three dimension sizes (30-*D*, 50-*D*, and 100-*D*), respectively. In the three tables, the bolded *p*-values denote that the proposed PCLPSO is significantly better than the compared algorithms on the associated problems. Besides, the symbol "+" above the *p*-values denotes that PCLPSO obtains significant superiority to the corresponding compared PSO variants on the related problems, the symbol "-" above the $p$-values denotes that PCLPSO obtains significant inferiority to the corresponding compared PSO variants on the associated problems, and the symbol "=" above the $p$-values denotes that PCLPSO obtains equivalent performance with the corresponding compared PSO variants on the associated problems. Furthermore, "+/=/−" in the three tables counts the numbers of "+", "=", and "−" in the whole benchmark set, respectively. Additionally, the average rank of each algorithm obtained from the Friedman test is also presented in the three tables. To clearly observe the statistical comparison results, Table 6 summarizes the statistical comparison results between PCLPSO and the seven compared peer PSO methods on the CEC 2017 benchmark set with the three different dimension sizes in terms of "+/=/−".

Observing Table 3, we summarize the comparison results between PCLPSO and the seven compared PSOs on the $30^{-D}$ CEC 2017 functions as follows:

(1) With respect to the Friedman test results, PCLPSO obtains the smallest rank, namely 1.97, and this rank value is much smaller than those of the seven compared algorithms (at least 2.66). This substantiates that PCLPSO achieves the best overall performance on the 30-$D$ CEC 2017 benchmark set and shows significant overall dominance to the seven compared PSO variants.

(2) From the perspective of the Wilcoxon rank sum test results, except for HCLPSO and CLPSO, PCLPSO significantly outperforms the seven compared PSO algorithms on at least 23 problems and shows worse performance on at most three problems. Compared with HCLPSO and CLPSO, PCLPSO shows significant superiority to them both on 17 problems and displays inferiority to them on at most nine problems. In particular, we find that PCLPSO shows significant superiority to PSO-DLP on all the 29 problems, and significantly outperforms AWPSO on 28 problems.

(3) In view of the comparison results on different kinds of optimization problems, on the two unimodal problems, PCLPSO dominates TCSPSO, AWPSO, and PSO-DLP on the two problems, and achieves competitive performance with the other four algorithms (CLPSO-LS, GL-PSO, HCLPSO, and CLPSO). On the seven simple multimodal problems, PCLPSO is significantly superior to PSO-DLP on all these problems and significantly dominates AWPSO and CLPSO-LS both on six problems. Competing with GL-PSO, PCLPSO shows significant dominance on five problems and displays no failure to GL-PSO. Compared with the other three algorithms, namely TCSPSO, HCLPSO, and CLPSO, PCLPSO achieves highly competitive performance with them. When it comes to the 10 hybrid problems, PCLPSO shows significant superiority to TCSPSO, AWPSO, CLPSO-LS, and PSO-DLP all on the 10 problems. Compared with GL-PSO, HCLPSO, and CLPSO, PCLPSO outperforms them significantly on nine, eight, and six problems, respectively. In terms of the 10 composition problems, PCLPSO significantly dominates AWPSO and PSO-DLP both on all these problems and outperforms GL-PSO on nine problems. In comparison with TCSPSO, CLPSO-LS, and CLPSO, PCLPSO shows significant superiority to them all on seven problems. Compared with HCLPSO, PCLPSO beats it on five problems and is defeated on only two problems.

**Table 3.** Global fitness comparisons between PCLPSO and the seven selected PSO methods on the CEC 2017 benchmark set with the dimensionality set as 30.

| *F* | Category | Quality | PCLPSO | TCSPSO | AWPSO | CLPSO-LS | PSO-DLP | GL-PSO | HCLPSO | CLPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $F_1$ | Unimodal Functions | Median | $6.17 \times 10^{2}$ | $2.08 \times 10^{3}$ | $7.16 \times 10^{9}$ | $1.42 \times 10^{4}$ | $2.20 \times 10^{9}$ | $2.10 \times 10^{3}$ | $5.49 \times 10^{3}$ | $7.62 \times 10^{2}$ |
| | | Mean | $1.33 \times 10^{3}$ | $3.78 \times 10^{3}$ | $7.79 \times 10^{9}$ | $1.67 \times 10^{4}$ | $3.17 \times 10^{9}$ | $2.34 \times 10^{4}$ | $9.72 \times 10^{3}$ | $3.88 \times 10^{2}$ |
| | | Std | $1.76 \times 10^{3}$ | $4.43 \times 10^{3}$ | $4.03 \times 10^{9}$ | $7.91 \times 10^{3}$ | $2.53 \times 10^{9}$ | $1.08 \times 10^{5}$ | $8.04 \times 10^{3}$ | $7.43 \times 10^{2}$ |
| | | *p*-value | - | $\mathbf{6.58 \times 10^{-3}}\,^{+}$ | $\mathbf{3.81 \times 10^{-15}}\,^{+}$ | $\mathbf{7.95 \times 10^{-15}}\,^{+}$ | $\mathbf{4.93 \times 10^{-9}}\,^{+}$ | $2.66 \times 10^{-1}\,^{=}$ | $\mathbf{6.46 \times 10^{-7}}\,^{+}$ | $9.21 \times 10^{-3}\,^{-}$ |
| $F_3$ | | Median | $1.78 \times 10^{3}$ | $2.23 \times 10^{4}$ | $1.39 \times 10^{4}$ | $6.80 \times 10^{-11}$ | $7.80 \times 10^{4}$ | $1.22 \times 10^{4}$ | $5.86 \times 10^{1}$ | $1.08 \times 10^{2}$ |
| | | Mean | $1.97 \times 10^{3}$ | $2.32 \times 10^{4}$ | $2.03 \times 10^{4}$ | $4.28 \times 10^{3}$ | $7.90 \times 10^{4}$ | $1.25 \times 10^{4}$ | $1.21 \times 10^{2}$ | $4.41 \times 10^{4}$ |
| | | Std | $8.25 \times 10^{2}$ | $6.36 \times 10^{3}$ | $1.62 \times 10^{4}$ | $1.64 \times 10^{4}$ | $1.14 \times 10^{4}$ | $6.13 \times 10^{3}$ | $1.50 \times 10^{2}$ | $1.02 \times 10^{4}$ |
| | | *p*-value | - | $\mathbf{1.42 \times 10^{-25}}\,^{+}$ | $\mathbf{7.13 \times 10^{-8}}\,^{+}$ | $4.45 \times 10^{-1}\,^{=}$ | $\mathbf{4.33 \times 10^{-42}}\,^{+}$ | $\mathbf{4.21 \times 10^{-13}}\,^{+}$ | $1.66 \times 10^{-17}\,^{-}$ | $\mathbf{1.78 \times 10^{-30}}\,^{+}$ |
| $F_{1\text{-}3}$ | +/=/− | | - | 2/0/0 | 2/0/0 | 1/1/0 | 2/0/0 | 1/1/0 | 1/0/1 | 1/0/1 |
| $F_4$ | Simple Multimodal Functions | Median | $1.56 \times 10^{2}$ | $1.29 \times 10^{2}$ | $6.71 \times 10^{2}$ | $8.90 \times 10^{1}$ | $1.10 \times 10^{3}$ | $2.10 \times 10^{2}$ | $8.56 \times 10^{1}$ | $1.72 \times 10^{2}$ |
| | | Mean | $1.55 \times 10^{2}$ | $1.28 \times 10^{2}$ | $7.12 \times 10^{2}$ | $8.90 \times 10^{1}$ | $1.26 \times 10^{3}$ | $2.15 \times 10^{2}$ | $8.66 \times 10^{1}$ | $9.12 \times 10^{1}$ |
| | | Std | $1.21 \times 10^{1}$ | $5.63 \times 10^{1}$ | $3.40 \times 10^{2}$ | $4.13 \times 10^{-1}$ | $7.70 \times 10^{2}$ | $5.81 \times 10^{1}$ | $8.78 \times 10^{0}$ | $1.59 \times 10^{0}$ |
| | | *p*-value | - | $1.17 \times 10^{-2}\,^{-}$ | $\mathbf{1.48 \times 10^{-12}}\,^{+}$ | $8.80 \times 10^{-37}\,^{-}$ | $\mathbf{9.96 \times 10^{-11}}\,^{+}$ | $\mathbf{8.58 \times 10^{-7}}\,^{+}$ | $1.01 \times 10^{-32}\,^{-}$ | $8.45 \times 10^{-36}\,^{-}$ |
| $F_5$ | | Median | $6.43 \times 10^{1}$ | $7.26 \times 10^{1}$ | $1.47 \times 10^{2}$ | $2.17 \times 10^{2}$ | $3.67 \times 10^{2}$ | $6.50 \times 10^{1}$ | $6.31 \times 10^{1}$ | $2.39 \times 10^{1}$ |
| | | Mean | $5.04 \times 10^{1}$ | $7.16 \times 10^{1}$ | $1.44 \times 10^{2}$ | $2.18 \times 10^{2}$ | $3.59 \times 10^{2}$ | $7.33 \times 10^{1}$ | $6.72 \times 10^{1}$ | $7.52 \times 10^{1}$ |
| | | Std | $2.51 \times 10^{1}$ | $2.19 \times 10^{1}$ | $3.16 \times 10^{1}$ | $1.23 \times 10^{1}$ | $2.66 \times 10^{1}$ | $3.00 \times 10^{1}$ | $1.66 \times 10^{1}$ | $6.81 \times 10^{0}$ |
| | | *p*-value | - | $\mathbf{9.29 \times 10^{-4}}\,^{+}$ | $\mathbf{1.83 \times 10^{-18}}\,^{+}$ | $\mathbf{3.61 \times 10^{-39}}\,^{+}$ | $\mathbf{1.82 \times 10^{-47}}\,^{+}$ | $\mathbf{2.25 \times 10^{-3}}\,^{+}$ | $\mathbf{3.41 \times 10^{-3}}\,^{+}$ | $\mathbf{2.54 \times 10^{-6}}\,^{+}$ |
| $F_6$ | | Median | $5.20 \times 10^{-2}$ | $1.69 \times 10^{2}$ | $1.39 \times 10^{1}$ | $4.32 \times 10^{-1}$ | $7.45 \times 10^{1}$ | $1.24 \times 10^{0}$ | $3.36 \times 10^{-4}$ | $3.87 \times 10^{-1}$ |
| | | Mean | $6.69 \times 10^{-2}$ | $6.97 \times 10^{2}$ | $1.58 \times 10^{1}$ | $9.35 \times 10^{-1}$ | $7.44 \times 10^{1}$ | $1.49 \times 10^{0}$ | $1.11 \times 10^{-3}$ | $2.86 \times 10^{-6}$ |
| | | Std | $6.30 \times 10^{-2}$ | $1.40 \times 10^{1}$ | $6.23 \times 10^{0}$ | $1.06 \times 10^{0}$ | $4.69 \times 10^{0}$ | $9.37 \times 10^{-1}$ | $1.62 \times 10^{-3}$ | $1.94 \times 10^{-6}$ |
| | | *p*-value | - | $9.23 \times 10^{-1}\,^{=}$ | $\mathbf{5.65 \times 10^{-20}}\,^{+}$ | $\mathbf{3.70 \times 10^{-5}}\,^{+}$ | $\mathbf{4.32 \times 10^{-63}}\,^{+}$ | $\mathbf{2.10 \times 10^{-11}}\,^{+}$ | $3.90 \times 10^{-7}\,^{-}$ | $2.69 \times 10^{-7}\,^{-}$ |
| $F_7$ | | Median | $1.10 \times 10^{2}$ | $1.18 \times 10^{2}$ | $1.66 \times 10^{2}$ | $2.37 \times 10^{2}$ | $5.73 \times 10^{2}$ | $1.02 \times 10^{2}$ | $9.32 \times 10^{1}$ | $7.11 \times 10^{1}$ |
| | | Mean | $1.10 \times 10^{2}$ | $1.24 \times 10^{2}$ | $1.91 \times 10^{2}$ | $2.33 \times 10^{2}$ | $5.75 \times 10^{2}$ | $1.00 \times 10^{2}$ | $9.38 \times 10^{1}$ | $9.05 \times 10^{1}$ |
| | | Std | $1.63 \times 10^{1}$ | $2.07 \times 10^{1}$ | $9.51 \times 10^{1}$ | $1.92 \times 10^{1}$ | $6.30 \times 10^{1}$ | $2.11 \times 10^{1}$ | $1.95 \times 10^{1}$ | $8.02 \times 10^{0}$ |
| | | *p*-value | - | $\mathbf{4.83 \times 10^{-3}}\,^{+}$ | $\mathbf{2.41 \times 10^{-5}}\,^{+}$ | $\mathbf{2.28 \times 10^{-34}}\,^{+}$ | $\mathbf{2.09 \times 10^{-43}}\,^{+}$ | $5.37 \times 10^{-2}\,^{=}$ | $1.20 \times 10^{-3}\,^{-}$ | $3.21 \times 10^{-7}\,^{-}$ |
| $F_8$ | | Median | $5.97 \times 10^{1}$ | $7.32 \times 10^{1}$ | $1.37 \times 10^{2}$ | $2.25 \times 10^{2}$ | $3.39 \times 10^{2}$ | $6.17 \times 10^{1}$ | $6.47 \times 10^{1}$ | $1.99 \times 10^{1}$ |
| | | Mean | $4.91 \times 10^{1}$ | $7.25 \times 10^{1}$ | $1.36 \times 10^{2}$ | $2.22 \times 10^{2}$ | $3.32 \times 10^{2}$ | $6.39 \times 10^{1}$ | $6.49 \times 10^{1}$ | $8.17 \times 10^{1}$ |
| | | Std | $2.72 \times 10^{1}$ | $1.91 \times 10^{1}$ | $3.38 \times 10^{1}$ | $1.17 \times 10^{1}$ | $2.27 \times 10^{1}$ | $1.93 \times 10^{1}$ | $1.87 \times 10^{1}$ | $1.00 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{2.85 \times 10^{-4}}\,^{+}$ | $\mathbf{8.32 \times 10^{-16}}\,^{+}$ | $\mathbf{1.40 \times 10^{-38}}\,^{+}$ | $\mathbf{4.40 \times 10^{-46}}\,^{+}$ | $\mathbf{1.83 \times 10^{-2}}\,^{+}$ | $\mathbf{1.14 \times 10^{-2}}\,^{+}$ | $\mathbf{7.28 \times 10^{-8}}\,^{+}$ |
| $F_9$ | | Median | $5.99 \times 10^{0}$ | $2.86 \times 10^{2}$ | $2.19 \times 10^{3}$ | $1.90 \times 10^{1}$ | $1.17 \times 10^{4}$ | $4.57 \times 10^{1}$ | $5.00 \times 10^{1}$ | $9.31 \times 10^{0}$ |
| | | Mean | $6.10 \times 10^{0}$ | $3.87 \times 10^{2}$ | $2.59 \times 10^{3}$ | $2.29 \times 10^{1}$ | $1.13 \times 10^{4}$ | $8.06 \times 10^{1}$ | $8.07 \times 10^{1}$ | $6.76 \times 10^{2}$ |
| | | Std | $4.09 \times 10^{0}$ | $4.05 \times 10^{2}$ | $1.70 \times 10^{3}$ | $2.78 \times 10^{1}$ | $1.30 \times 10^{3}$ | $1.15 \times 10^{2}$ | $1.43 \times 10^{2}$ | $2.85 \times 10^{2}$ |
| | | *p*-value | - | $\mathbf{3.28 \times 10^{-6}}\,^{+}$ | $\mathbf{1.78 \times 10^{-11}}\,^{+}$ | $\mathbf{1.83 \times 10^{-3}}\,^{+}$ | $\mathbf{3.86 \times 10^{-48}}\,^{+}$ | $\mathbf{7.85 \times 10^{-4}}\,^{+}$ | $\mathbf{6.02 \times 10^{-3}}\,^{+}$ | $\mathbf{1.12 \times 10^{-18}}\,^{+}$ |
| $F_{10}$ | | Median | $5.79 \times 10^{3}$ | $2.97 \times 10^{3}$ | $3.91 \times 10^{3}$ | $6.42 \times 10^{3}$ | $7.27 \times 10^{3}$ | $5.81 \times 10^{3}$ | $2.83 \times 10^{3}$ | $5.68 \times 10^{3}$ |
| | | Mean | $5.83 \times 10^{3}$ | $3.04 \times 10^{3}$ | $3.88 \times 10^{3}$ | $6.26 \times 10^{3}$ | $7.21 \times 10^{3}$ | $5.58 \times 10^{3}$ | $2.88 \times 10^{3}$ | $2.94 \times 10^{3}$ |
| | | Std | $3.68 \times 10^{2}$ | $7.37 \times 10^{2}$ | $7.23 \times 10^{2}$ | $6.47 \times 10^{2}$ | $2.65 \times 10^{2}$ | $1.26 \times 10^{3}$ | $5.25 \times 10^{2}$ | $2.82 \times 10^{2}$ |
| | | *p*-value | - | $4.55 \times 10^{-26}\,^{-}$ | $4.12 \times 10^{-19}\,^{-}$ | $\mathbf{2.55 \times 10^{-3}}\,^{+}$ | $\mathbf{9.14 \times 10^{-24}}\,^{+}$ | $2.91 \times 10^{-1}\,^{=}$ | $5.37 \times 10^{-33}\,^{-}$ | $4.10 \times 10^{-40}\,^{-}$ |
| $F_{4\text{-}10}$ | +/=/− | | - | 4/1/2 | 6/0/1 | 6/0/1 | 7/0/0 | 5/2/0 | 3/0/4 | 3/0/4 |

**Table 3.** *Cont.*

| *F* | Category | Quality | PCLPSO | TCSPSO | AWPSO | CLPSO-LS | PSO-DLP | GL-PSO | HCLPSO | CLPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $F_{11}$ | Hybrid Functions | Median | $5.04 \times 10^{1}$ | $8.89 \times 10^{1}$ | $4.20 \times 10^{2}$ | $1.82 \times 10^{2}$ | $2.67 \times 10^{3}$ | $1.34 \times 10^{2}$ | $6.80 \times 10^{1}$ | $7.91 \times 10^{1}$ |
| | | Mean | $6.04 \times 10^{1}$ | $9.24 \times 10^{1}$ | $4.84 \times 10^{2}$ | $1.81 \times 10^{2}$ | $2.64 \times 10^{3}$ | $1.55 \times 10^{2}$ | $8.31 \times 10^{1}$ | $1.16 \times 10^{2}$ |
| | | Std | $3.43 \times 10^{1}$ | $4.40 \times 10^{1}$ | $2.44 \times 10^{2}$ | $4.10 \times 10^{1}$ | $6.51 \times 10^{2}$ | $8.82 \times 10^{1}$ | $4.40 \times 10^{1}$ | $1.75 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{2.62 \times 10^{-3+}}$ | $\mathbf{2.66 \times 10^{-13+}}$ | $\mathbf{6.84 \times 10^{-18+}}$ | $\mathbf{1.73 \times 10^{-29+}}$ | $\mathbf{9.07 \times 10^{-7+}}$ | $\mathbf{2.96 \times 10^{-2+}}$ | $\mathbf{8.89 \times 10^{-11+}}$ |
| $F_{12}$ | | Median | $8.65 \times 10^{4}$ | $3.68 \times 10^{5}$ | $2.11 \times 10^{8}$ | $4.69 \times 10^{5}$ | $6.18 \times 10^{8}$ | $4.02 \times 10^{6}$ | $1.86 \times 10^{5}$ | $1.59 \times 10^{5}$ |
| | | Mean | $2.26 \times 10^{5}$ | $5.85 \times 10^{5}$ | $3.11 \times 10^{8}$ | $9.27 \times 10^{5}$ | $6.23 \times 10^{8}$ | $1.08 \times 10^{7}$ | $2.81 \times 10^{5}$ | $2.01 \times 10^{6}$ |
| | | Std | $4.63 \times 10^{5}$ | $7.80 \times 10^{5}$ | $3.57 \times 10^{8}$ | $8.92 \times 10^{5}$ | $3.20 \times 10^{8}$ | $2.90 \times 10^{7}$ | $2.64 \times 10^{5}$ | $1.12 \times 10^{6}$ |
| | | *p*-value | - | $\mathbf{3.47 \times 10^{-2+}}$ | $\mathbf{1.29 \times 10^{-5+}}$ | $\mathbf{3.29 \times 10^{-4+}}$ | $\mathbf{2.77 \times 10^{-15+}}$ | $5.04 \times 10^{-2=}$ | $5.80 \times 10^{-1=}$ | $\mathbf{5.05 \times 10^{-11+}}$ |
| $F_{13}$ | | Median | $1.35 \times 10^{3}$ | $1.34 \times 10^{4}$ | $4.57 \times 10^{6}$ | $6.55 \times 10^{3}$ | $9.07 \times 10^{7}$ | $1.07 \times 10^{4}$ | $4.02 \times 10^{4}$ | $1.08 \times 10^{3}$ |
| | | Mean | $2.01 \times 10^{3}$ | $1.73 \times 10^{4}$ | $1.83 \times 10^{7}$ | $1.74 \times 10^{4}$ | $8.54 \times 10^{7}$ | $6.13 \times 10^{4}$ | $3.36 \times 10^{4}$ | $3.40 \times 10^{3}$ |
| | | Std | $1.71 \times 10^{3}$ | $1.67 \times 10^{4}$ | $3.01 \times 10^{7}$ | $2.22 \times 10^{4}$ | $5.47 \times 10^{7}$ | $1.52 \times 10^{5}$ | $2.53 \times 10^{4}$ | $1.45 \times 10^{3}$ |
| | | *p*-value | - | $\mathbf{6.12 \times 10^{-6+}}$ | $\mathbf{1.52 \times 10^{-3+}}$ | $\mathbf{3.51 \times 10^{-4+}}$ | $\mathbf{7.28 \times 10^{-12+}}$ | $\mathbf{3.63 \times 10^{-2+}}$ | $\mathbf{5.76 \times 10^{-9+}}$ | $\mathbf{1.22 \times 10^{-3+}}$ |
| $F_{14}$ | | Median | $6.18 \times 10^{2}$ | $8.61 \times 10^{3}$ | $3.66 \times 10^{4}$ | $1.11 \times 10^{5}$ | $3.49 \times 10^{5}$ | $1.43 \times 10^{4}$ | $1.78 \times 10^{4}$ | $1.70 \times 10^{3}$ |
| | | Mean | $1.06 \times 10^{3}$ | $3.31 \times 10^{4}$ | $6.71 \times 10^{4}$ | $1.06 \times 10^{5}$ | $3.56 \times 10^{5}$ | $2.92 \times 10^{4}$ | $2.11 \times 10^{4}$ | $4.93 \times 10^{4}$ |
| | | Std | $9.76 \times 10^{2}$ | $7.29 \times 10^{4}$ | $6.16 \times 10^{4}$ | $5.41 \times 10^{4}$ | $1.88 \times 10^{5}$ | $4.70 \times 10^{4}$ | $2.04 \times 10^{4}$ | $3.46 \times 10^{4}$ |
| | | *p*-value | - | $\mathbf{1.92 \times 10^{-2+}}$ | $\mathbf{2.22 \times 10^{-7+}}$ | $\mathbf{2.66 \times 10^{-15+}}$ | $\mathbf{8.66 \times 10^{-15+}}$ | $\mathbf{1.76 \times 10^{-3+}}$ | $\mathbf{1.54 \times 10^{-6+}}$ | $\mathbf{2.66 \times 10^{-10+}}$ |
| $F_{15}$ | | Median | $1.17 \times 10^{3}$ | $9.37 \times 10^{3}$ | $6.85 \times 10^{4}$ | $4.13 \times 10^{4}$ | $8.31 \times 10^{6}$ | $3.86 \times 10^{3}$ | $1.13 \times 10^{4}$ | $5.20 \times 10^{2}$ |
| | | Mean | $1.63 \times 10^{3}$ | $1.30 \times 10^{4}$ | $1.18 \times 10^{5}$ | $3.60 \times 10^{4}$ | $1.00 \times 10^{7}$ | $6.31 \times 10^{3}$ | $1.51 \times 10^{4}$ | $4.59 \times 10^{2}$ |
| | | Std | $1.56 \times 10^{3}$ | $1.21 \times 10^{4}$ | $1.30 \times 10^{5}$ | $8.78 \times 10^{3}$ | $8.20 \times 10^{6}$ | $7.20 \times 10^{3}$ | $1.28 \times 10^{4}$ | $2.49 \times 10^{2}$ |
| | | *p*-value | - | $\mathbf{3.84 \times 10^{-6+}}$ | $\mathbf{7.74 \times 10^{-6+}}$ | $\mathbf{7.12 \times 10^{-29+}}$ | $\mathbf{9.52 \times 10^{-9+}}$ | $\mathbf{9.53 \times 10^{-4+}}$ | $\mathbf{4.57 \times 10^{-7+}}$ | $1.47 \times 10^{-4-}$ |
| $F_{16}$ | | Median | $5.61 \times 10^{2}$ | $8.54 \times 10^{2}$ | $1.04 \times 10^{3}$ | $1.29 \times 10^{3}$ | $2.52 \times 10^{3}$ | $1.16 \times 10^{3}$ | $6.92 \times 10^{2}$ | $5.07 \times 10^{2}$ |
| | | Mean | $5.22 \times 10^{2}$ | $8.26 \times 10^{2}$ | $1.08 \times 10^{3}$ | $1.14 \times 10^{3}$ | $2.50 \times 10^{3}$ | $1.11 \times 10^{3}$ | $6.74 \times 10^{2}$ | $6.24 \times 10^{2}$ |
| | | Std | $2.47 \times 10^{2}$ | $3.71 \times 10^{2}$ | $3.43 \times 10^{2}$ | $4.17 \times 10^{2}$ | $2.25 \times 10^{2}$ | $4.17 \times 10^{2}$ | $2.68 \times 10^{2}$ | $1.67 \times 10^{2}$ |
| | | *p*-value | - | $\mathbf{4.28 \times 10^{-4+}}$ | $\mathbf{1.17 \times 10^{-9+}}$ | $\mathbf{2.96 \times 10^{-9+}}$ | $\mathbf{7.86 \times 10^{-39+}}$ | $\mathbf{9.41 \times 10^{-9+}}$ | $\mathbf{2.65 \times 10^{-2+}}$ | $6.66 \times 10^{-2=}$ |
| $F_{17}$ | | Median | $6.94 \times 10^{1}$ | $2.90 \times 10^{2}$ | $5.78 \times 10^{2}$ | $6.85 \times 10^{2}$ | $1.02 \times 10^{3}$ | $3.35 \times 10^{2}$ | $2.98 \times 10^{2}$ | $8.12 \times 10^{1}$ |
| | | Mean | $8.91 \times 10^{1}$ | $2.96 \times 10^{2}$ | $5.58 \times 10^{2}$ | $9.36 \times 10^{2}$ | $9.90 \times 10^{2}$ | $3.25 \times 10^{2}$ | $2.97 \times 10^{2}$ | $1.88 \times 10^{2}$ |
| | | Std | $4.72 \times 10^{1}$ | $1.63 \times 10^{2}$ | $1.84 \times 10^{2}$ | $6.50 \times 10^{2}$ | $1.57 \times 10^{2}$ | $1.73 \times 10^{2}$ | $1.56 \times 10^{2}$ | $6.70 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{1.07 \times 10^{-8+}}$ | $\mathbf{1.40 \times 10^{-19+}}$ | $\mathbf{1.90 \times 10^{-9+}}$ | $\mathbf{4.13 \times 10^{-37+}}$ | $\mathbf{1.23 \times 10^{-9+}}$ | $\mathbf{3.22 \times 10^{-9+}}$ | $\mathbf{1.42 \times 10^{-8+}}$ |
| $F_{18}$ | | Median | $7.21 \times 10^{4}$ | $2.75 \times 10^{5}$ | $6.51 \times 10^{5}$ | $6.81 \times 10^{5}$ | $4.18 \times 10^{6}$ | $1.44 \times 10^{5}$ | $1.58 \times 10^{5}$ | $7.96 \times 10^{4}$ |
| | | Mean | $8.30 \times 10^{4}$ | $4.08 \times 10^{5}$ | $1.23 \times 10^{6}$ | $2.44 \times 10^{6}$ | $4.66 \times 10^{6}$ | $2.10 \times 10^{5}$ | $2.01 \times 10^{5}$ | $2.46 \times 10^{5}$ |
| | | Std | $4.04 \times 10^{4}$ | $3.70 \times 10^{5}$ | $2.05 \times 10^{6}$ | $4.19 \times 10^{6}$ | $2.40 \times 10^{6}$ | $2.01 \times 10^{5}$ | $1.26 \times 10^{5}$ | $1.58 \times 10^{5}$ |
| | | *p*-value | - | $\mathbf{1.23 \times 10^{-5+}}$ | $\mathbf{3.33 \times 10^{-3+}}$ | $\mathbf{3.20 \times 10^{-3+}}$ | $\mathbf{5.88 \times 10^{-15+}}$ | $\mathbf{1.24 \times 10^{-3+}}$ | $\mathbf{9.00 \times 10^{-6+}}$ | $\mathbf{1.02 \times 10^{-6+}}$ |
| $F_{19}$ | | Median | $3.71 \times 10^{3}$ | $5.01 \times 10^{3}$ | $3.81 \times 10^{5}$ | $3.51 \times 10^{4}$ | $2.58 \times 10^{7}$ | $5.94 \times 10^{3}$ | $1.07 \times 10^{4}$ | $4.35 \times 10^{3}$ |
| | | Mean | $4.24 \times 10^{3}$ | $1.19 \times 10^{4}$ | $7.01 \times 10^{6}$ | $3.51 \times 10^{4}$ | $2.31 \times 10^{7}$ | $9.63 \times 10^{3}$ | $1.77 \times 10^{4}$ | $1.35 \times 10^{2}$ |
| | | Std | $3.20 \times 10^{3}$ | $1.39 \times 10^{4}$ | $1.81 \times 10^{7}$ | $1.98 \times 10^{4}$ | $1.28 \times 10^{7}$ | $9.84 \times 10^{3}$ | $2.03 \times 10^{4}$ | $8.51 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{4.76 \times 10^{-3+}}$ | $\mathbf{3.86 \times 10^{-2+}}$ | $\mathbf{1.08 \times 10^{-11+}}$ | $\mathbf{4.67 \times 10^{-14+}}$ | $\mathbf{6.01 \times 10^{-3+}}$ | $\mathbf{6.74 \times 10^{-4+}}$ | $2.74 \times 10^{-9-}$ |

**Table 3.** *Cont.*

| $F$ | Category | Quality | PCLPSO | TCSPSO | AWPSO | CLPSO-LS | PSO-DLP | GL-PSO | HCLPSO | CLPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $F_{20}$ | Hybrid Functions | Median | $1.79 \times 10^{2}$ | $2.67 \times 10^{2}$ | $3.76 \times 10^{2}$ | $5.99 \times 10^{2}$ | $7.86 \times 10^{2}$ | $2.07 \times 10^{2}$ | $1.78 \times 10^{2}$ | $1.39 \times 10^{2}$ |
| | | Mean | $1.95 \times 10^{2}$ | $2.94 \times 10^{2}$ | $3.83 \times 10^{2}$ | $5.87 \times 10^{2}$ | $7.79 \times 10^{2}$ | $2.56 \times 10^{2}$ | $1.89 \times 10^{2}$ | $1.88 \times 10^{2}$ |
| | | Std | $8.82 \times 10^{1}$ | $1.67 \times 10^{2}$ | $1.30 \times 10^{2}$ | $1.57 \times 10^{2}$ | $9.66 \times 10^{1}$ | $1.32 \times 10^{2}$ | $1.20 \times 10^{2}$ | $6.68 \times 10^{1}$ |
| | | $p$-value | - | $\mathbf{5.36 \times 10^{-3}}{}^{+}$ | $\mathbf{1.37 \times 10^{-8}}{}^{+}$ | $\mathbf{3.19 \times 10^{-17}}{}^{+}$ | $\mathbf{2.83 \times 10^{-32}}{}^{+}$ | $\mathbf{3.92 \times 10^{-2}}{}^{+}$ | $8.32 \times 10^{-1}{}^{=}$ | $7.64 \times 10^{-1}{}^{=}$ |
| $F_{11\text{-}20}$ | +/=/− | | - | 10/0/0 | 10/0/0 | 10/0/0 | 10/0/0 | 9/1/0 | 8/2/0 | 6/2/2 |
| $F_{21}$ | Composition Functions | Median | $2.21 \times 10^{2}$ | $2.78 \times 10^{2}$ | $3.26 \times 10^{2}$ | $4.00 \times 10^{2}$ | $5.36 \times 10^{2}$ | $2.83 \times 10^{2}$ | $2.73 \times 10^{2}$ | $2.25 \times 10^{2}$ |
| | | Mean | $2.36 \times 10^{2}$ | $2.78 \times 10^{2}$ | $3.35 \times 10^{2}$ | $4.02 \times 10^{2}$ | $5.38 \times 10^{2}$ | $3.02 \times 10^{2}$ | $2.72 \times 10^{2}$ | $2.83 \times 10^{2}$ |
| | | Std | $2.43 \times 10^{1}$ | $1.98 \times 10^{1}$ | $3.09 \times 10^{1}$ | $7.91 \times 10^{0}$ | $2.07 \times 10^{1}$ | $5.42 \times 10^{1}$ | $1.56 \times 10^{1}$ | $2.58 \times 10^{1}$ |
| | | $p$-value | - | $\mathbf{8.89 \times 10^{-10}}{}^{+}$ | $\mathbf{6.25 \times 10^{-20}}{}^{+}$ | $\mathbf{4.39 \times 10^{-41}}{}^{+}$ | $\mathbf{3.11 \times 10^{-50}}{}^{+}$ | $\mathbf{1.02 \times 10^{-7}}{}^{+}$ | $\mathbf{4.30 \times 10^{-9}}{}^{+}$ | $\mathbf{7.62 \times 10^{-10}}{}^{+}$ |
| $F_{22}$ | | Median | $1.00 \times 10^{2}$ | $1.03 \times 10^{2}$ | $4.07 \times 10^{3}$ | $6.99 \times 10^{3}$ | $4.16 \times 10^{3}$ | $1.00 \times 10^{2}$ | $1.00 \times 10^{2}$ | $1.00 \times 10^{2}$ |
| | | Mean | $1.02 \times 10^{2}$ | $1.20 \times 10^{3}$ | $3.61 \times 10^{3}$ | $6.95 \times 10^{3}$ | $4.08 \times 10^{3}$ | $1.02 \times 10^{2}$ | $1.88 \times 10^{2}$ | $8.25 \times 10^{2}$ |
| | | Std | $1.94 \times 10^{0}$ | $1.63 \times 10^{3}$ | $1.47 \times 10^{3}$ | $3.16 \times 10^{2}$ | $5.78 \times 10^{2}$ | $3.77 \times 10^{0}$ | $4.81 \times 10^{2}$ | $1.20 \times 10^{3}$ |
| | | $p$-value | - | $\mathbf{4.92 \times 10^{-4}}{}^{+}$ | $\mathbf{6.07 \times 10^{-19}}{}^{+}$ | $\mathbf{5.53 \times 10^{-71}}{}^{+}$ | $\mathbf{1.74 \times 10^{-42}}{}^{+}$ | $3.24 \times 10^{-1}{}^{=}$ | $3.27 \times 10^{-1}{}^{=}$ | $\mathbf{1.68 \times 10^{-3}}{}^{+}$ |
| $F_{23}$ | | Median | $3.85 \times 10^{2}$ | $4.36 \times 10^{2}$ | $5.46 \times 10^{2}$ | $5.61 \times 10^{2}$ | $9.41 \times 10^{2}$ | $5.75 \times 10^{2}$ | $4.49 \times 10^{2}$ | $3.91 \times 10^{2}$ |
| | | Mean | $3.88 \times 10^{2}$ | $4.34 \times 10^{2}$ | $5.50 \times 10^{2}$ | $5.57 \times 10^{2}$ | $9.33 \times 10^{2}$ | $5.56 \times 10^{2}$ | $4.51 \times 10^{2}$ | $4.45 \times 10^{2}$ |
| | | Std | $1.67 \times 10^{1}$ | $2.71 \times 10^{1}$ | $4.98 \times 10^{1}$ | $1.42 \times 10^{1}$ | $3.80 \times 10^{1}$ | $4.79 \times 10^{1}$ | $2.01 \times 10^{1}$ | $1.10 \times 10^{1}$ |
| | | $p$-value | - | $\mathbf{8.35 \times 10^{-11}}{}^{+}$ | $\mathbf{4.33 \times 10^{-24}}{}^{+}$ | $\mathbf{2.13 \times 10^{-45}}{}^{+}$ | $\mathbf{2.13 \times 10^{-58}}{}^{+}$ | $\mathbf{1.30 \times 10^{-25}}{}^{+}$ | $\mathbf{3.45 \times 10^{-19}}{}^{+}$ | $\mathbf{1.11 \times 1^{-22}}{}^{+}$ |
| $F_{24}$ | | Median | $4.50 \times 10^{2}$ | $5.02 \times 10^{2}$ | $6.29 \times 10^{2}$ | $6.26 \times 10^{2}$ | $9.61 \times 10^{2}$ | $6.44 \times 10^{2}$ | $5.38 \times 10^{2}$ | $4.61 \times 10^{2}$ |
| | | Mean | $4.53 \times 10^{2}$ | $5.19 \times 10^{2}$ | $6.37 \times 10^{2}$ | $6.20 \times 10^{2}$ | $9.64 \times 10^{2}$ | $6.36 \times 10^{2}$ | $5.41 \times 10^{2}$ | $5.60 \times 10^{2}$ |
| | | Std | $1.80 \times 10^{1}$ | $4.20 \times 10^{1}$ | $3.88 \times 10^{1}$ | $1.04 \times 10^{1}$ | $6.49 \times 10^{1}$ | $3.39 \times 10^{1}$ | $2.50 \times 10^{1}$ | $1.69 \times 10^{1}$ |
| | | $p$-value | - | $\mathbf{1.04 \times 10^{-10}}{}^{+}$ | $\mathbf{2.66 \times 10^{-31}}{}^{+}$ | $\mathbf{2.95 \times 10^{-46}}{}^{+}$ | $\mathbf{7.59 \times 10^{-45}}{}^{+}$ | $\mathbf{9.03 \times 10^{-34}}{}^{+}$ | $\mathbf{2.16 \times 10^{-22}}{}^{+}$ | $\mathbf{1.96 \times 10^{-31}}{}^{+}$ |
| $F_{25}$ | | Median | $4.09 \times 10^{2}$ | $4.03 \times 10^{2}$ | $6.22 \times 10^{2}$ | $3.88 \times 10^{2}$ | $7.79 \times 10^{2}$ | $4.31 \times 10^{2}$ | $3.89 \times 10^{2}$ | $4.03 \times 10^{2}$ |
| | | Mean | $4.10 \times 10^{2}$ | $4.12 \times 10^{2}$ | $6.56 \times 10^{2}$ | $3.88 \times 10^{2}$ | $8.21 \times 10^{2}$ | $4.31 \times 10^{2}$ | $3.89 \times 10^{2}$ | $3.88 \times 10^{2}$ |
| | | Std | $1.42 \times 10^{1}$ | $2.08 \times 10^{1}$ | $1.89 \times 10^{2}$ | $4.34 \times 10^{-1}$ | $1.74 \times 10^{2}$ | $2.41 \times 10^{1}$ | $6.70 \times 10^{0}$ | $6.07 \times 10^{-1}$ |
| | | $p$-value | - | $7.26 \times 10^{-1}{}^{=}$ | $\mathbf{1.97 \times 10^{-9}}{}^{+}$ | $3.87 \times 10^{-12}{}^{-}$ | $\mathbf{1.14 \times 10^{-18}}{}^{+}$ | $\mathbf{1.32 \times 10^{-4}}{}^{+}$ | $7.48 \times 10^{-10}{}^{-}$ | $9.15 \times 10^{-12}{}^{-}$ |
| $F_{26}$ | | Median | $1.27 \times 10^{3}$ | $2.08 \times 10^{3}$ | $2.98 \times 10^{3}$ | $3.16 \times 10^{3}$ | $5.79 \times 10^{3}$ | $2.85 \times 10^{3}$ | $1.93 \times 10^{3}$ | $1.44 \times 10^{3}$ |
| | | Mean | $1.27 \times 10^{3}$ | $1.86 \times 10^{3}$ | $3.06 \times 10^{3}$ | $3.14 \times 10^{3}$ | $5.69 \times 10^{3}$ | $2.67 \times 10^{3}$ | $1.77 \times 10^{3}$ | $1.58 \times 10^{3}$ |
| | | Std | $6.53 \times 10^{1}$ | $6.08 \times 10^{2}$ | $6.43 \times 10^{2}$ | $1.10 \times 10^{2}$ | $7.96 \times 10^{2}$ | $9.70 \times 10^{2}$ | $6.17 \times 10^{2}$ | $4.77 \times 10^{2}$ |
| | | $p$-value | - | $\mathbf{2.16 \times 10^{-6}}{}^{+}$ | $\mathbf{8.97 \times 10^{-22}}{}^{+}$ | $\mathbf{4.08 \times 10^{-61}}{}^{+}$ | $\mathbf{3.06 \times 10^{-37}}{}^{+}$ | $\mathbf{1.09 \times 10^{-10}}{}^{+}$ | $\mathbf{4.98 \times 10^{-5}}{}^{+}$ | $\mathbf{7.88 \times 10^{-4}}{}^{+}$ |
| $F_{27}$ | | Median | $5.45 \times 10^{2}$ | $5.74 \times 10^{2}$ | $5.68 \times 10^{2}$ | $5.19 \times 10^{2}$ | $1.04 \times 10^{3}$ | $5.77 \times 10^{2}$ | $5.13 \times 10^{2}$ | $5.52 \times 10^{2}$ |
| | | Mean | $5.50 \times 10^{2}$ | $5.76 \times 10^{2}$ | $5.81 \times 10^{2}$ | $5.22 \times 10^{2}$ | $1.04 \times 10^{3}$ | $5.80 \times 10^{2}$ | $5.15 \times 10^{2}$ | $5.10 \times 10^{2}$ |
| | | Std | $1.45 \times 10^{1}$ | $2.61 \times 10^{1}$ | $4.60 \times 10^{1}$ | $1.55 \times 10^{1}$ | $7.32 \times 10^{1}$ | $1.96 \times 10^{1}$ | $1.43 \times 10^{1}$ | $4.60 \times 10^{0}$ |
| | | $p$-value | - | $\mathbf{1.05 \times 10^{-5}}{}^{+}$ | $\mathbf{8.04 \times 10^{-4}}{}^{+}$ | $7.41 \times 10^{-10}{}^{-}$ | $\mathbf{3.18 \times 10^{-41}}{}^{+}$ | $\mathbf{1.35 \times 10^{-8}}{}^{+}$ | $2.71 \times 10^{-13}{}^{-}$ | $5.03 \times 10^{-21}{}^{-}$ |
| $F_{28}$ | | Median | $4.45 \times 10^{2}$ | $4.49 \times 10^{2}$ | $7.85 \times 10^{2}$ | $3.51 \times 10^{3}$ | $1.15 \times 10^{3}$ | $5.06 \times 10^{2}$ | $4.45 \times 10^{2}$ | $4.73 \times 10^{2}$ |
| | | Mean | $4.50 \times 10^{2}$ | $4.61 \times 10^{2}$ | $9.64 \times 10^{2}$ | $3.07 \times 10^{3}$ | $1.24 \times 10^{3}$ | $5.20 \times 10^{2}$ | $4.49 \times 10^{2}$ | $4.83 \times 10^{2}$ |
| | | Std | $3.61 \times 10^{1}$ | $4.81 \times 10^{1}$ | $4.10 \times 10^{2}$ | $9.21 \times 10^{2}$ | $3.87 \times 10^{2}$ | $6.10 \times 10^{1}$ | $3.83 \times 10^{1}$ | $2.94 \times 10^{1}$ |
| | | $p$-value | - | $3.32 \times 10^{-1}{}^{=}$ | $\mathbf{5.56 \times 10^{-9}}{}^{+}$ | $\mathbf{2.33 \times 10^{-22}}{}^{+}$ | $\mathbf{5.37 \times 10^{-16}}{}^{+}$ | $\mathbf{1.29 \times 10^{-6}}{}^{+}$ | $8.75 \times 10^{-1}{}^{=}$ | $\mathbf{2.97 \times 10^{-4}}{}^{+}$ |

**Table 3.** *Cont.*

| *F* | Category | Quality | PCLPSO | TCSPSO | AWPSO | CLPSO-LS | PSO-DLP | GL-PSO | HCLPSO | CLPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $F_{29}$ | Composition Functions | Median | $5.12 \times 10^{2}$ | $7.99 \times 10^{2}$ | $9.78 \times 10^{2}$ | $9.33 \times 10^{2}$ | $2.41 \times 10^{3}$ | $7.81 \times 10^{2}$ | $6.50 \times 10^{2}$ | $5.30 \times 10^{2}$ |
| | | Mean | $5.16 \times 10^{2}$ | $8.33 \times 10^{2}$ | $1.02 \times 10^{3}$ | $1.11 \times 10^{3}$ | $2.35 \times 10^{3}$ | $8.29 \times 10^{2}$ | $7.08 \times 10^{2}$ | $6.46 \times 10^{2}$ |
| | | Std | $3.50 \times 10^{1}$ | $2.17 \times 10^{2}$ | $2.75 \times 10^{2}$ | $6.12 \times 10^{2}$ | $2.17 \times 10^{2}$ | $1.73 \times 10^{2}$ | $2.04 \times 10^{2}$ | $7.33 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{9.37 \times 10^{-11}}^{+}$ | $\mathbf{3.97 \times 10^{-14}}^{+}$ | $\mathbf{1.67 \times 10^{-6}}^{+}$ | $\mathbf{4.36 \times 10^{-47}}^{+}$ | $\mathbf{9.05 \times 10^{-14}}^{+}$ | $\mathbf{4.40 \times 10^{-6}}^{+}$ | $\mathbf{3.26 \times 10^{-12}}^{+}$ |
| $F_{30}$ | | Median | $9.18 \times 10^{3}$ | $1.21 \times 10^{4}$ | $2.19 \times 10^{6}$ | $1.43 \times 10^{4}$ | $4.35 \times 10^{7}$ | $3.78 \times 10^{4}$ | $6.27 \times 10^{3}$ | $1.30 \times 10^{4}$ |
| | | Mean | $1.11 \times 10^{4}$ | $2.26 \times 10^{4}$ | $7.75 \times 10^{6}$ | $1.38 \times 10^{4}$ | $4.69 \times 10^{7}$ | $6.87 \times 10^{4}$ | $8.22 \times 10^{3}$ | $1.37 \times 10^{4}$ |
| | | Std | $1.02 \times 10^{4}$ | $3.46 \times 10^{4}$ | $1.52 \times 10^{7}$ | $1.54 \times 10^{3}$ | $2.39 \times 10^{7}$ | $8.22 \times 10^{4}$ | $5.11 \times 10^{3}$ | $4.06 \times 10^{3}$ |
| | | *p*-value | - | $8.61 \times 10^{-2\,=}$ | $\mathbf{7.23 \times 10^{-3}}^{+}$ | $1.54 \times 10^{-1\,=}$ | $\mathbf{1.86 \times 10^{-15}}^{+}$ | $\mathbf{3.42 \times 10^{-4}}^{+}$ | $1.72 \times 10^{-1\,=}$ | $1.96 \times 10^{-1\,=}$ |
| $F_{21\text{-}30}$ | +/=/− | | - | 7/3/0 | 10/0/0 | 7/1/2 | 10/0/0 | 9/1/0 | 5/3/2 | 7/1/2 |
| | +/=/− | | - | 23/4/2 | 28/0/1 | 24/2/3 | 29/0/0 | 24/5/0 | 17/5/7 | 17/3/9 |
| | *Rank* | | **1.97** | 3.93 | 6.31 | 5.62 | 7.90 | 4.62 | 2.66 | 3.00 |

**Table 4.** Global fitness comparisons between PCLPSO and the seven selected PSO methods on the CEC 2017 benchmark set with the dimensionality set as 50.

| *F* | Category | Quality | PCLPSO | TCSPSO | AWPSO | CLPSO-LS | PSO-DLP | GL-PSO | HCLPSO | CLPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $F_{1}$ | Unimodal Functions | Median | $4.39 \times 10^{3}$ | $1.69 \times 10^{3}$ | $7.16 \times 10^{9}$ | $4.59 \times 10^{7}$ | $5.96 \times 10^{9}$ | $3.12 \times 10^{3}$ | $3.72 \times 10^{3}$ | $2.06 \times 10^{3}$ |
| | | Mean | $4.19 \times 10^{3}$ | $4.35 \times 10^{3}$ | $3.92 \times 10^{0}$ | $1.29 \times 10^{8}$ | $8.04 \times 10^{9}$ | $1.67 \times 10^{7}$ | $4.06 \times 10^{3}$ | $2.59 \times 10^{3}$ |
| | | Std | $2.12 \times 10^{3}$ | $5.68 \times 10^{3}$ | $1.16 \times 10^{0}$ | $3.39 \times 10^{8}$ | $6.22 \times 10^{9}$ | $6.58 \times 10^{7}$ | $1.86 \times 10^{3}$ | $1.91 \times 10^{3}$ |
| | | *p*-value | - | $8.85 \times 10^{-1\,=}$ | $\mathbf{3.92 \times 10^{-26}}^{+}$ | $\mathbf{4.08 \times 10^{-2}}^{+}$ | $\mathbf{2.11 \times 10^{-9}}^{+}$ | $\mathbf{2.96 \times 10^{-2}}^{+}$ | $3.38 \times 10^{-6\,-}$ | $3.80 \times 10^{-3\,-}$ |
| $F_{3}$ | | Median | $1.63 \times 10^{4}$ | $7.75 \times 10^{4}$ | $1.39 \times 10^{4}$ | $5.20 \times 10^{-10}$ | $1.85 \times 10^{5}$ | $1.97 \times 10^{4}$ | $1.76 \times 10^{2}$ | $1.31 \times 10^{5}$ |
| | | Mean | $1.64 \times 10^{4}$ | $7.86 \times 10^{4}$ | $9.74 \times 10^{4}$ | $2.37 \times 10^{4}$ | $1.82 \times 10^{5}$ | $2.25 \times 10^{4}$ | $1.62 \times 10^{2}$ | $1.31 \times 10^{5}$ |
| | | Std | $4.17 \times 10^{3}$ | $1.15 \times 10^{4}$ | $3.70 \times 10^{4}$ | $6.17 \times 10^{4}$ | $2.03 \times 10^{4}$ | $1.49 \times 10^{4}$ | $4.43 \times 10^{1}$ | $2.25 \times 10^{4}$ |
| | | *p*-value | - | $\mathbf{8.24 \times 10^{-36}}^{+}$ | $\mathbf{2.69 \times 10^{-17}}^{+}$ | $5.17 \times 10^{-1\,=}$ | $\mathbf{2.32 \times 10^{-46}}^{+}$ | $\mathbf{4.54 \times 10^{-3}}^{+}$ | $2.61 \times 10^{-26\,-}$ | $\mathbf{5.35 \times 10^{-35}}^{+}$ |
| $F_{1\text{-}3}$ | +/=/− | | - | 1/1/0 | 2/0/0 | 1/1/0 | 2/0/0 | 2/0/0 | 0/0/2 | 1/0/1 |
| $F_{4}$ | Simple Multimodal Functions | Median | $3.53 \times 10^{2}$ | $2.95 \times 10^{2}$ | $6.71 \times 10^{2}$ | $2.38 \times 10^{2}$ | $2.72 \times 10^{3}$ | $4.38 \times 10^{2}$ | $1.77 \times 10^{2}$ | $1.89 \times 10^{2}$ |
| | | Mean | $3.53 \times 10^{2}$ | $3.13 \times 10^{2}$ | $5.15 \times 10^{3}$ | $2.40 \times 10^{2}$ | $3.40 \times 10^{3}$ | $4.49 \times 10^{2}$ | $1.78 \times 10^{2}$ | $1.87 \times 10^{2}$ |
| | | Std | $4.89 \times 10^{1}$ | $1.05 \times 10^{2}$ | $2.19 \times 10^{3}$ | $3.19 \times 10^{1}$ | $2.48 \times 10^{3}$ | $1.63 \times 10^{2}$ | $3.69 \times 10^{1}$ | $2.04 \times 10^{1}$ |
| | | *p*-value | - | $5.87 \times 10^{-2\,=}$ | $\mathbf{2.32 \times 10^{-17}}^{+}$ | $3.46 \times 10^{-15\,-}$ | $\mathbf{8.39 \times 10^{-9}}^{+}$ | $\mathbf{1.64 \times 10^{-3}}^{+}$ | $1.01 \times 10^{-22\,-}$ | $1.82 \times 10^{-24\,-}$ |
| $F_{5}$ | | Median | $3.80 \times 10^{1}$ | $1.68 \times 10^{2}$ | $1.47 \times 10^{2}$ | $4.40 \times 10^{2}$ | $6.95 \times 10^{2}$ | $1.45 \times 10^{2}$ | $2.79 \times 10^{-3}$ | $2.02 \times 10^{2}$ |
| | | Mean | $5.17 \times 10^{1}$ | $1.76 \times 10^{2}$ | $3.32 \times 10^{2}$ | $4.41 \times 10^{2}$ | $6.99 \times 10^{2}$ | $1.52 \times 10^{2}$ | $4.32 \times 10^{-3}$ | $1.98 \times 10^{2}$ |
| | | Std | $6.70 \times 10^{1}$ | $4.28 \times 10^{1}$ | $5.14 \times 10^{1}$ | $2.13 \times 10^{1}$ | $3.26 \times 10^{1}$ | $4.76 \times 10^{1}$ | $6.43 \times 10^{-3}$ | $1.57 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{7.82 \times 10^{-17}}^{+}$ | $\mathbf{1.82 \times 10^{-31}}^{+}$ | $\mathbf{4.28 \times 10^{-48}}^{+}$ | $\mathbf{1.83 \times 10^{-57}}^{+}$ | $\mathbf{4.06 \times 10^{-14}}^{+}$ | $1.48 \times 10^{-18\,-}$ | $\mathbf{2.14 \times 10^{-26}}^{+}$ |
| $F_{6}$ | | Median | $8.72 \times 10^{-1}$ | $1.94 \times 10^{0}$ | $1.39 \times 10^{1}$ | $5.71 \times 10^{0}$ | $9.56 \times 10^{1}$ | $9.40 \times 10^{-1}$ | $2.00 \times 10^{2}$ | $1.23 \times 10^{-8}$ |
| | | Mean | $8.50 \times 10^{-1}$ | $2.22 \times 10^{0}$ | $3.78 \times 10^{1}$ | $6.01 \times 10^{0}$ | $9.50 \times 10^{1}$ | $1.07 \times 10^{0}$ | $2.05 \times 10^{2}$ | $2.45 \times 10^{-3}$ |
| | | Std | $2.23 \times 10^{-1}$ | $1.56 \times 10^{0}$ | $1.05 \times 10^{1}$ | $1.24 \times 10^{0}$ | $4.62 \times 10^{0}$ | $1.41 \times 10^{0}$ | $3.08 \times 10^{1}$ | $1.34 \times 10^{-2}$ |
| | | *p*-value | - | $\mathbf{2.34 \times 10^{-5}}^{+}$ | $\mathbf{7.24 \times 10^{-27}}^{+}$ | $\mathbf{4.41 \times 10^{-29}}^{+}$ | $\mathbf{2.69 \times 10^{-69}}^{+}$ | $9.56 \times 10^{-2\,=}$ | $\mathbf{6.41 \times 10^{-14}}^{+}$ | $5.98 \times 10^{-14\,-}$ |

**Table 4.** *Cont.*

| $F$ | Category | Quality | PCLPSO | TCSPSO | AWPSO | CLPSO-LS | PSO-DLP | GL-PSO | HCLPSO | CLPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $F_7$ | Simple Multimodal Functions | Median | $2.27 \times 10^{2}$ | $2.84 \times 10^{2}$ | $1.66 \times 10^{2}$ | $5.15 \times 10^{2}$ | $1.25 \times 10^{3}$ | $2.49 \times 10^{2}$ | $1.65 \times 10^{2}$ | $2.10 \times 10^{2}$ |
| | | Mean | $2.11 \times 10^{2}$ | $2.86 \times 10^{2}$ | $6.93 \times 10^{2}$ | $5.15 \times 10^{2}$ | $1.26 \times 10^{3}$ | $2.54 \times 10^{2}$ | $1.70 \times 10^{2}$ | $2.10 \times 10^{2}$ |
| | | Std | $2.07 \times 10^{1}$ | $5.16 \times 10^{1}$ | $2.42 \times 10^{2}$ | $3.55 \times 10^{1}$ | $1.24 \times 10^{2}$ | $4.58 \times 10^{1}$ | $3.26 \times 10^{1}$ | $1.47 \times 10^{1}$ |
| | | $p$-value | - | $\mathbf{2.24 \times 10^{-6\,+}}$ | $\mathbf{3.76 \times 10^{-15\,+}}$ | $\mathbf{5.94 \times 10^{-32\,+}}$ | $\mathbf{4.78 \times 10^{-45\,+}}$ | $\mathbf{1.74 \times 10^{-3\,+}}$ | $6.00 \times 10^{-1\,=}$ | $9.10 \times 10^{-1\,=}$ |
| $F_8$ | | Median | $3.34 \times 10^{1}$ | $1.83 \times 10^{2}$ | $1.37 \times 10^{2}$ | $4.44 \times 10^{2}$ | $6.93 \times 10^{2}$ | $1.54 \times 10^{2}$ | $1.28 \times 10^{3}$ | $1.96 \times 10^{2}$ |
| | | Mean | $5.39 \times 10^{1}$ | $1.79 \times 10^{2}$ | $3.43 \times 10^{2}$ | $4.45 \times 10^{2}$ | $6.87 \times 10^{2}$ | $1.54 \times 10^{2}$ | $1.52 \times 10^{3}$ | $1.97 \times 10^{2}$ |
| | | Std | $6.63 \times 10^{1}$ | $3.90 \times 10^{1}$ | $5.32 \times 10^{1}$ | $1.36 \times 10^{1}$ | $4.15 \times 10^{1}$ | $6.49 \times 10^{1}$ | $7.16 \times 10^{2}$ | $1.64 \times 10^{1}$ |
| | | $p$-value | - | $\mathbf{6.30 \times 10^{-16\,+}}$ | $\mathbf{6.20 \times 10^{-30\,+}}$ | $\mathbf{1.23 \times 10^{-45\,+}}$ | $\mathbf{1.45 \times 10^{-51\,+}}$ | $\mathbf{5.41 \times 10^{-13\,+}}$ | $\mathbf{8.58 \times 10^{-16\,+}}$ | $\mathbf{3.50 \times 10^{-22\,+}}$ |
| $F_9$ | | Median | $4.82 \times 10^{1}$ | $2.15 \times 10^{3}$ | $2.19 \times 10^{3}$ | $1.10 \times 10^{3}$ | $4.02 \times 10^{4}$ | $1.37 \times 10^{3}$ | $5.26 \times 10^{3}$ | $3.95 \times 10^{3}$ |
| | | Mean | $5.26 \times 10^{1}$ | $2.57 \times 10^{3}$ | $1.01 \times 10^{4}$ | $1.13 \times 10^{3}$ | $4.01 \times 10^{4}$ | $1.92 \times 10^{3}$ | $5.34 \times 10^{3}$ | $4.22 \times 10^{3}$ |
| | | Std | $1.80 \times 10^{1}$ | $1.71 \times 10^{3}$ | $4.58 \times 10^{3}$ | $3.05 \times 10^{2}$ | $4.04 \times 10^{3}$ | $8.15 \times 10^{2}$ | $6.76 \times 10^{2}$ | $1.22 \times 10^{3}$ |
| | | $p$-value | - | $\mathbf{4.85 \times 10^{-11\,+}}$ | $\mathbf{2.00 \times 10^{-17\,+}}$ | $\mathbf{7.63 \times 10^{-27\,+}}$ | $\mathbf{2.20 \times 10^{-51\,+}}$ | $\mathbf{9.63 \times 10^{-8\,+}}$ | $\mathbf{3.95 \times 10^{-16\,+}}$ | $\mathbf{3.36 \times 10^{-26\,+}}$ |
| $F_{10}$ | | Median | $1.09 \times 10^{4}$ | $5.22 \times 10^{3}$ | $3.91 \times 10^{3}$ | $1.31 \times 10^{4}$ | $1.35 \times 10^{4}$ | $7.54 \times 10^{3}$ | $2.34 \times 10^{2}$ | $6.65 \times 10^{3}$ |
| | | Mean | $1.09 \times 10^{4}$ | $5.44 \times 10^{3}$ | $7.54 \times 10^{3}$ | $1.31 \times 10^{4}$ | $1.34 \times 10^{4}$ | $8.22 \times 10^{3}$ | $2.44 \times 10^{2}$ | $6.59 \times 10^{3}$ |
| | | Std | $3.87 \times 10^{2}$ | $8.75 \times 10^{2}$ | $9.13 \times 10^{2}$ | $4.42 \times 10^{2}$ | $3.19 \times 10^{2}$ | $2.20 \times 10^{3}$ | $9.16 \times 10^{1}$ | $4.51 \times 10^{2}$ |
| | | $p$-value | - | $3.87 \times 10^{-37\,-}$ | $1.91 \times 10^{-25\,-}$ | $\mathbf{1.80 \times 10^{-25\,+}}$ | $\mathbf{6.54 \times 10^{-31\,+}}$ | $4.41 \times 10^{-8\,-}$ | $5.81 \times 10^{-42\,-}$ | $2.08 \times 10^{-41\,-}$ |
| $F_{4\text{-}10}$ | +/=/− | | - | 5/1/1 | 6/0/1 | 6/0/1 | 7/0/0 | 5/1/1 | 3/1/3 | 3/1/3 |
| $F_{11}$ | Hybrid Functions | Median | $1.99 \times 10^{2}$ | $2.02 \times 10^{2}$ | $4.20 \times 10^{2}$ | $2.75 \times 10^{2}$ | $1.35 \times 10^{4}$ | $1.26 \times 10^{3}$ | $2.94 \times 10^{6}$ | $1.93 \times 10^{2}$ |
| | | Mean | $1.97 \times 10^{2}$ | $2.41 \times 10^{2}$ | $2.19 \times 10^{3}$ | $1.24 \times 10^{3}$ | $1.27 \times 10^{4}$ | $1.73 \times 10^{3}$ | $4.77 \times 10^{6}$ | $1.92 \times 10^{2}$ |
| | | Std | $6.72 \times 10^{1}$ | $1.11 \times 10^{2}$ | $1.84 \times 10^{3}$ | $5.30 \times 10^{3}$ | $2.66 \times 10^{3}$ | $7.59 \times 10^{2}$ | $5.30 \times 10^{6}$ | $4.16 \times 10^{1}$ |
| | | $p$-value | - | $5.36 \times 10^{-2\,=}$ | $\mathbf{1.74 \times 10^{-7\,+}}$ | $2.84 \times 10^{-1\,=}$ | $\mathbf{1.78 \times 10^{-33\,+}}$ | $\mathbf{4.49 \times 10^{-7\,+}}$ | $\mathbf{1.58 \times 10^{-2\,+}}$ | $6.56 \times 10^{-1\,=}$ |
| $F_{12}$ | | Median | $1.89 \times 10^{6}$ | $2.62 \times 10^{6}$ | $2.11 \times 10^{8}$ | $2.56 \times 10^{7}$ | $7.50 \times 10^{8}$ | $5.68 \times 10^{6}$ | $3.13 \times 10^{4}$ | $1.86 \times 10^{7}$ |
| | | Mean | $2.66 \times 10^{6}$ | $1.28 \times 10^{7}$ | $7.55 \times 10^{9}$ | $2.70 \times 10^{7}$ | $1.55 \times 10^{9}$ | $5.23 \times 10^{7}$ | $2.80 \times 10^{4}$ | $1.96 \times 10^{7}$ |
| | | Std | $1.81 \times 10^{6}$ | $3.91 \times 10^{7}$ | $4.61 \times 10^{9}$ | $1.15 \times 10^{7}$ | $3.13 \times 10^{9}$ | $3.63 \times 10^{8}$ | $1.12 \times 10^{4}$ | $9.65 \times 10^{6}$ |
| | | $p$-value | - | $1.63 \times 10^{-1\,=}$ | $\mathbf{1.44 \times 10^{-12\,+}}$ | $\mathbf{2.19 \times 10^{-16\,+}}$ | $\mathbf{8.95 \times 10^{-3\,+}}$ | $\mathbf{3.73 \times 10^{-2\,+}}$ | $5.02 \times 10^{-2\,=}$ | $\mathbf{3.29 \times 10^{-13\,+}}$ |
| $F_{13}$ | | Median | $3.20 \times 10^{3}$ | $6.21 \times 10^{3}$ | $4.57 \times 10^{6}$ | $3.78 \times 10^{4}$ | $4.84 \times 10^{7}$ | $3.45 \times 10^{3}$ | $9.69 \times 10^{4}$ | $1.11 \times 10^{4}$ |
| | | Mean | $3.90 \times 10^{3}$ | $7.87 \times 10^{3}$ | $1.77 \times 10^{9}$ | $1.03 \times 10^{6}$ | $1.29 \times 10^{8}$ | $8.10 \times 10^{5}$ | $9.77 \times 10^{4}$ | $1.13 \times 10^{4}$ |
| | | Std | $1.75 \times 10^{3}$ | $7.86 \times 10^{3}$ | $1.55 \times 10^{9}$ | $5.43 \times 10^{6}$ | $2.07 \times 10^{8}$ | $9.74 \times 10^{6}$ | $7.55 \times 10^{4}$ | $3.09 \times 10^{3}$ |
| | | $p$-value | - | $\mathbf{9.17 \times 10^{-3\,+}}$ | $\mathbf{5.29 \times 10^{-8\,+}}$ | $3.06 \times 10^{-1\,=}$ | $\mathbf{1.14 \times 10^{-3\,+}}$ | $3.14 \times 10^{-1\,=}$ | $\mathbf{7.97 \times 10^{-17\,+}}$ | $\mathbf{3.96 \times 10^{-16\,+}}$ |
| $F_{14}$ | | Median | $5.42 \times 10^{4}$ | $7.69 \times 10^{4}$ | $3.66 \times 10^{4}$ | $3.32 \times 10^{5}$ | $4.62 \times 10^{6}$ | $1.24 \times 10^{5}$ | $1.81 \times 10^{4}$ | $4.64 \times 10^{5}$ |
| | | Mean | $5.50 \times 10^{4}$ | $1.19 \times 10^{5}$ | $6.82 \times 10^{5}$ | $3.92 \times 10^{5}$ | $4.56 \times 10^{6}$ | $1.21 \times 10^{6}$ | $1.62 \times 10^{4}$ | $5.29 \times 10^{5}$ |
| | | Std | $2.34 \times 10^{4}$ | $1.94 \times 10^{5}$ | $7.16 \times 10^{5}$ | $3.49 \times 10^{5}$ | $2.31 \times 10^{6}$ | $1.82 \times 10^{6}$ | $1.08 \times 10^{4}$ | $2.69 \times 10^{5}$ |
| | | $p$-value | - | $7.84 \times 10^{-2\,=}$ | $\mathbf{1.18 \times 10^{-5\,+}}$ | $\mathbf{2.10 \times 10^{-6\,+}}$ | $\mathbf{2.65 \times 10^{-15\,+}}$ | $\mathbf{3.83 \times 10^{-3\,+}}$ | $4.65 \times 10^{-3\,-}$ | $\mathbf{1.23 \times 10^{-13\,+}}$ |
| $F_{15}$ | | Median | $1.09 \times 10^{3}$ | $4.76 \times 10^{3}$ | $6.85 \times 10^{4}$ | $3.16 \times 10^{4}$ | $2.96 \times 10^{7}$ | $2.99 \times 10^{3}$ | $1.70 \times 10^{3}$ | $8.15 \times 10^{2}$ |
| | | Mean | $1.43 \times 10^{3}$ | $6.94 \times 10^{3}$ | $3.75 \times 10^{7}$ | $2.35 \times 10^{7}$ | $6.54 \times 10^{7}$ | $4.18 \times 10^{3}$ | $1.66 \times 10^{3}$ | $9.31 \times 10^{2}$ |
| | | Std | $1.82 \times 10^{3}$ | $6.93 \times 10^{3}$ | $1.06 \times 10^{8}$ | $7.85 \times 10^{7}$ | $8.03 \times 10^{7}$ | $9.37 \times 10^{3}$ | $4.24 \times 10^{2}$ | $4.56 \times 10^{2}$ |
| | | $p$-value | - | $\mathbf{7.08 \times 10^{-5\,+}}$ | $5.79 \times 10^{-2\,=}$ | $1.07 \times 10^{-1\,=}$ | $\mathbf{3.82 \times 10^{-5\,+}}$ | $\mathbf{1.58 \times 10^{-3\,+}}$ | $\mathbf{5.23 \times 10^{-10\,+}}$ | $4.69 \times 10^{-2\,-}$ |

**Table 4.** *Cont.*

| *F* | Category | Quality | PCLPSO | TCSPSO | AWPSO | CLPSO-LS | PSO-DLP | GL-PSO | HCLPSO | CLPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $F_{16}$ | Hybrid Functions | Median | $9.05 \times 10^{2}$ | $1.59 \times 10^{3}$ | $1.04 \times 10^{3}$ | $3.20 \times 10^{3}$ | $4.92 \times 10^{3}$ | $1.70 \times 10^{3}$ | $1.24 \times 10^{3}$ | $1.45 \times 10^{3}$ |
| | | Mean | $8.76 \times 10^{2}$ | $1.55 \times 10^{3}$ | $2.40 \times 10^{3}$ | $3.21 \times 10^{3}$ | $4.89 \times 10^{3}$ | $1.67 \times 10^{3}$ | $1.21 \times 10^{3}$ | $1.41 \times 10^{3}$ |
| | | Std | $3.89 \times 10^{2}$ | $4.01 \times 10^{2}$ | $5.44 \times 10^{2}$ | $2.00 \times 10^{2}$ | $3.07 \times 10^{2}$ | $7.06 \times 10^{2}$ | $2.71 \times 10^{2}$ | $2.03 \times 10^{2}$ |
| | | *p*-value | - | $\mathbf{3.44 \times 10^{-9}}$ $^{+}$ | $\mathbf{7.98 \times 10^{-19}}$ $^{+}$ | $\mathbf{1.23 \times 10^{-38}}$ $^{+}$ | $\mathbf{3.54 \times 10^{-48}}$ $^{+}$ | $\mathbf{7.80 \times 10^{-10}}$ $^{+}$ | $\mathbf{9.36 \times 10^{-11}}$ $^{+}$ | $\mathbf{8.25 \times 10^{-10}}$ $^{+}$ |
| $F_{17}$ | | Median | $8.27 \times 10^{2}$ | $1.17 \times 10^{3}$ | $5.78 \times 10^{2}$ | $2.07 \times 10^{3}$ | $3.09 \times 10^{3}$ | $1.33 \times 10^{3}$ | $2.68 \times 10^{5}$ | $1.05 \times 10^{3}$ |
| | | Mean | $8.16 \times 10^{2}$ | $1.16 \times 10^{3}$ | $2.06 \times 10^{3}$ | $2.20 \times 10^{3}$ | $3.05 \times 10^{3}$ | $1.30 \times 10^{3}$ | $3.69 \times 10^{5}$ | $1.04 \times 10^{3}$ |
| | | Std | $1.80 \times 10^{2}$ | $2.41 \times 10^{2}$ | $4.49 \times 10^{2}$ | $7.21 \times 10^{2}$ | $3.20 \times 10^{2}$ | $3.35 \times 10^{2}$ | $3.10 \times 10^{5}$ | $1.98 \times 10^{2}$ |
| | | *p*-value | - | $\mathbf{4.68 \times 10^{-7}}$ $^{+}$ | $\mathbf{9.63 \times 10^{-20}}$ $^{+}$ | $\mathbf{2.63 \times 10^{-14}}$ $^{+}$ | $\mathbf{3.84 \times 10^{-38}}$ $^{+}$ | $\mathbf{2.90 \times 10^{-9}}$ $^{+}$ | $\mathbf{5.80 \times 10^{-8}}$ $^{+}$ | $\mathbf{1.28 \times 10^{-4}}$ $^{+}$ |
| $F_{18}$ | | Median | $2.04 \times 10^{5}$ | $2.45 \times 10^{6}$ | $6.51 \times 10^{5}$ | $5.68 \times 10^{6}$ | $2.59 \times 10^{7}$ | $5.67 \times 10^{6}$ | $2.02 \times 10^{4}$ | $1.14 \times 10^{6}$ |
| | | Mean | $2.20 \times 10^{5}$ | $4.08 \times 10^{6}$ | $5.74 \times 10^{6}$ | $8.35 \times 10^{6}$ | $2.64 \times 10^{7}$ | $7.66 \times 10^{6}$ | $2.12 \times 10^{4}$ | $1.31 \times 10^{6}$ |
| | | Std | $1.42 \times 10^{5}$ | $4.41 \times 10^{6}$ | $9.83 \times 10^{6}$ | $6.47 \times 10^{6}$ | $1.17 \times 10^{7}$ | $8.33 \times 10^{6}$ | $1.69 \times 10^{4}$ | $7.76 \times 10^{5}$ |
| | | *p*-value | - | $\mathbf{1.22 \times 10^{-5}}$ $^{+}$ | $\mathbf{3.20 \times 10^{-3}}$ $^{+}$ | $\mathbf{4.49 \times 10^{-9}}$ $^{+}$ | $\mathbf{1.19 \times 10^{-17}}$ $^{+}$ | $\mathbf{1.98 \times 10^{-5}}$ $^{+}$ | $1.37 \times 10^{-2}$ $^{-}$ | $\mathbf{2.73 \times 10^{-10}}$ $^{+}$ |
| $F_{19}$ | | Median | $1.59 \times 10^{4}$ | $6.45 \times 10^{3}$ | $3.81 \times 10^{5}$ | $2.52 \times 10^{3}$ | $2.03 \times 10^{7}$ | $7.66 \times 10^{3}$ | $7.72 \times 10^{2}$ | $3.36 \times 10^{2}$ |
| | | Mean | $1.52 \times 10^{4}$ | $1.05 \times 10^{4}$ | $2.02 \times 10^{7}$ | $2.51 \times 10^{3}$ | $2.46 \times 10^{7}$ | $1.52 \times 10^{4}$ | $7.36 \times 10^{2}$ | $5.26 \times 10^{2}$ |
| | | Std | $4.55 \times 10^{3}$ | $1.04 \times 10^{4}$ | $4.65 \times 10^{7}$ | $1.75 \times 10^{1}$ | $1.71 \times 10^{7}$ | $1.14 \times 10^{4}$ | $2.19 \times 10^{2}$ | $5.12 \times 10^{2}$ |
| | | *p*-value | - | $2.36 \times 10^{-2}$ $^{-}$ | $\mathbf{2.08 \times 10^{-2}}$ $^{+}$ | $7.69 \times 10^{-26}$ $^{-}$ | $\mathbf{9.26 \times 10^{-11}}$ $^{+}$ | $9.88 \times 10^{-1}$ $^{=}$ | $6.36 \times 10^{-2}$ $^{=}$ | $7.82 \times 10^{-29}$ $^{-}$ |
| $F_{20}$ | | Median | $6.56 \times 10^{2}$ | $8.66 \times 10^{2}$ | $3.76 \times 10^{2}$ | $1.73 \times 10^{3}$ | $1.92 \times 10^{3}$ | $8.65 \times 10^{2}$ | $3.91 \times 10^{2}$ | $5.92 \times 10^{2}$ |
| | | Mean | $6.50 \times 10^{2}$ | $8.43 \times 10^{2}$ | $1.10 \times 10^{3}$ | $1.70 \times 10^{3}$ | $1.93 \times 10^{3}$ | $9.08 \times 10^{2}$ | $3.89 \times 10^{2}$ | $6.14 \times 10^{2}$ |
| | | Std | $1.34 \times 10^{2}$ | $2.78 \times 10^{2}$ | $3.20 \times 10^{2}$ | $1.37 \times 10^{2}$ | $1.39 \times 10^{2}$ | $4.28 \times 10^{2}$ | $3.19 \times 10^{1}$ | $1.32 \times 10^{2}$ |
| | | *p*-value | - | $\mathbf{2.97 \times 10^{-3}}$ $^{+}$ | $\mathbf{1.93 \times 10^{-8}}$ $^{+}$ | $\mathbf{7.31 \times 10^{-32}}$ $^{+}$ | $\mathbf{2.87 \times 10^{-36}}$ $^{+}$ | $\mathbf{3.22 \times 10^{-3}}$ $^{+}$ | $1.16 \times 10^{-1}$ $^{=}$ | $4.01 \times 10^{-1}$ $^{=}$ |
| $F_{11\text{-}20}$ | +/=/− | | - | 6/3/1 | 9/1/0 | 6/3/1 | 10/0/0 | 8/2/0 | 5/3/2 | 6/1/3 |
| $F_{21}$ | Composition Functions | Median | $2.40 \times 10^{2}$ | $3.71 \times 10^{2}$ | $3.26 \times 10^{2}$ | $6.34 \times 10^{2}$ | $9.26 \times 10^{2}$ | $3.67 \times 10^{2}$ | $6.17 \times 10^{3}$ | $4.21 \times 10^{2}$ |
| | | Mean | $2.53 \times 10^{2}$ | $3.67 \times 10^{2}$ | $5.58 \times 10^{2}$ | $6.36 \times 10^{2}$ | $9.27 \times 10^{2}$ | $3.68 \times 10^{2}$ | $5.80 \times 10^{3}$ | $4.21 \times 10^{2}$ |
| | | Std | $6.37 \times 10^{1}$ | $3.71 \times 10^{1}$ | $5.12 \times 10^{1}$ | $1.67 \times 10^{1}$ | $4.03 \times 10^{1}$ | $5.91 \times 10^{1}$ | $1.74 \times 10^{3}$ | $1.53 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{5.12 \times 10^{-17}}$ $^{+}$ | $\mathbf{8.32 \times 10^{-34}}$ $^{+}$ | $\mathbf{1.08 \times 10^{-49}}$ $^{+}$ | $\mathbf{1.57 \times 10^{-56}}$ $^{+}$ | $\mathbf{3.82 \times 10^{-16}}$ $^{+}$ | $\mathbf{1.06 \times 10^{-21}}$ $^{+}$ | $\mathbf{2.50 \times 10^{-30}}$ $^{+}$ |
| $F_{22}$ | | Median | $1.04 \times 10^{2}$ | $5.83 \times 10^{3}$ | $4.07 \times 10^{3}$ | $1.33 \times 10^{4}$ | $1.39 \times 10^{4}$ | $8.24 \times 10^{3}$ | $6.53 \times 10^{2}$ | $7.13 \times 10^{3}$ |
| | | Mean | $2.59 \times 10^{3}$ | $5.78 \times 10^{3}$ | $8.32 \times 10^{3}$ | $1.32 \times 10^{4}$ | $1.38 \times 10^{4}$ | $7.49 \times 10^{3}$ | $6.56 \times 10^{2}$ | $7.14 \times 10^{3}$ |
| | | Std | $4.19 \times 10^{3}$ | $1.47 \times 10^{3}$ | $1.02 \times 10^{3}$ | $3.72 \times 10^{2}$ | $4.19 \times 10^{2}$ | $2.89 \times 10^{3}$ | $3.49 \times 10^{1}$ | $2.79 \times 10^{2}$ |
| | | *p*-value | - | $\mathbf{5.67 \times 10^{-4}}$ $^{+}$ | $\mathbf{9.05 \times 10^{-9}}$ $^{+}$ | $\mathbf{1.91 \times 10^{-18}}$ $^{+}$ | $\mathbf{2.13 \times 10^{-19}}$ $^{+}$ | $\mathbf{3.77 \times 10^{-5}}$ $^{+}$ | $6.51 \times 10^{-4}$ $^{-}$ | $\mathbf{1.07 \times 10^{-6}}$ $^{+}$ |
| $F_{23}$ | | Median | $4.98 \times 10^{2}$ | $6.28 \times 10^{2}$ | $5.46 \times 10^{2}$ | $8.58 \times 10^{2}$ | $1.69 \times 10^{3}$ | $8.10 \times 10^{2}$ | $7.30 \times 10^{2}$ | $6.66 \times 10^{2}$ |
| | | Mean | $4.98 \times 10^{2}$ | $6.45 \times 10^{2}$ | $9.60 \times 10^{2}$ | $8.56 \times 10^{2}$ | $1.69 \times 10^{3}$ | $8.06 \times 10^{2}$ | $7.24 \times 10^{2}$ | $6.66 \times 10^{2}$ |
| | | Std | $1.81 \times 10^{1}$ | $6.32 \times 10^{1}$ | $9.13 \times 10^{1}$ | $1.40 \times 10^{1}$ | $6.44 \times 10^{1}$ | $9.20 \times 10^{1}$ | $3.78 \times 10^{1}$ | $1.93 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{1.17 \times 10^{-17}}$ $^{+}$ | $\mathbf{1.17 \times 10^{-34}}$ $^{+}$ | $\mathbf{4.21 \times 10^{-62}}$ $^{+}$ | $\mathbf{5.77 \times 10^{-66}}$ $^{+}$ | $\mathbf{5.96 \times 10^{-31}}$ $^{+}$ | $\mathbf{1.24 \times 10^{-29}}$ $^{+}$ | $\mathbf{4.80 \times 10^{-40}}$ $^{+}$ |
| $F_{24}$ | | Median | $5.81 \times 10^{2}$ | $7.29 \times 10^{2}$ | $6.29 \times 10^{2}$ | $9.06 \times 10^{2}$ | $1.79 \times 10^{3}$ | $9.00 \times 10^{2}$ | $4.92 \times 10^{2}$ | $8.04 \times 10^{2}$ |
| | | Mean | $5.83 \times 10^{2}$ | $7.42 \times 10^{2}$ | $1.03 \times 10^{3}$ | $9.06 \times 10^{2}$ | $1.78 \times 10^{3}$ | $8.67 \times 10^{2}$ | $5.13 \times 10^{2}$ | $8.05 \times 10^{2}$ |
| | | Std | $1.59 \times 10^{1}$ | $9.22 \times 10^{1}$ | $7.75 \times 10^{1}$ | $1.24 \times 10^{1}$ | $1.53 \times 10^{2}$ | $3.68 \times 10^{1}$ | $3.99 \times 10^{1}$ | $2.69 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{4.50 \times 10^{-13}}$ $^{+}$ | $\mathbf{1.69 \times 10^{-37}}$ $^{+}$ | $\mathbf{8.12 \times 10^{-63}}$ $^{+}$ | $\mathbf{2.04 \times 10^{-45}}$ $^{+}$ | $\mathbf{7.03 \times 10^{-25}}$ $^{+}$ | $2.85 \times 10^{-26}$ $^{-}$ | $\mathbf{5.04 \times 10^{-43}}$ $^{+}$ |

**Table 4.** *Cont.*

| *F* | Category | Quality | PCLPSO | TCSPSO | AWPSO | CLPSO-LS | PSO-DLP | GL-PSO | HCLPSO | CLPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $F_{25}$ | Composition Functions | Median | $6.47 \times 10^{2}$ | $6.66 \times 10^{2}$ | $6.22 \times 10^{2}$ | $5.58 \times 10^{2}$ | $1.52 \times 10^{3}$ | $7.68 \times 10^{2}$ | $3.58 \times 10^{3}$ | $5.31 \times 10^{2}$ |
| | | Mean | $6.49 \times 10^{2}$ | $6.76 \times 10^{2}$ | $3.42 \times 10^{3}$ | $5.59 \times 10^{2}$ | $1.77 \times 10^{3}$ | $7.87 \times 10^{2}$ | $3.53 \times 10^{3}$ | $5.30 \times 10^{2}$ |
| | | Std | $2.77 \times 10^{1}$ | $6.94 \times 10^{1}$ | $1.61 \times 10^{3}$ | $9.54 \times 10^{0}$ | $8.20 \times 10^{2}$ | $8.76 \times 10^{1}$ | $3.33 \times 10^{2}$ | $6.36 \times 10^{0}$ |
| | | *p*-value | - | $\mathbf{4.31 \times 10^{-2}}^{+}$ | $\mathbf{2.70 \times 10^{-13}}^{+}$ | $1.54 \times 10^{-30}{}^{-}$ | $\mathbf{3.98 \times 10^{-10}}^{+}$ | $8.64 \times 10^{-14}{}^{-}$ | $\mathbf{6.06 \times 10^{-24}}^{+}$ | $2.14 \times 10^{-38}{}^{-}$ |
| $F_{26}$ | | Median | $1.75 \times 10^{3}$ | $3.28 \times 10^{3}$ | $2.98 \times 10^{3}$ | $5.62 \times 10^{3}$ | $1.22 \times 10^{4}$ | $3.72 \times 10^{3}$ | $6.49 \times 10^{2}$ | $3.59 \times 10^{3}$ |
| | | Mean | $1.77 \times 10^{3}$ | $3.22 \times 10^{3}$ | $7.17 \times 10^{3}$ | $5.60 \times 10^{3}$ | $1.19 \times 10^{4}$ | $3.95 \times 10^{3}$ | $6.55 \times 10^{2}$ | $3.52 \times 10^{3}$ |
| | | Std | $1.57 \times 10^{2}$ | $8.61 \times 10^{2}$ | $1.22 \times 10^{3}$ | $1.89 \times 10^{2}$ | $1.30 \times 10^{3}$ | $1.17 \times 10^{3}$ | $6.21 \times 10^{1}$ | $3.43 \times 10^{2}$ |
| | | *p*-value | - | $\mathbf{8.61 \times 10^{-13}}^{+}$ | $\mathbf{5.87 \times 10^{-32}}^{+}$ | $\mathbf{1.09 \times 10^{-65}}^{+}$ | $\mathbf{1.44 \times 10^{-45}}^{+}$ | $\mathbf{1.70 \times 10^{-11}}^{+}$ | $6.03 \times 10^{-35}{}^{-}$ | $\mathbf{2.93 \times 10^{-34}}^{+}$ |
| $F_{27}$ | | Median | $8.64 \times 10^{2}$ | $9.28 \times 10^{2}$ | $5.68 \times 10^{2}$ | $7.18 \times 10^{2}$ | $2.71 \times 10^{3}$ | $9.41 \times 10^{2}$ | $4.86 \times 10^{2}$ | $6.35 \times 10^{2}$ |
| | | Mean | $8.74 \times 10^{2}$ | $9.24 \times 10^{2}$ | $1.10 \times 10^{3}$ | $7.32 \times 10^{2}$ | $2.72 \times 10^{3}$ | $9.65 \times 10^{2}$ | $4.86 \times 10^{2}$ | $6.33 \times 10^{2}$ |
| | | Std | $6.67 \times 10^{1}$ | $8.92 \times 10^{1}$ | $1.57 \times 10^{2}$ | $7.97 \times 10^{1}$ | $2.69 \times 10^{2}$ | $9.99 \times 10^{1}$ | $2.67 \times 10^{1}$ | $2.81 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{2.37 \times 10^{-2}}^{+}$ | $\mathbf{1.88 \times 10^{-9}}^{+}$ | $4.61 \times 10^{-9}{}^{-}$ | $\mathbf{2.05 \times 10^{-41}}^{+}$ | $\mathbf{7.54 \times 10^{-4}}^{+}$ | $4.44 \times 10^{-17}{}^{-}$ | $2.64 \times 10^{-22}{}^{-}$ |
| $F_{28}$ | | Median | $6.94 \times 10^{2}$ | $6.53 \times 10^{2}$ | $7.85 \times 10^{2}$ | $5.45 \times 10^{3}$ | $2.46 \times 10^{3}$ | $8.42 \times 10^{2}$ | $1.18 \times 10^{3}$ | $1.70 \times 10^{3}$ |
| | | Mean | $6.96 \times 10^{2}$ | $6.61 \times 10^{2}$ | $4.86 \times 10^{3}$ | $5.34 \times 10^{3}$ | $2.50 \times 10^{3}$ | $8.65 \times 10^{2}$ | $1.18 \times 10^{3}$ | $1.79 \times 10^{3}$ |
| | | Std | $5.33 \times 10^{1}$ | $6.70 \times 10^{1}$ | $1.72 \times 10^{3}$ | $4.56 \times 10^{2}$ | $8.15 \times 10^{2}$ | $1.59 \times 10^{2}$ | $2.98 \times 10^{2}$ | $4.52 \times 10^{2}$ |
| | | *p*-value | - | $1.54 \times 10^{-2}{}^{-}$ | $\mathbf{3.55 \times 10^{-19}}^{+}$ | $\mathbf{4.77 \times 10^{-52}}^{+}$ | $\mathbf{1.65 \times 10^{-17}}^{+}$ | $\mathbf{2.70 \times 10^{-9}}^{+}$ | $\mathbf{3.66 \times 10^{-33}}^{+}$ | $\mathbf{3.82 \times 10^{-19}}^{+}$ |
| $F_{29}$ | | Median | $6.25 \times 10^{2}$ | $1.23 \times 10^{3}$ | $9.78 \times 10^{2}$ | $2.10 \times 10^{3}$ | $5.19 \times 10^{3}$ | $1.27 \times 10^{3}$ | $1.13 \times 10^{6}$ | $1.01 \times 10^{3}$ |
| | | Mean | $6.39 \times 10^{2}$ | $1.25 \times 10^{3}$ | $2.85 \times 10^{3}$ | $2.24 \times 10^{3}$ | $5.32 \times 10^{3}$ | $1.36 \times 10^{3}$ | $1.26 \times 10^{6}$ | $1.02 \times 10^{3}$ |
| | | Std | $9.20 \times 10^{1}$ | $2.82 \times 10^{2}$ | $6.90 \times 10^{2}$ | $5.96 \times 10^{2}$ | $7.66 \times 10^{2}$ | $5.30 \times 10^{2}$ | $5.20 \times 10^{5}$ | $1.51 \times 10^{2}$ |
| | | *p*-value | - | $\mathbf{3.25 \times 10^{-16}}^{+}$ | $\mathbf{1.13 \times 10^{-24}}^{+}$ | $\mathbf{5.67 \times 10^{-21}}^{+}$ | $\mathbf{2.02 \times 10^{-39}}^{+}$ | $\mathbf{3.99 \times 10^{-13}}^{+}$ | $\mathbf{1.81 \times 10^{-13}}^{+}$ | $\mathbf{1.26 \times 10^{-16}}^{+}$ |
| $F_{30}$ | | Median | $2.15 \times 10^{6}$ | $2.32 \times 10^{6}$ | $2.19 \times 10^{6}$ | $1.46 \times 10^{6}$ | $2.70 \times 10^{8}$ | $3.01 \times 10^{6}$ | $3.14 \times 10^{6}$ | $7.30 \times 10^{5}$ |
| | | Mean | $2.19 \times 10^{6}$ | $2.36 \times 10^{6}$ | $8.22 \times 10^{7}$ | $1.16 \times 10^{7}$ | $2.98 \times 10^{8}$ | $3.08 \times 10^{6}$ | $9.14 \times 10^{3}$ | $7.40 \times 10^{5}$ |
| | | Std | $4.13 \times 10^{5}$ | $6.23 \times 10^{5}$ | $1.07 \times 10^{8}$ | $5.47 \times 10^{7}$ | $1.24 \times 10^{8}$ | $2.84 \times 10^{6}$ | $4.83 \times 10^{6}$ | $7.63 \times 10^{4}$ |
| | | *p*-value | - | $1.97 \times 10^{-1}{}^{=}$ | $\mathbf{1.28 \times 10^{-4}}^{+}$ | $3.52 \times 10^{-1}{}^{=}$ | $\mathbf{5.91 \times 10^{-19}}^{+}$ | $\mathbf{4.37 \times 10^{-4}}^{+}$ | $1.34 \times 10^{-10}{}^{-}$ | $7.81 \times 10^{-28}{}^{-}$ |
| $F_{21–30}$ | +/=/− | | - | 8/1/1 | 10/0/0 | 7/1/2 | 10/0/0 | 9/0/1 | 5/0/5 | 7/0/3 |
| | +/=/− | | - | 20/6/3 | 27/1/1 | 20/5/4 | 29/0/0 | 24/3/2 | 13/4/12 | 17/2/10 |
| | *Rank* | | **2.34** | 3.45 | 6.45 | 5.38 | 4.55 | 3.34 | 2.97 | 3.34 |

**Table 5.** Global fitness comparisons between PCLPSO and the seven selected PSO methods on the CEC 2017 benchmark set with the dimensionality set as 100.

| *F* | Category | Quality | PCLPSO | TCSPSO | AWPSO | CLPSO-LS | PSO-DLP | GL-PSO | HCLPSO | CLPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $F_{1}$ | Unimodal Functions | Median | $3.23 \times 10^{3}$ | $8.51 \times 10^{3}$ | $1.64 \times 10^{11}$ | $1.21 \times 10^{10}$ | $5.11 \times 10^{10}$ | $1.47 \times 10^{4}$ | $1.37 \times 10^{4}$ | $1.63 \times 10^{9}$ |
| | | Mean | $4.23 \times 10^{3}$ | $8.15 \times 10^{3}$ | $1.63 \times 10^{11}$ | $1.22 \times 10^{10}$ | $5.39 \times 10^{10}$ | $3.49 \times 10^{4}$ | $2.79 \times 10^{4}$ | $1.78 \times 10^{9}$ |
| | | Std | $3.99 \times 10^{3}$ | $1.15 \times 10^{4}$ | $2.13 \times 10^{10}$ | $1.22 \times 10^{9}$ | $1.15 \times 10^{10}$ | $5.48 \times 10^{4}$ | $2.56 \times 10^{4}$ | $1.51 \times 10^{9}$ |
| | | *p*-value | - | $\mathbf{8.84 \times 10^{-4}}^{+}$ | $\mathbf{4.98 \times 10^{-45}}^{+}$ | $\mathbf{1.18 \times 10^{-51}}^{+}$ | $\mathbf{2.16 \times 10^{-33}}^{+}$ | $\mathbf{3.37 \times 10^{-3}}^{+}$ | $\mathbf{5.59 \times 10^{-6}}^{+}$ | $\mathbf{2.26 \times 10^{-8}}^{+}$ |

**Table 5.** *Cont.*

| *F* | Category | Quality | PCLPSO | TCSPSO | AWPSO | CLPSO-LS | PSO-DLP | GL-PSO | HCLPSO | CLPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $F_3$ | Unimodal Functions | Median | $1.68 \times 10^{5}$ | $2.66 \times 10^{5}$ | $4.62 \times 10^{5}$ | $2.68 \times 10^{-8}$ | $4.83 \times 10^{5}$ | $1.93 \times 10^{5}$ | $7.13 \times 10^{4}$ | $5.10 \times 10^{5}$ |
| | | Mean | $1.71 \times 10^{5}$ | $2.57 \times 10^{5}$ | $4.65 \times 10^{5}$ | $3.61 \times 10^{4}$ | $4.66 \times 10^{5}$ | $1.98 \times 10^{5}$ | $7.37 \times 10^{4}$ | $5.11 \times 10^{5}$ |
| | | Std | $1.22 \times 10^{4}$ | $2.73 \times 10^{4}$ | $7.87 \times 10^{4}$ | $1.38 \times 10^{5}$ | $5.37 \times 10^{4}$ | $2.75 \times 10^{4}$ | $2.21 \times 10^{4}$ | $3.89 \times 10^{4}$ |
| | | *p*-value | - | $\mathbf{3.12 \times 10^{-26\,+}}$ | $\mathbf{5.99 \times 10^{-28\,+}}$ | $1.71 \times 10^{-6\,-}$ | $\mathbf{1.67 \times 10^{-36\,+}}$ | $\mathbf{5.44 \times 10^{-6\,+}}$ | $8.33 \times 10^{-29\,-}$ | $\mathbf{3.28 \times 10^{-47\,+}}$ |
| $F_{1\text{-}3}$ | +/=/− | | - | 2/0/0 | 2/0/0 | 1/0/1 | 2/0/0 | 2/0/0 | 1/0/1 | 2/0/0 |
| $F_4$ | Simple Multimodal Functions | Median | $6.46 \times 10^{2}$ | $6.33 \times 10^{2}$ | $3.03 \times 10^{4}$ | $1.01 \times 10^{3}$ | $9.72 \times 10^{3}$ | $1.62 \times 10^{3}$ | $2.36 \times 10^{2}$ | $3.17 \times 10^{2}$ |
| | | Mean | $6.43 \times 10^{2}$ | $6.36 \times 10^{2}$ | $3.09 \times 10^{4}$ | $1.01 \times 10^{3}$ | $1.12 \times 10^{4}$ | $1.58 \times 10^{3}$ | $2.47 \times 10^{2}$ | $3.26 \times 10^{2}$ |
| | | Std | $8.68 \times 10^{1}$ | $2.68 \times 10^{2}$ | $9.08 \times 10^{3}$ | $1.31 \times 10^{2}$ | $4.20 \times 10^{3}$ | $4.10 \times 10^{2}$ | $2.61 \times 10^{1}$ | $4.38 \times 10^{1}$ |
| | | *p*-value | - | $1.49 \times 10^{-1\,=}$ | $\mathbf{1.08 \times 10^{-25\,+}}$ | $\mathbf{1.08 \times 10^{-18\,+}}$ | $\mathbf{7.60 \times 10^{-20\,+}}$ | $\mathbf{1.19 \times 10^{-17\,+}}$ | $1.02 \times 10^{-31\,-}$ | $3.03 \times 10^{-25\,-}$ |
| $F_5$ | | Median | $8.81 \times 10^{1}$ | $4.37 \times 10^{2}$ | $9.87 \times 10^{2}$ | $1.06 \times 10^{3}$ | $1.63 \times 10^{3}$ | $4.45 \times 10^{2}$ | $4.81 \times 10^{2}$ | $7.46 \times 10^{2}$ |
| | | Mean | $8.83 \times 10^{1}$ | $5.27 \times 10^{2}$ | $1.01 \times 10^{3}$ | $1.06 \times 10^{3}$ | $1.63 \times 10^{3}$ | $4.51 \times 10^{2}$ | $4.89 \times 10^{2}$ | $7.46 \times 10^{2}$ |
| | | Std | $1.38 \times 10^{1}$ | $6.06 \times 10^{1}$ | $1.21 \times 10^{2}$ | $2.62 \times 10^{1}$ | $6.02 \times 10^{1}$ | $8.76 \times 10^{1}$ | $7.94 \times 10^{1}$ | $4.21 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{2.04 \times 10^{-38\,+}}$ | $\mathbf{8.26 \times 10^{-45\,+}}$ | $\mathbf{2.50 \times 10^{-81\,+}}$ | $\mathbf{1.61 \times 10^{-74\,+}}$ | $\mathbf{3.18 \times 10^{-30\,+}}$ | $\mathbf{9.75 \times 10^{-35\,+}}$ | $\mathbf{1.79 \times 10^{-61\,+}}$ |
| $F_6$ | | Median | $2.70 \times 10^{0}$ | $1.49 \times 10^{1}$ | $5.97 \times 10^{1}$ | $3.28 \times 10^{1}$ | $1.17 \times 10^{2}$ | $2.42 \times 10^{0}$ | $9.16 \times 10^{-3}$ | $1.07 \times 10^{-2}$ |
| | | Mean | $2.57 \times 10^{0}$ | $1.64 \times 10^{1}$ | $5.98 \times 10^{1}$ | $3.28 \times 10^{1}$ | $1.17 \times 10^{2}$ | $3.03 \times 10^{0}$ | $1.25 \times 10^{-2}$ | $2.64 \times 10^{-2}$ |
| | | Std | $6.18 \times 10^{-1}$ | $6.35 \times 10^{0}$ | $7.97 \times 10^{0}$ | $1.94 \times 10^{0}$ | $3.82 \times 10^{0}$ | $2.00 \times 10^{0}$ | $1.04 \times 10^{-2}$ | $3.26 \times 10^{-2}$ |
| | | *p*-value | - | $\mathbf{5.02 \times 10^{-17\,+}}$ | $\mathbf{1.86 \times 10^{-43\,+}}$ | $\mathbf{2.09 \times 10^{-61\,+}}$ | $\mathbf{1.14 \times 10^{-78\,+}}$ | $2.39 \times 10^{-1\,=}$ | $1.58 \times 10^{-30\,-}$ | $2.25 \times 10^{-30\,-}$ |
| $F_7$ | | Median | $2.89 \times 10^{2}$ | $9.67 \times 10^{2}$ | $3.45 \times 10^{3}$ | $1.45 \times 10^{3}$ | $3.43 \times 10^{3}$ | $7.43 \times 10^{2}$ | $6.66 \times 10^{2}$ | $7.07 \times 10^{2}$ |
| | | Mean | $3.91 \times 10^{2}$ | $1.11 \times 10^{3}$ | $3.46 \times 10^{3}$ | $1.53 \times 10^{3}$ | $3.50 \times 10^{3}$ | $7.62 \times 10^{2}$ | $6.57 \times 10^{2}$ | $7.02 \times 10^{2}$ |
| | | Std | $1.81 \times 10^{2}$ | $1.35 \times 10^{2}$ | $6.74 \times 10^{2}$ | $1.75 \times 10^{2}$ | $3.46 \times 10^{2}$ | $1.21 \times 10^{2}$ | $1.06 \times 10^{2}$ | $6.45 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{2.01 \times 10^{-20\,+}}$ | $\mathbf{6.86 \times 10^{-32\,+}}$ | $\mathbf{1.96 \times 10^{-32\,+}}$ | $\mathbf{4.89 \times 10^{-46\,+}}$ | $\mathbf{3.93 \times 10^{-13\,+}}$ | $\mathbf{3.76 \times 10^{-9\,+}}$ | $\mathbf{2.11 \times 10^{-12\,+}}$ |
| $F_8$ | | Median | $9.19 \times 10^{1}$ | $4.90 \times 10^{2}$ | $1.07 \times 10^{3}$ | $1.07 \times 10^{3}$ | $1.74 \times 10^{3}$ | $4.68 \times 10^{2}$ | $5.69 \times 10^{2}$ | $7.41 \times 10^{2}$ |
| | | Mean | $9.49 \times 10^{1}$ | $5.26 \times 10^{2}$ | $1.07 \times 10^{3}$ | $1.07 \times 10^{3}$ | $1.74 \times 10^{3}$ | $5.10 \times 10^{2}$ | $5.71 \times 1^{2}$ | $7.48 \times 10^{2}$ |
| | | Std | $1.76 \times 10^{1}$ | $1.04 \times 10^{2}$ | $9.22 \times 10^{1}$ | $2.22 \times 10^{1}$ | $5.03 \times 10^{1}$ | $1.55 \times 10^{2}$ | $7.23 \times 10^{1}$ | $3.27 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{1.35 \times 10^{-28\,+}}$ | $\mathbf{1.50 \times 10^{-52\,+}}$ | $\mathbf{1.34 \times 10^{-82\,+}}$ | $\mathbf{9.45 \times 10^{-80\,+}}$ | $\mathbf{5.21 \times 10^{-21\,+}}$ | $\mathbf{1.01 \times 10^{-40\,+}}$ | $\mathbf{1.07 \times 10^{-65\,+}}$ |
| $F_9$ | | Median | $2.44 \times 10^{2}$ | $1.11 \times 10^{4}$ | $3.37 \times 10^{4}$ | $1.57 \times 10^{4}$ | $1.17 \times 10^{5}$ | $1.28 \times 10^{4}$ | $8.66 \times 10^{3}$ | $2.24 \times 10^{4}$ |
| | | Mean | $2.62 \times 10^{2}$ | $1.31 \times 10^{4}$ | $3.56 \times 10^{4}$ | $1.56 \times 10^{4}$ | $1.16 \times 10^{5}$ | $1.41 \times 10^{4}$ | $8.77 \times 10^{3}$ | $2.32 \times 10^{4}$ |
| | | Std | $8.93 \times 10^{1}$ | $4.36 \times 10^{3}$ | $7.71 \times 10^{3}$ | $1.83 \times 10^{3}$ | $5.75 \times 10^{3}$ | $7.21 \times 10^{3}$ | $2.87 \times 10^{3}$ | $4.55 \times 10^{3}$ |
| | | *p*-value | - | $\mathbf{2.77 \times 10^{-20\,+}}$ | $\mathbf{8.13 \times 10^{-33\,+}}$ | $\mathbf{3.59 \times 10^{-47\,+}}$ | $\mathbf{3.50 \times 10^{-69\,+}}$ | $\mathbf{4.18 \times 10^{-15\,+}}$ | $\mathbf{3.40 \times 10^{-23\,+}}$ | $\mathbf{4.80 \times 10^{-35\,+}}$ |
| $F_{10}$ | | Median | $2.63 \times 10^{4}$ | $1.33 \times 10^{4}$ | $1.84 \times 10^{4}$ | $3.02 \times 10^{4}$ | $3.01 \times 10^{4}$ | $1.98 \times 10^{4}$ | $1.28 \times 10^{4}$ | $2.17 \times 10^{4}$ |
| | | Mean | $2.64 \times 10^{4}$ | $1.33 \times 10^{4}$ | $1.83 \times 10^{4}$ | $3.01 \times 10^{4}$ | $3.00 \times 10^{4}$ | $2.07 \times 10^{4}$ | $1.31 \times 10^{4}$ | $2.18 \times 10^{4}$ |
| | | Std | $5.97 \times 10^{2}$ | $3.56 \times 10^{3}$ | $1.69 \times 10^{3}$ | $3.82 \times 10^{2}$ | $6.27 \times 10^{2}$ | $3.75 \times 10^{3}$ | $1.20 \times 10^{3}$ | $5.11 \times 10^{2}$ |
| | | *p*-value | - | $2.56 \times 10^{-26\,-}$ | $2.43 \times 10^{-32\,-}$ | $\mathbf{4.10 \times 10^{-36\,+}}$ | $\mathbf{4.68 \times 10^{-31\,+}}$ | $3.18 \times 10^{-11\,-}$ | $1.74 \times 10^{-51\,-}$ | $2.12 \times 10^{-38\,-}$ |
| $F_4{}^{-10}$ | +/=/− | | - | 2005/1/1 | 6/0/1 | 7/0/0 | 7/0/0 | 2005/1/1 | 4/0/3 | 4/0/3 |
| $F_{11}$ | Hybrid Functions | Median | $2.32 \times 10^{3}$ | $1.28 \times 10^{4}$ | $3.22 \times 10^{4}$ | $1.56 \times 10^{3}$ | $1.66 \times 10^{5}$ | $2.51 \times 10^{4}$ | $8.47 \times 10^{2}$ | $1.34 \times 10^{3}$ |
| | | Mean | $2.65 \times 10^{3}$ | $4.95 \times 10^{3}$ | $4.29 \times 10^{4}$ | $6.42 \times 10^{3}$ | $1.67 \times 10^{5}$ | $2.38 \times 10^{4}$ | $8.43 \times 10^{2}$ | $1.34 \times 10^{3}$ |
| | | Std | $1.74 \times 10^{3}$ | $3.75 \times 10^{3}$ | $3.51 \times 10^{4}$ | $9.25 \times 10^{3}$ | $2.35 \times 10^{4}$ | $8.97 \times 10^{3}$ | $2.49 \times 10^{2}$ | $1.62 \times 10^{2}$ |
| | | *p*-value | - | $\mathbf{2.37 \times 10^{-18\,+}}$ | $\mathbf{4.75 \times 10^{-8\,+}}$ | $\mathbf{3.20 \times 10^{-2\,+}}$ | $\mathbf{9.37 \times 10^{-43\,+}}$ | $\mathbf{2.16 \times 10^{-18\,+}}$ | $5.55 \times 10^{-7\,-}$ | $1.31 \times 10^{-4\,-}$ |

**Table 5.** *Cont.*

| *F* | Category | Quality | PCLPSO | TCSPSO | AWPSO | CLPSO-LS | PSO-DLP | GL-PSO | HCLPSO | CLPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $F_{12}$ | Hybrid Functions | Median | $4.16 \times 10^{7}$ | $2.57 \times 10^{7}$ | $4.54 \times 10^{10}$ | $8.74 \times 10^{8}$ | $1.72 \times 10^{10}$ | $2.31 \times 10^{8}$ | $1.32 \times 10^{7}$ | $7.81 \times 10^{7}$ |
| | | Mean | $4.57 \times 10^{7}$ | $8.39 \times 10^{7}$ | $4.69 \times 10^{10}$ | $9.29 \times 10^{8}$ | $1.94 \times 10^{10}$ | $3.13 \times 10^{8}$ | $2.77 \times 10^{7}$ | $8.97 \times 10^{7}$ |
| | | Std | $3.51 \times 10^{7}$ | $5.12 \times 10^{7}$ | $1.70 \times 10^{10}$ | $3.17 \times 10^{8}$ | $1.17 \times 10^{10}$ | $2.69 \times 10^{8}$ | $7.32 \times 10^{7}$ | $4.15 \times 10^{7}$ |
| | | *p*-value | - | $7.98 \times 10^{-1}$ $^{=}$ | **$9.90 \times 10^{-22}$ $^{+}$** | **$8.13 \times 10^{-22}$ $^{+}$** | **$9.09 \times 10^{-13}$ $^{+}$** | **$1.31 \times 10^{-6}$ $^{+}$** | $2.29 \times 10^{-1}$ $^{=}$ | **$4.26 \times 10^{-5}$ $^{+}$** |
| $F_{13}$ | | Median | $3.50 \times 10^{3}$ | $4.12 \times 10^{3}$ | $5.19 \times 10^{9}$ | $1.23 \times 10^{4}$ | $1.94 \times 10^{8}$ | $1.18 \times 10^{4}$ | $2.20 \times 10^{4}$ | $3.82 \times 10^{4}$ |
| | | Mean | $3.74 \times 10^{3}$ | $5.47 \times 10^{3}$ | $5.72 \times 10^{9}$ | $5.54 \times 10^{6}$ | $2.64 \times 10^{8}$ | $2.56 \times 10^{7}$ | $2.24 \times 10^{4}$ | $4.13 \times 10^{4}$ |
| | | Std | $1.80 \times 10^{3}$ | $6.69 \times 10^{3}$ | $2.99 \times 10^{9}$ | $2.10 \times 10^{7}$ | $2.39 \times 10^{8}$ | $1.38 \times 10^{8}$ | $1.29 \times 10^{4}$ | $1.61 \times 10^{4}$ |
| | | *p*-value | - | **$4.73 \times 10^{-2}$ $^{+}$** | **$5.27 \times 10^{-15}$ $^{+}$** | $1.54 \times 10^{-1}$ $^{=}$ | **$1.12 \times 10^{-7}$ $^{+}$** | $3.15 \times 10^{-1}$ $^{=}$ | **$1.18 \times 10^{-10}$ $^{+}$** | **$2.47 \times 10^{-18}$ $^{+}$** |
| $F_{14}$ | | Median | $3.68 \times 10^{5}$ | $1.05 \times 10^{6}$ | $1.52 \times 10^{7}$ | $4.31 \times 10^{6}$ | $1.84 \times 10^{7}$ | $5.15 \times 10^{5}$ | $6.31 \times 10^{5}$ | $4.85 \times 10^{6}$ |
| | | Mean | $4.87 \times 10^{5}$ | $1.61 \times 10^{6}$ | $1.84 \times 10^{7}$ | $5.24 \times 10^{6}$ | $1.94 \times 10^{7}$ | $2.70 \times 10^{6}$ | $9.83 \times 10^{5}$ | $5.01 \times 10^{6}$ |
| | | Std | $3.47 \times 10^{5}$ | $1.44 \times 10^{6}$ | $1.53 \times 10^{7}$ | $3.40 \times 10^{6}$ | $9.98 \times 10^{6}$ | $4.02 \times 10^{6}$ | $9.35 \times 10^{5}$ | $1.22 \times 10^{6}$ |
| | | *p*-value | - | **$6.93 \times 10^{-5}$ $^{+}$** | **$2.67 \times 10^{-8}$ $^{+}$** | **$2.68 \times 10^{-10}$ $^{+}$** | **$8.31 \times 10^{-15}$ $^{+}$** | **$3.92 \times 10^{-3}$ $^{+}$** | **$8.47 \times 10^{-3}$ $^{+}$** | **$3.23 \times 10^{-27}$ $^{+}$** |
| $F_{15}$ | | Median | $9.50 \times 10^{2}$ | $2.53 \times 10^{3}$ | $1.96 \times 10^{9}$ | $2.77 \times 10^{4}$ | $9.67 \times 10^{5}$ | $2.72 \times 10^{3}$ | $7.93 \times 10^{3}$ | $4.82 \times 10^{3}$ |
| | | Mean | $1.08 \times 10^{3}$ | $1.57 \times 10^{5}$ | $1.86 \times 10^{9}$ | $4.53 \times 10^{7}$ | $6.15 \times 10^{6}$ | $1.17 \times 10^{5}$ | $1.16 \times 10^{4}$ | $5.32 \times 10^{3}$ |
| | | Std | $6.51 \times 10^{2}$ | $3.13 \times 10^{6}$ | $1.13 \times 10^{9}$ | $2.45 \times 10^{8}$ | $1.67 \times 10^{7}$ | $6.08 \times 10^{5}$ | $1.04 \times 10^{4}$ | $2.74 \times 10^{3}$ |
| | | *p*-value | - | $2.42 \times 10^{-1}$ $^{=}$ | **$1.22 \times 10^{-12}$ $^{+}$** | $3.15 \times 10^{-1}$ $^{=}$ | **$4.88 \times 10^{-2}$ $^{+}$** | $2.99 \times 10^{-1}$ $^{=}$ | **$8.77 \times 10^{-7}$ $^{+}$** | **$2.42 \times 10^{-11}$ $^{+}$** |
| $F_{16}$ | | Median | $2.03 \times 10^{3}$ | $3.98 \times 10^{3}$ | $7.73 \times 10^{3}$ | $8.44 \times 10^{3}$ | $1.31 \times 10^{4}$ | $4.64 \times 10^{3}$ | $4.15 \times 10^{3}$ | $4.02 \times 10^{3}$ |
| | | Mean | $2.17 \times 10^{3}$ | $3.96 \times 10^{3}$ | $7.73 \times 10^{3}$ | $8.43 \times 10^{3}$ | $1.29 \times 10^{4}$ | $4.96 \times 10^{3}$ | $4.08 \times 10^{3}$ | $4.07 \times 10^{3}$ |
| | | Std | $9.82 \times 10^{2}$ | $8.18 \times 10^{2}$ | $1.32 \times 10^{3}$ | $3.39 \times 10^{2}$ | $8.74 \times 10^{2}$ | $1.69 \times 10^{3}$ | $8.40 \times 10^{2}$ | $3.43 \times 10^{2}$ |
| | | *p*-value | - | **$1.30 \times 10^{-10}$ $^{+}$** | **$5.52 \times 10^{-26}$ $^{+}$** | **$2.65 \times 10^{-39}$ $^{+}$** | **$9.60 \times 10^{-47}$ $^{+}$** | **$1.20 \times 10^{-10}$ $^{+}$** | **$4.43 \times 10^{-11}$ $^{+}$** | **$2.98 \times 10^{-14}$ $^{+}$** |
| $F_{17}$ | | Median | $2.43 \times 10^{3}$ | $2.90 \times 10^{3}$ | $8.10 \times 10^{3}$ | $5.65 \times 10^{3}$ | $6.80 \times 10^{3}$ | $3.42 \times 10^{3}$ | $3.76 \times 10^{3}$ | $3.23 \times 10^{3}$ |
| | | Mean | $2.13 \times 10^{3}$ | $2.91 \times 10^{3}$ | $9.09 \times 10^{3}$ | $5.78 \times 10^{3}$ | $7.15 \times 10^{3}$ | $3.35 \times 10^{3}$ | $3.86 \times 10^{3}$ | $3.21 \times 10^{3}$ |
| | | Std | $7.24 \times 10^{2}$ | $4.93 \times 10^{2}$ | $3.14 \times 10^{3}$ | $8.25 \times 10^{2}$ | $2.42 \times 10^{3}$ | $1.03 \times 10^{3}$ | $7.27 \times 10^{2}$ | $2.86 \times 10^{2}$ |
| | | *p*-value | - | **$4.27 \times 10^{-5}$ $^{+}$** | **$4.30 \times 10^{-17}$ $^{+}$** | **$1.08 \times 10^{-25}$ $^{+}$** | **$1.15 \times 10^{-15}$ $^{+}$** | **$1.71 \times 10^{-6}$ $^{+}$** | **$5.38 \times 10^{-13}$ $^{+}$** | **$2.70 \times 10^{-10}$ $^{+}$** |
| $F_{18}$ | | Median | $4.35 \times 10^{5}$ | $3.46 \times 10^{6}$ | $1.76 \times 10^{7}$ | $2.66 \times 10^{7}$ | $1.59 \times 10^{7}$ | $6.31 \times 10^{5}$ | $1.10 \times 10^{6}$ | $7.77 \times 10^{6}$ |
| | | Mean | $4.30 \times 10^{5}$ | $4.36 \times 10^{6}$ | $2.51 \times 10^{7}$ | $3.23 \times 10^{7}$ | $1.89 \times 10^{7}$ | $1.32 \times 10^{6}$ | $1.97 \times 10^{6}$ | $7.50 \times 10^{6}$ |
| | | Std | $8.99 \times 10^{4}$ | $2.32 \times 10^{6}$ | $2.29 \times 10^{7}$ | $2.52 \times 10^{7}$ | $9.71 \times 10^{6}$ | $1.72 \times 10^{6}$ | $1.65 \times 10^{6}$ | $2.45 \times 10^{6}$ |
| | | *p*-value | - | **$3.35 \times 10^{-11}$ $^{+}$** | **$2.09 \times 10^{-7}$ $^{+}$** | **$4.22 \times 10^{-9}$ $^{+}$** | **$7.29 \times 10^{-15}$ $^{+}$** | **$6.39 \times 10^{-3}$ $^{+}$** | **$3.81 \times 10^{-6}$ $^{+}$** | **$1.25 \times 10^{-22}$ $^{+}$** |
| $F_{19}$ | | Median | $7.87 \times 10^{2}$ | $4.74 \times 10^{3}$ | $9.56 \times 10^{8}$ | $1.26 \times 10^{4}$ | $1.02 \times 1^{7}$ | $1.37 \times 10^{3}$ | $1.28 \times 10^{4}$ | $1.78 \times 10^{3}$ |
| | | Mean | $9.54 \times 10^{2}$ | $3.51 \times 10^{3}$ | $1.28 \times 10^{9}$ | $3.76 \times 10^{5}$ | $2.00 \times 10^{7}$ | $9.64 \times 10^{5}$ | $3.40 \times 10^{5}$ | $1.97 \times 10^{3}$ |
| | | Std | $7.68 \times 10^{2}$ | $7.98 \times 10^{5}$ | $9.44 \times 10^{8}$ | $1.23 \times 10^{6}$ | $3.72 \times 10^{7}$ | $5.17 \times 10^{6}$ | $1.76 \times 10^{6}$ | $7.25 \times 10^{2}$ |
| $F_{11\text{–}20}$ | +/=/− | | - | 6/3/1 | 10/0/0 | 7/3/0 | 10/0/0 | 7/3/0 | 6/2/2 | 8/0/2 |
| $F_{21}$ | Composition Functions | Median | $3.38 \times 10^{2}$ | $7.14 \times 10^{2}$ | $1.39 \times 10^{3}$ | $1.31 \times 10^{3}$ | $2.28 \times 10^{3}$ | $7.46 \times 10^{2}$ | $8.27 \times 10^{2}$ | $9.66 \times 10^{2}$ |
| | | Mean | $3.40 \times 10^{2}$ | $7.54 \times 10^{2}$ | $1.39 \times 10^{3}$ | $1.31 \times 10^{3}$ | $2.27 \times 10^{3}$ | $7.89 \times 10^{2}$ | $8.36 \times 10^{2}$ | $9.62 \times 10^{2}$ |
| | | Std | $1.80 \times 10^{1}$ | $8.75 \times 10^{1}$ | $1.41 \times 10^{2}$ | $2.82 \times 10^{1}$ | $1.01 \times 10^{2}$ | $1.84 \times 10^{2}$ | $7.27 \times 10^{1}$ | $3.09 \times 10^{1}$ |
| | | *p*-value | - | **$6.33 \times 10^{-31}$ $^{+}$** | **$4.09 \times 10^{-44}$ $^{+}$** | **$3.08 \times 10^{-78}$ $^{+}$** | **$2.30 \times 10^{-67}$ $^{+}$** | **$2.61 \times 10^{-19}$ $^{+}$** | **$1.48 \times 10^{-41}$ $^{+}$** | **$1.87 \times 10^{-65}$ $^{+}$** |
| $F_{22}$ | | Median | $2.73 \times 10^{4}$ | $1.50 \times 10^{4}$ | $1.96 \times 10^{4}$ | $3.04 \times 10^{4}$ | $3.19 \times 10^{4}$ | $2.30 \times 10^{4}$ | $1.43 \times 10^{4}$ | $2.25 \times 10^{4}$ |
| | | Mean | $2.62 \times 10^{4}$ | $1.53 \times 10^{4}$ | $1.99 \times 10^{4}$ | $3.03 \times 10^{4}$ | $3.17 \times 10^{4}$ | $2.35 \times 10^{4}$ | $1.41 \times 10^{4}$ | $2.23 \times 10^{4}$ |
| | | Std | $5.01 \times 10^{3}$ | $1.51 \times 10^{3}$ | $2.08 \times 10^{3}$ | $6.31 \times 10^{2}$ | $4.70 \times 10^{2}$ | $4.30 \times 10^{3}$ | $9.64 \times 10^{2}$ | $6.65 \times 10^{2}$ |
| | | *p*-value | - | $5.37 \times 10^{-17}$ $^{-}$ | $3.37 \times 10^{-8}$ $^{-}$ | **$3.01 \times 10^{-5}$ $^{+}$** | **$1.18 \times 10^{-7}$ $^{+}$** | $3.08 \times 10^{-2}$ $^{-}$ | $8.06 \times 10^{-19}$ $^{-}$ | $9.34 \times 10^{-5}$ $^{-}$ |

**Table 5.** *Cont.*

| *F* | Category | Quality | PCLPSO | TCSPSO | AWPSO | CLPSO-LS | PSO-DLP | GL-PSO | HCLPSO | CLPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $F_{23}$ | Composition Functions | Median | $7.76 \times 10^{2}$ | $9.75 \times 10^{2}$ | $2.04 \times 10^{3}$ | $1.52 \times 10^{3}$ | $3.62 \times 10^{3}$ | $1.39 \times 10^{3}$ | $8.91 \times 10^{2}$ | $9.00 \times 10^{2}$ |
| | | Mean | $7.71 \times 10^{2}$ | $1.02 \times 10^{3}$ | $2.03 \times 10^{3}$ | $1.52 \times 10^{3}$ | $3.62 \times 10^{3}$ | $1.39 \times 10^{3}$ | $8.80 \times 10^{2}$ | $9.01 \times 10^{2}$ |
| | | Std | $3.46 \times 10^{1}$ | $1.22 \times 10^{2}$ | $1.64 \times 10^{2}$ | $3.13 \times 10^{1}$ | $1.57 \times 10^{2}$ | $1.96 \times 10^{2}$ | $3.82 \times 10^{1}$ | $2.70 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{1.04 \times 10^{-13}}^{+}$ | $\mathbf{1.31 \times 10^{-44}}^{+}$ | $\mathbf{2.84 \times 10^{-63}}^{+}$ | $\mathbf{6.65 \times 10^{-66}}^{+}$ | $\mathbf{2.33 \times 10^{-24}}^{+}$ | $\mathbf{1.05 \times 10^{-16}}^{+}$ | $\mathbf{3.86 \times 10^{-23}}^{+}$ |
| $F_{24}$ | | Median | $1.23 \times 10^{3}$ | $1.51 \times 10^{3}$ | $2.80 \times 10^{3}$ | $1.87 \times 10^{3}$ | $6.05 \times 10^{3}$ | $2.05 \times 10^{3}$ | $1.52 \times 10^{3}$ | $1.49 \times 10^{3}$ |
| | | Mean | $1.23 \times 10^{3}$ | $1.61 \times 10^{3}$ | $2.87 \times 10^{3}$ | $1.88 \times 10^{3}$ | $5.91 \times 10^{3}$ | $2.03 \times 10^{3}$ | $1.53 \times 10^{3}$ | $1.49 \times 10^{3}$ |
| | | Std | $5.27 \times 10^{1}$ | $1.33 \times 10^{2}$ | $2.06 \times 10^{2}$ | $7.30 \times 10^{1}$ | $6.40 \times 10^{2}$ | $1.34 \times 10^{2}$ | $6.56 \times 10^{1}$ | $2.54 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{3.61 \times 10^{-16}}^{+}$ | $\mathbf{3.29 \times 10^{-45}}^{+}$ | $\mathbf{1.85 \times 10^{-43}}^{+}$ | $\mathbf{7.83 \times 10^{-44}}^{+}$ | $\mathbf{3.84 \times 10^{-37}}^{+}$ | $\mathbf{1.62 \times 10^{-26}}^{+}$ | $\mathbf{1.64 \times 10^{-31}}^{+}$ |
| $F_{25}$ | | Median | $1.52 \times 10^{3}$ | $1.31 \times 10^{3}$ | $1.43 \times 10^{4}$ | $3.37 \times 10^{3}$ | $5.16 \times 10^{3}$ | $1.79 \times 10^{3}$ | $7.87 \times 10^{2}$ | $9.02 \times 10^{2}$ |
| | | Mean | $1.52 \times 10^{3}$ | $1.31 \times 10^{3}$ | $1.48 \times 10^{4}$ | $3.43 \times 10^{3}$ | $5.08 \times 10^{3}$ | $1.77 \times 10^{3}$ | $7.82 \times 10^{2}$ | $9.07 \times 10^{2}$ |
| | | Std | $1.58 \times 10^{2}$ | $2.86 \times 10^{2}$ | $4.22 \times 10^{3}$ | $2.65 \times 10^{2}$ | $1.05 \times 10^{3}$ | $2.80 \times 10^{2}$ | $7.70 \times 10^{1}$ | $4.86 \times 10^{1}$ |
| | | *p*-value | - | $1.15 \times 10^{-2\,-}$ | $\mathbf{1.63 \times 10^{-24}}^{+}$ | $\mathbf{5.68 \times 10^{-40}}^{+}$ | $\mathbf{6.55 \times 10^{-26}}^{+}$ | $\mathbf{6.36 \times 10^{-5}}^{+}$ | $8.41 \times 10^{-31\,-}$ | $5.54 \times 10^{-28\,-}$ |
| $F_{26}$ | | Median | $5.61 \times 10^{3}$ | $1.00 \times 10^{4}$ | $2.57 \times 10^{4}$ | $1.41 \times 10^{4}$ | $3.15 \times 10^{4}$ | $1.04 \times 10^{4}$ | $1.11 \times 10^{4}$ | $1.09 \times 10^{4}$ |
| | | Mean | $5.60 \times 10^{3}$ | $9.97 \times 10^{3}$ | $2.58 \times 10^{4}$ | $1.40 \times 10^{4}$ | $3.15 \times 10^{4}$ | $1.10 \times 10^{4}$ | $1.10 \times 10^{4}$ | $1.10 \times 10^{4}$ |
| | | Std | $4.01 \times 10^{2}$ | $1.23 \times 10^{3}$ | $2.67 \times 10^{3}$ | $3.41 \times 10^{2}$ | $2.73 \times 10^{3}$ | $2.32 \times 10^{3}$ | $8.60 \times 10^{2}$ | $3.14 \times 10^{2}$ |
| | | *p*-value | - | $\mathbf{8.98 \times 10^{-26}}^{+}$ | $\mathbf{1.53 \times 10^{-44}}^{+}$ | $\mathbf{2.43 \times 10^{-63}}^{+}$ | $\mathbf{3.75 \times 10^{-50}}^{+}$ | $\mathbf{2.74 \times 10^{-18}}^{+}$ | $\mathbf{4.68 \times 10^{-38}}^{+}$ | $\mathbf{7.82 \times 10^{-53}}^{+}$ |
| $F_{27}$ | | Median | $1.12 \times 10^{3}$ | $1.17 \times 10^{3}$ | $2.09 \times 10^{3}$ | $7.13 \times 10^{2}$ | $6.21 \times 10^{3}$ | $1.23 \times 10^{3}$ | $7.96 \times 10^{2}$ | $7.58 \times 10^{0}$ |
| | | Mean | $1.11 \times 10^{3}$ | $1.13 \times 10^{3}$ | $2.09 \times 10^{3}$ | $7.22 \times 10^{2}$ | $6.03 \times 10^{3}$ | $1.26 \times 10^{3}$ | $7.95 \times 10^{2}$ | $7.59 \times 10^{2}$ |
| | | Std | $5.83 \times 10^{1}$ | $1.23 \times 10^{2}$ | $3.93 \times 10^{2}$ | $3.75 \times 10^{1}$ | $8.23 \times 10^{2}$ | $9.93 \times 10^{1}$ | $6.38 \times 10^{1}$ | $2.25 \times 10^{1}$ |
| | | *p*-value | - | $\mathbf{3.00 \times 10^{-3}}^{+}$ | $\mathbf{1.72 \times 10^{-19}}^{+}$ | $1.46 \times 10^{-37\,-}$ | $\mathbf{5.07 \times 10^{-39}}^{+}$ | $\mathbf{4.61 \times 10^{-9}}^{+}$ | $1.11 \times 10^{-27\,-}$ | $1.20 \times 10^{-37\,-}$ |
| $F_{28}$ | | Median | $1.60 \times 10^{3}$ | $1.27 \times 10^{3}$ | $2.06 \times 10^{4}$ | $1.30 \times 10^{4}$ | $9.66 \times 10^{3}$ | $2.20 \times 10^{3}$ | $5.84 \times 10^{2}$ | $1.28 \times 10^{4}$ |
| | | Mean | $1.61 \times 10^{3}$ | $1.27 \times 10^{3}$ | $2.10 \times 10^{4}$ | $1.30 \times 10^{4}$ | $1.02 \times 10^{4}$ | $2.29 \times 10^{3}$ | $3.14 \times 10^{3}$ | $1.28 \times 10^{4}$ |
| | | Std | $1.61 \times 10^{2}$ | $4.02 \times 10^{2}$ | $2.02 \times 10^{3}$ | $1.88 \times 10^{2}$ | $2.24 \times 10^{3}$ | $5.73 \times 10^{2}$ | $4.58 \times 10^{3}$ | $4.87 \times 10^{1}$ |
| | | *p*-value | - | $3.48 \times 10^{-3\,-}$ | $\mathbf{1.28 \times 10^{-50}}^{+}$ | $\mathbf{6.31 \times 10^{-90}}^{+}$ | $\mathbf{1.13 \times 10^{-28}}^{+}$ | $\mathbf{3.87 \times 10^{-8}}^{+}$ | $7.28 \times 10^{-2\,=}$ | $\mathbf{4.65 \times 10^{-99}}^{+}$ |
| $F_{29}$ | | Median | $2.21 \times 10^{3}$ | $3.63 \times 10^{3}$ | $8.50 \times 10^{3}$ | $6.16 \times 10^{3}$ | $1.13 \times 10^{4}$ | $4.38 \times 10^{3}$ | $3.95 \times 10^{3}$ | $3.30 \times 10^{3}$ |
| | | Mean | $2.20 \times 10^{3}$ | $3.80 \times 10^{3}$ | $8.91 \times 10^{3}$ | $6.16 \times 10^{3}$ | $1.16 \times 10^{4}$ | $4.43 \times 10^{3}$ | $3.93 \times 10^{3}$ | $3.30 \times 10^{3}$ |
| | | Std | $3.15 \times 10^{2}$ | $5.72 \times 10^{2}$ | $1.90 \times 10^{3}$ | $1.19 \times 10^{3}$ | $2.46 \times 10^{3}$ | $6.94 \times 10^{2}$ | $5.24 \times 10^{2}$ | $2.75 \times 10^{2}$ |
| | | *p*-value | - | $\mathbf{2.80 \times 10^{-18}}^{+}$ | $\mathbf{1.19 \times 10^{-26}}^{+}$ | $\mathbf{6.60 \times 10^{-25}}^{+}$ | $\mathbf{1.32 \times 10^{-28}}^{+}$ | $\mathbf{5.01 \times 10^{-23}}^{+}$ | $\mathbf{2.41 \times 10^{-22}}^{+}$ | $\mathbf{7.36 \times 10^{-21}}^{+}$ |
| $F_{30}$ | Composition Functions | Median | $7.15 \times 10^{5}$ | $8.13 \times 10^{4}$ | $3.86 \times 10^{9}$ | $2.17 \times 10^{4}$ | $7.41 \times 10^{8}$ | $1.83 \times 10^{6}$ | $5.58 \times 10^{3}$ | $5.76 \times 10^{4}$ |
| | | Mean | $9.64 \times 10^{5}$ | $1.64 \times 10^{5}$ | $4.41 \times 10^{9}$ | $3.42 \times 10^{7}$ | $1.15 \times 10^{9}$ | $3.54 \times 10^{6}$ | $1.85 \times 10^{4}$ | $7.32 \times 10^{4}$ |
| | | Std | $6.34 \times 10^{5}$ | $1.22 \times 10^{5}$ | $2.16 \times 10^{9}$ | $1.19 \times 10^{8}$ | $1.12 \times 10^{9}$ | $3.91 \times 10^{6}$ | $2.63 \times 10^{4}$ | $5.28 \times 10^{4}$ |
| | | *p*-value | - | $3.58 \times 10^{-9\,-}$ | $\mathbf{3.81 \times 10^{-16}}^{+}$ | $1.31 \times 10^{-1\,=}$ | $\mathbf{5.74 \times 10^{-7}}^{+}$ | $\mathbf{7.21 \times 10^{-4}}^{+}$ | $3.26 \times 10^{-11\,-}$ | $2.18 \times 10^{-10\,-}$ |
| $F_{21\text{-}30}$ | +/=/− | | - | 6/0/4 | 9/0/1 | 8/1/1 | 10/0/0 | 9/0/1 | 5/1/4 | 6/0/4 |
| | +/=/− | | - | 19/4/6 | 27/0/2 | 23/4/2 | 29/0/0 | 23/4/2 | 16/3/10 | 20/0/9 |
| | *Rank* | | **2.14** | 3.17 | 6.9 | 5.83 | 7.41 | 4.41 | 2.59 | 3.55 |

**Table 6.** Statistical comparison results between PCLPSO and the seven compared PSO methods on the CEC 2017 benchmark problem set with the three dimension sizes in terms of "+/=/−".

| Category | *D* | TCSPSO | AWPSO | CLPSO-LS | PSO-DLP | GL-PSO | HCLPSO | CLPSO |
|---|---|---|---|---|---|---|---|---|
| Unimodal Functions | 30 | 2/0/0 | 2/0/0 | 1/1/0 | 2/0/0 | 1/1/0 | 1/0/1 | 1/0/1 |
| | 50 | 1/1/0 | 2/0/0 | 1/1/0 | 2/0/0 | 2/0/0 | 0/0/2 | 1/0/1 |
| | 100 | 2/0/0 | 2/0/0 | 1/0/1 | 2/0/0 | 2/0/0 | 1/0/1 | 2/0/0 |
| Simple Multimodal Functions | 30 | 4/1/2 | 6/0/1 | 6/0/1 | 7/0/0 | 5/2/0 | 3/0/4 | 3/0/4 |
| | 50 | 5/1/1 | 6/0/1 | 6/0/1 | 7/0/0 | 5/1/1 | 3/1/3 | 3/1/3 |
| | 100 | 5/1/1 | 6/0/1 | 7/0/0 | 7/0/0 | 5/1/1 | 4/0/3 | 4/0/3 |
| Hybrid Functions | 30 | 10/0/0 | 10/0/0 | 10/0/0 | 10/0/0 | 9/1/0 | 8/2/0 | 6/2/2 |
| | 50 | 6/3/1 | 9/1/0 | 6/3/1 | 10/0/0 | 7/2/1 | 5/3/2 | 6/1/3 |
| | 100 | 6/3/1 | 10/0/0 | 7/3/0 | 10/0/0 | 8/2/0 | 6/2/2 | 8/0/2 |
| Composition Functions | 30 | 7/3/0 | 10/0/0 | 7/1/2 | 10/0/0 | 9/1/0 | 5/3/2 | 7/1/2 |
| | 50 | 8/1/1 | 10/0/0 | 7/1/2 | 10/0/0 | 9/0/1 | 5/0/5 | 7/0/3 |
| | 100 | 6/0/4 | 9/0/1 | 8/1/1 | 10/0/0 | 9/0/1 | 5/1/4 | 6/0/4 |
| Whole Set | 30 | 23/4/2 | 28/0/1 | 24/2/3 | 29/0/0 | 24/5/0 | 17/5/7 | 17/3/9 |
| | 50 | 20/6/3 | 27/1/1 | 20/5/4 | 29/0/0 | 24/3/2 | 13/4/12 | 17/2/10 |
| | 100 | 19/4/6 | 27/0/2 | 23/4/2 | 29/0/0 | 23/4/2 | 16/3/10 | 20/0/9 |

Taking a look at Table 4, we obtain the following findings from the comparison results between PCLPSO and the seven compared PSOs on the 50-*D* CEC 2017 functions:

(1) In regard to the Friedman test results, PCLPSO still gains the lowest rank (2.34) among all algorithms. Moreover, such a rank value is still much lower than those of the seven PSO algorithms (at least 2.97). This substantiates that PCLPSO still performs the best on the whole 50-*D* CEC 2017 benchmark set, and its overall performance is significantly better than those of the seven compared PSO variants.

(2) According to the Wilcoxon rank sum test results, PCLPSO presents significant dominance to AWPSO, PSO-DLP, and GL-PSO on 27, 29, and 24 problems, respectively. Compared with TCSPSO and CLPSO-LS, PCLPSO obtains significantly better performance on 20 problems. In competition with HCLPSO, PCLPSO achieves significant superiority on 13 problems and shows inferiority on 12 problems. This demonstrates that PCLPSO obtains highly competitive performance with HCLPSO on the 50-*D* CEC 2017 benchmark set.

(3) Regarding the comparison results on different kinds of benchmark problems, on the two unimodal problems, PCLPSO performs much better than AWPSO, PSO-DLP, and GL-PSO all on these two problems, and performs competitively with TCSPSO, CLPSO-LS, and CLPSO. On the seven simple multimodal problems, PCLPSO performs significantly better than TCSPSO, AWPSO, CLPSO-LS, and PSO-DLP on at least five problems, and obtains highly competitive performance with HCLPSO and CLPSO. In face of the 10 hybrid problems, PCLPSO exhibits much better performance than the compared PSO variants on at least five problems, while it performs worse than them on at most three functions. In particular, in comparison with AWPSO, PSO-DLP, and GL-PSO, PCLPSO displays significant superiority to them on 9, 10, and 8 problems, respectively. When tackling the 10 composition problems, PCLPSO is better than TCSPSO, AWPSO, CLPSO-LS, PSO-DLP, GL-PSO, and CLPSO on at least seven problems, and performs similarly with HCLPSO.

Lastly, observing of Table 5, we achieve the following observations from the comparison results between PCLPSO and the seven selected PSO methods on the 100-*D* CEC 2017 problems:

(1) In terms of the Friedman test results, PCLPSO still obtains the lowest rank value (2.14) among all algorithms. This demonstrates that PCLPSO consistently obtains the best overall performance on the whole 100-*D* CEC 2017 benchmark set.

(2) Regarding the Wilcoxon rank sum test presented in the second to last row, PCLPSO outperforms TCSPSO, AWPSO, CLPSO-LS, PSO-DLP, and GL-PSO on 20, 27, 20,

29, and 24 problems respectively. Competing with HCLPSO and CLPSO, PCLPSO performs significantly better than them on 13 and 17 problems, respectively.

(3) Regarding the optimization performance on different kinds of problems, on the two unimodal problems, PCLPSO beats AWPSO, PSO-DLP, and GL-PSO all on the two problems, and performs competitively with TCSPSO, CLPSO-LS, and CLPSO. On the seven simple multimodal problems, PCLPSO significantly outperforms TCSPSO, AWPSO, CLPSO-LS, PSO-DLP, and GLPSO on at least five problems, and performs worse than them on at most one problem. In comparison with HCLPSO and CLPSO, PCLPSO achieves very competitive performance with them. On the 10 hybrid problems, PCLPSO obtains significantly better performance than the seven PSO variants on at least five problems and obtains worse performance than them on at most three problems. On the 10 composition problems, except for HCLPSO, PCLPSO presents its dominance over the other six compared methods on at least seven problems.

In summary, from Table 6, we can see that PCLPSO consistently performs the best and exhibits significant superiority to the seven compared PSO methods on the CEC 2017 problem set with the three settings of dimensionality. This substantiates that PCLPSO is promising for dealing with optimization problems and has a good scalability in solving various optimization problems. In particular, PCLPSO performs much better than the compared methods on complex problems, such as the hybrid problems and the composition problems. This verifies that PCLPSO has a good optimization ability in dealing with complicated optimization problems.

The superiority of PCLPSO mainly profits from the devised predominant learning strategy, which could construct promising and effective guiding exemplars to update particles. In addition, the proposed dynamic parameter strategies also contribute to the good performance of PCLPSO in improving the swarm diversity. With the cohesive cooperation among the above techniques, PCLPSO could compromise the search diversity and the search effectiveness of particles well to search the solution space to obtain satisfactory performance.

### *4.3. Deep Investigations on PCLPSO*

This section presents experiments for deep observations on PCLPSO to investigate the usefulness of each component, so that it can be determined what contributes to the good performance of PCLPSO.

#### 4.3.1. Effectiveness of the Predominant Cognitive Learning Strategy

First, we carried out experiments to validate the usefulness of the devised predominant cognitive learning strategy. To achieve this goal, we first developed three additional versions of PCLPSO to make comparisons with the proposed PCLPSO. The first is to remove the predominant cognitive learning strategy and directly use the personal best position of the updated particle to guide its update. We name this variant of PCLPSO as "PCLPSO-WPCL". The second variant is to randomly pick a ***pbest*** from those of the other particles to generate an exemplar to update each particle in Equation (11) instead of using a random definitely better one. We name this variant as "PCLPSO-Rand". The third method is to use ***gbest*** to construct the guiding exemplar in Equation (11). We name this PCLPSO variant as "PCLPSO-Gbest".

After the above preparation, we executed experiments on the 50-D CEC 2017 problem set to compare PCLPSO with the three variants. Table 7 presents the comparison results among these four versions of PCLPSO, which are all the mean fitness values of the global best solutions found at the end of the algorithms over 30 independent runs.

From Table 7, it is found that with respect to both the Friedman test results and the number of problems where the associated algorithm performs the best, PCLPSO obtains the best overall performance. In particular, PCLPSO-WPCL achieves the worst performance. This demonstrates that the proposed PCL strategy is effective. Compared with PCLPSO-Gbest, the proposed PCLPSO and PCLPSO-Rand perform much better. This is because in

PCLPSO-Gbest, there is only one predominant position, namely the ***gbest***, used to construct the guiding exemplar. This leads to the diversity of the constructed exemplars being very limited, and thus the learning diversity of particles is not high enough, leading to the swarm diversity being improved limitedly and thus easily falls into local regions. In competition with PCLPSO-Rand, the proposed PCLPSO achieves the best results on more problems (14 problems) and obtains smaller rank (1.52). This demonstrates the superiority of using a predominant cognitive best position over using a random one to construct the learning exemplar to update each particle.

**Table 7.** Comparison results between PCLPSO with different learning strategies on 50-*D* CEC 2017 benchmark problems.

| *F* | **PCLPSO** | **PCLPSO-WPCL** | **PCLPSO-Rand** | **PCLPSO-Gbest** |
|---|---|---|---|---|
| $F_1$ | $\mathbf{4.19 \times 10^{3}}$ | $2.25 \times 10^{1}$ | $4.76 \times 10^{3}$ | $1.51 \times 10^{4}$ |
| $F_3$ | $1.64 \times 10^{4}$ | $5.66 \times 10^{7}$ | $3.85 \times 10^{4}$ | $\mathbf{9.12 \times 10^{3}}$ |
| $F_4$ | $3.53 \times 10^{2}$ | $8.98 \times 10^{4}$ | $\mathbf{2.94 \times 10^{2}}$ | $5.42 \times 10^{2}$ |
| $F_5$ | $\mathbf{5.17 \times 10^{1}}$ | $1.12 \times 10^{3}$ | $1.22 \times 10^{2}$ | $1.49 \times 10^{2}$ |
| $F_6$ | $8.50 \times 10^{-1}$ | $1.42 \times 10^{2}$ | $\mathbf{8.20 \times 10^{-3}}$ | $9.57 \times 10^{0}$ |
| $F_7$ | $\mathbf{2.11 \times 10^{2}}$ | $4.81 \times 10^{3}$ | $2.17 \times 10^{2}$ | $3.01 \times 10^{2}$ |
| $F_8$ | $\mathbf{5.39 \times 10^{1}}$ | $1.12 \times 10^{3}$ | $1.27 \times 10^{2}$ | $1.40 \times 10^{2}$ |
| $F_9$ | $5.26 \times 10^{1}$ | $8.61 \times 10^{4}$ | $\mathbf{6.88 \times 10^{0}}$ | $1.66 \times 10^{3}$ |
| $F_{10}$ | $1.09 \times 10^{4}$ | $1.60 \times 10^{4}$ | $1.13 \times 10^{4}$ | $\mathbf{5.66 \times 10^{3}}$ |
| $F_{11}$ | $1.97 \times 10^{2}$ | $8.71 \times 10^{4}$ | $\mathbf{1.36 \times 10^{2}}$ | $4.71 \times 10^{2}$ |
| $F_{12}$ | $2.66 \times 10^{6}$ | $1.13 \times 10^{1}$ | $\mathbf{2.11 \times 10^{6}}$ | $3.77 \times 10^{7}$ |
| $F_{13}$ | $\mathbf{3.90 \times 10^{3}}$ | $5.84 \times 10^{0}$ | $6.21 \times 10^{3}$ | $9.59 \times 10^{3}$ |
| $F_{14}$ | $\mathbf{5.50 \times 10^{4}}$ | $1.99 \times 10^{8}$ | $1.90 \times 10^{5}$ | $1.35 \times 10^{5}$ |
| $F_{15}$ | $1.43 \times 10^{3}$ | $2.21 \times 10^{0}$ | $\mathbf{6.84 \times 10^{2}}$ | $3.58 \times 10^{3}$ |
| $F_{16}$ | $\mathbf{8.76 \times 10^{2}}$ | $1.05 \times 10^{4}$ | $1.29 \times 10^{3}$ | $1.70 \times 10^{3}$ |
| $F_{17}$ | $\mathbf{8.16 \times 10^{2}}$ | $3.26 \times 10^{5}$ | $8.47 \times 10^{2}$ | $1.10 \times 10^{3}$ |
| $F_{18}$ | $\mathbf{2.20 \times 10^{5}}$ | $7.66 \times 10^{8}$ | $5.17 \times 10^{5}$ | $2.42 \times 10^{6}$ |
| $F_{19}$ | $1.52 \times 10^{4}$ | $1.03 \times 10^{0}$ | $1.60 \times 10^{4}$ | $\mathbf{1.17 \times 10^{4}}$ |
| $F_{20}$ | $\mathbf{6.50 \times 10^{2}}$ | $3.22 \times 10^{3}$ | $7.89 \times 10^{2}$ | $7.69 \times 10^{2}$ |
| $F_{21}$ | $\mathbf{2.53 \times 10^{2}}$ | $1.35 \times 10^{3}$ | $3.25 \times 10^{2}$ | $3.38 \times 10^{2}$ |
| $F_{22}$ | $\mathbf{2.59 \times 10^{3}}$ | $1.67 \times 10^{4}$ | $3.16 \times 10^{3}$ | $3.50 \times 10^{3}$ |
| $F_{23}$ | $4.98 \times 10^{2}$ | $2.52 \times 10^{3}$ | $\mathbf{4.82 \times 10^{2}}$ | $6.76 \times 10^{2}$ |
| $F_{24}$ | $5.83 \times 10^{2}$ | $2.88 \times 10^{3}$ | $\mathbf{5.74 \times 10^{2}}$ | $7.63 \times 10^{2}$ |
| $F_{25}$ | $6.49 \times 10^{2}$ | $4.62 \times 10^{4}$ | $\mathbf{6.18 \times 10^{2}}$ | $8.24 \times 10^{2}$ |
| $F_{26}$ | $1.77 \times 10^{3}$ | $2.53 \times 10^{4}$ | $\mathbf{1.56 \times 10^{3}}$ | $3.17 \times 10^{3}$ |
| $F_{27}$ | $\mathbf{8.74 \times 10^{2}}$ | $5.24 \times 10^{3}$ | $8.85 \times 10^{2}$ | $1.05 \times 10^{3}$ |
| $F_{28}$ | $6.96 \times 10^{2}$ | $1.94 \times 10^{4}$ | $\mathbf{6.14 \times 10^{2}}$ | $8.91 \times 10^{2}$ |
| $F_{29}$ | $\mathbf{6.39 \times 10^{2}}$ | $4.54 \times 10^{5}$ | $6.43 \times 10^{2}$ | $1.44 \times 10^{3}$ |
| $F_{30}$ | $2.19 \times 10^{6}$ | $1.31 \times 10^{10}$ | $\mathbf{2.05 \times 10^{6}}$ | $9.10 \times 10^{6}$ |
| *Rank* | **1.52** | 4.00 | 1.76 | 2.72 |

On the whole, based on the above comparison experiments, the effectiveness of the PCL strategy is demonstrated, which could construct effective exemplars to direct the updating of particles.

### 4.3.2. Effectiveness of the Adaptive Strategy for *F*

Subsequently, we carried out experiments to validate the usefulness of the devised adaptive strategy (Equation (15)) for the learning step ***F***. To this end, we set ***F*** with different fixed values ranging from 0.1 to 0.9. Table 8 shows the comparison results in view of the mean fitness values of the global best solutions found at the end of the associated algorithm over 30 independent runs between PCLPSO with the adaptive *F* and those with different fixed settings of *F* on the 50-*D* CEC 2017 benchmark set. The bolded values in this table mean that the associated algorithms achieve the best performance on the corresponding problems. In addition, the average rank of each configuration of $F$ attained from the Friedman test is also presented in this table.

**Table 8.** Comparison results between PCLPSO with and without the adaptive strategy for $\boldsymbol{F}$ on the 50-$D$ CEC 2017 benchmark set.

| $F$ | **Adaptive-$F$** | $F$ **= 0.1** | $F$ **= 0.2** | $F$ **= 0.3** | $F$ **= 0.4** | $F$ **= 0.5** | $F$ **= 0.6** | $F$ **= 0.7** | $F$ **= 0.8** | $F$ **= 0.9** |
|---|---|---|---|---|---|---|---|---|---|---|
| $F_1$ | $4.19 \times 10^{3}$ | $4.07 \times 10^{6}$ | $1.95 \times 10^{4}$ | $\mathbf{3.43 \times 10^{3}}$ | $4.97 \times 10^{3}$ | $4.24 \times 10^{3}$ | $5.48 \times 10^{3}$ | $6.85 \times 10^{3}$ | $6.08 \times 10^{3}$ | $5.90 \times 10^{3}$ |
| $F_3$ | $1.64 \times 10^{4}$ | $6.45 \times 10^{4}$ | $4.51 \times 10^{4}$ | $3.19 \times 10^{4}$ | $3.44 \times 10^{4}$ | $2.73 \times 10^{4}$ | $\mathbf{9.85 \times 10^{3}}$ | $1.68 \times 10^{4}$ | $1.67 \times 10^{4}$ | $3.07 \times 10^{4}$ |
| $F_4$ | $3.53 \times 10^{2}$ | $\mathbf{2.42 \times 10^{2}}$ | $2.82 \times 10^{2}$ | $3.34 \times 10^{2}$ | $3.34 \times 10^{2}$ | $3.77 \times 10^{2}$ | $3.97 \times 10^{2}$ | $3.23 \times 10^{2}$ | $3.64 \times 10^{2}$ | $3.14 \times 10^{2}$ |
| $F_5$ | $\mathbf{5.17 \times 10^{1}}$ | $2.86 \times 10^{2}$ | $1.39 \times 10^{2}$ | $6.18 \times 10^{1}$ | $6.66 \times 10^{1}$ | $6.54 \times 10^{1}$ | $6.94 \times 10^{1}$ | $1.87 \times 10^{2}$ | $1.70 \times 10^{2}$ | $1.30 \times 10^{2}$ |
| $F_6$ | $8.50 \times 10^{-1}$ | $1.25 \times 10^{1}$ | $2.23 \times 10^{0}$ | $2.64 \times 10^{0}$ | $2.09 \times 10^{0}$ | $2.01 \times 10^{0}$ | $3.25 \times 10^{0}$ | $9.57 \times 10^{-1}$ | $8.87 \times 10^{-1}$ | $\mathbf{1.12 \times 10^{-1}}$ |
| $F_7$ | $\mathbf{2.11 \times 10^{2}}$ | $3.67 \times 10^{2}$ | $2.84 \times 10^{2}$ | $2.35 \times 10^{2}$ | $2.69 \times 10^{2}$ | $3.11 \times 10^{2}$ | $2.31 \times 10^{2}$ | $3.01 \times 10^{2}$ | $2.94 \times 10^{2}$ | $3.01 \times 10^{2}$ |
| $F_8$ | $5.39 \times 10^{1}$ | $2.64 \times 10^{2}$ | $1.33 \times 10^{2}$ | $5.58 \times 10^{1}$ | $\mathbf{3.24 \times 10^{1}}$ | $4.78 \times 10^{1}$ | $3.48 \times 10^{1}$ | $1.48 \times 10^{2}$ | $1.44 \times 10^{2}$ | $1.27 \times 10^{2}$ |
| $F_9$ | $5.26 \times 10^{1}$ | $2.45 \times 10^{3}$ | $2.81 \times 10^{2}$ | $9.37 \times 10^{1}$ | $6.08 \times 10^{1}$ | $5.87 \times 10^{1}$ | $7.73 \times 10^{1}$ | $3.26 \times 10^{1}$ | $4.25 \times 10^{1}$ | $\mathbf{2.40 \times 10^{1}}$ |
| $F_{10}$ | $\mathbf{1.09 \times 10^{4}}$ | $1.17 \times 10^{4}$ | $1.22 \times 10^{4}$ | $1.23 \times 10^{4}$ | $1.23 \times 10^{4}$ | $1.25 \times 10^{4}$ | $1.23 \times 10^{4}$ | $1.27 \times 10^{4}$ | $1.24 \times 10^{4}$ | $1.27 \times 10^{4}$ |
| $F_{11}$ | $\mathbf{1.97 \times 10^{2}}$ | $2.61 \times 10^{2}$ | $2.52 \times 10^{2}$ | $2.55 \times 10^{2}$ | $2.20 \times 10^{2}$ | $2.50 \times 10^{2}$ | $2.28 \times 10^{2}$ | $2.08 \times 10^{2}$ | $2.28 \times 10^{2}$ | $2.08 \times 10^{2}$ |
| $F_{12}$ | $2.66 \times 10^{6}$ | $4.01 \times 10^{6}$ | $2.59 \times 10^{6}$ | $2.83 \times 10^{6}$ | $2.19 \times 10^{6}$ | $\mathbf{2.01 \times 10^{6}}$ | $3.32 \times 10^{6}$ | $4.49 \times 10^{6}$ | $2.88 \times 10^{6}$ | $2.57 \times 10^{6}$ |
| $F_{13}$ | $3.90 \times 10^{3}$ | $4.72 \times 10^{3}$ | $3.60 \times 10^{3}$ | $\mathbf{3.48 \times 10^{3}}$ | $4.28 \times 10^{3}$ | $4.91 \times 10^{3}$ | $4.74 \times 10^{3}$ | $4.52 \times 10^{3}$ | $6.36 \times 10^{3}$ | $7.24 \times 10^{3}$ |
| $F_{14}$ | $5.50 \times 10^{4}$ | $4.09 \times 10^{4}$ | $\mathbf{3.37 \times 10^{4}}$ | $4.34 \times 10^{4}$ | $5.29 \times 10^{4}$ | $5.22 \times 10^{4}$ | $1.98 \times 10^{5}$ | $9.20 \times 10^{4}$ | $2.31 \times 10^{5}$ | $1.22 \times 10^{5}$ |
| $F_{15}$ | $1.43 \times 10^{3}$ | $5.07 \times 10^{3}$ | $3.78 \times 10^{3}$ | $2.05 \times 10^{3}$ | $3.22 \times 10^{3}$ | $1.40 \times 10^{3}$ | $\mathbf{1.30 \times 10^{3}}$ | $2.03 \times 10^{3}$ | $1.50 \times 10^{3}$ | $2.33 \times 10^{3}$ |
| $F_{16}$ | $\mathbf{8.76 \times 10^{2}}$ | $2.08 \times 10^{3}$ | $1.88 \times 10^{3}$ | $1.33 \times 10^{3}$ | $1.13 \times 10^{3}$ | $1.12 \times 10^{3}$ | $9.82 \times 10^{2}$ | $9.45 \times 10^{2}$ | $1.79 \times 10^{3}$ | $2.17 \times 10^{3}$ |
| $F_{17}$ | $\mathbf{8.16 \times 10^{2}}$ | $1.34 \times 10^{3}$ | $1.29 \times 10^{3}$ | $1.23 \times 10^{3}$ | $1.20 \times 10^{3}$ | $1.24 \times 10^{3}$ | $1.08 \times 10^{3}$ | $1.18 \times 10^{3}$ | $1.26 \times 10^{3}$ | $1.43 \times 10^{3}$ |
| $F_{18}$ | $\mathbf{2.20 \times 10^{5}}$ | $7.02 \times 10^{5}$ | $6.30 \times 10^{5}$ | $4.40 \times 10^{5}$ | $4.15 \times 10^{5}$ | $2.30 \times 10^{5}$ | $2.65 \times 10^{5}$ | $3.95 \times 10^{5}$ | $5.76 \times 10^{5}$ | $1.03 \times 10^{6}$ |
| $F_{19}$ | $1.52 \times 10^{4}$ | $1.59 \times 10^{4}$ | $1.25 \times 10^{4}$ | $\mathbf{1.22 \times 10^{4}}$ | $1.42 \times 10^{4}$ | $1.31 \times 10^{4}$ | $1.56 \times 10^{4}$ | $1.67 \times 10^{4}$ | $1.71 \times 10^{4}$ | $1.66 \times 10^{4}$ |
| $F_{20}$ | $\mathbf{6.50 \times 10^{2}}$ | $1.15 \times 10^{3}$ | $1.22 \times 10^{3}$ | $1.24 \times 10^{3}$ | $1.21 \times 10^{3}$ | $1.13 \times 10^{3}$ | $8.33 \times 10^{2}$ | $1.30 \times 10^{3}$ | $1.25 \times 10^{3}$ | $1.37 \times 10^{3}$ |
| $F_{21}$ | $2.53 \times 10^{2}$ | $4.23 \times 10^{2}$ | $3.10 \times 10^{2}$ | $2.51 \times 10^{2}$ | $2.46 \times 10^{2}$ | $2.53 \times 10^{2}$ | $\mathbf{2.42 \times 10^{2}}$ | $3.17 \times 10^{2}$ | $3.37 \times 10^{2}$ | $3.19 \times 10^{2}$ |
| $F_{22}$ | $2.59 \times 10^{3}$ | $9.76 \times 10^{2}$ | $\mathbf{1.20 \times 10^{2}}$ | $9.66 \times 10^{2}$ | $9.41 \times 10^{2}$ | $3.21 \times 10^{3}$ | $5.13 \times 10^{3}$ | $5.53 \times 10^{3}$ | $6.74 \times 10^{3}$ | $4.72 \times 10^{3}$ |
| $F_{23}$ | $4.98 \times 10^{2}$ | $6.51 \times 10^{2}$ | $5.24 \times 10^{2}$ | $5.16 \times 10^{2}$ | $5.00 \times 10^{2}$ | $4.89 \times 10^{2}$ | $4.93 \times 10^{2}$ | $\mathbf{4.85 \times 10^{2}}$ | $5.09 \times 10^{2}$ | $5.49 \times 10^{2}$ |
| $F_{24}$ | $5.83 \times 10^{2}$ | $6.76 \times 10^{2}$ | $5.92 \times 10^{2}$ | $5.92 \times 10^{2}$ | $5.73 \times 10^{2}$ | $5.71 \times 10^{2}$ | $5.69 \times 10^{2}$ | $\mathbf{5.54 \times 10^{2}}$ | $5.90 \times 10^{2}$ | $6.34 \times 10^{2}$ |
| $F_{25}$ | $\mathbf{6.49 \times 10^{2}}$ | $7.65 \times 10^{2}$ | $7.33 \times 10^{2}$ | $6.74 \times 10^{2}$ | $6.61 \times 10^{2}$ | $6.50 \times 10^{2}$ | $6.83 \times 10^{2}$ | $6.55 \times 10^{2}$ | $6.57 \times 10^{2}$ | $6.55 \times 10^{2}$ |
| $F_{26}$ | $1.77 \times 10^{3}$ | $2.58 \times 10^{3}$ | $1.78 \times 10^{3}$ | $1.80 \times 10^{3}$ | $1.70 \times 10^{3}$ | $\mathbf{1.64 \times 10^{3}}$ | $1.86 \times 10^{3}$ | $1.73 \times 10^{3}$ | $2.17 \times 10^{3}$ | $2.71 \times 10^{3}$ |
| $F_{27}$ | $8.74 \times 10^{2}$ | $8.95 \times 10^{2}$ | $8.32 \times 10^{2}$ | $8.62 \times 10^{2}$ | $8.36 \times 10^{2}$ | $8.43 \times 10^{2}$ | $9.06 \times 10^{2}$ | $8.64 \times 10^{2}$ | $8.63 \times 10^{2}$ | $8.61 \times 10^{2}$ |
| $F_{28}$ | $6.96 \times 10^{2}$ | $9.06 \times 10^{2}$ | $8.25 \times 10^{2}$ | $7.40 \times 10^{2}$ | $7.15 \times 10^{2}$ | $6.77 \times 10^{2}$ | $6.63 \times 10^{2}$ | $6.57 \times 10^{2}$ | $6.62 \times 10^{2}$ | $\mathbf{6.41 \times 10^{2}}$ |
| $F_{29}$ | $6.39 \times 10^{2}$ | $1.62 \times 10^{3}$ | $1.09 \times 10^{3}$ | $8.34 \times 10^{2}$ | $7.60 \times 10^{2}$ | $\mathbf{6.07 \times 10^{2}}$ | $7.08 \times 10^{2}$ | $6.65 \times 10^{2}$ | $8.25 \times 10^{2}$ | $1.01 \times 10^{3}$ |
| $F_{30}$ | $\mathbf{2.19 \times 10^{6}}$ | $7.24 \times 10^{6}$ | $4.40 \times 10^{6}$ | $3.67 \times 10^{6}$ | $3.07 \times 10^{6}$ | $2.39 \times 10^{6}$ | $2.89 \times 10^{6}$ | $2.53 \times 10^{6}$ | $2.37 \times 10^{6}$ | $2.35 \times 10^{6}$ |
| *Rank* | **3.17** | 8.28 | 6.17 | 5.34 | 4.55 | 4.17 | 5.03 | 5.28 | 6.52 | 6.48 |

Taking a careful look at Table 8, we attain the following observations:

(1) In view of the Friedman test results, the PCLPSO with the adaptive $\boldsymbol{F}$ achieves the lowest rank and its rank value is much lower than those of the others. This demonstrates that the PCLPSO with adaptive strategy obtains the best overall performance, which demonstrates the great superiority of the adaptive strategy to the fixed ones.

(2) In-depth observations demonstrate that the PCLPSO with the adaptive strategy performs the best on 10 problems, while those with the fixed values obtain the best results on at most 3 problems. Moreover, the results obtained by the adaptive PCLPSO on the other 19 problems are very close to the best results obtained by PCLPSO with the associated optimal $\boldsymbol{F}$. In particular, we find that the optimal $\boldsymbol{F}$ for PCLPSO is different on different optimization problems.

In conclusion, the adaptive strategy for $\boldsymbol{F}$ not only helps PCLPSO achieve more promising performance, but also helps PCLPSO get out of being sensitive to the parameter $\boldsymbol{F}$. The great effectiveness of the adaptive strategy mainly benefits from the fact that it takes the difference among ***pbest***s into consideration to set the learning step ***F***. In this way, not only the search effectiveness, but also the search diversity of particles could be improved to a large extent, and thus PCLPSO with this strategy could achieve good performance.

### 4.3.3. Effectiveness of the Dynamic Strategy for *c*

At last, we executed experiments to validate the usefulness of the dynamic acceleration coefficient strategy (Equation (16)). To achieve this goal, we set ***c*** with different fixed values ranging from 0.8 to 2.0 with a step size of 0.2. Table 9 displays the comparison results in terms of the mean fitness values of the global best solutions found at the end of the associated algorithm over 30 independent runs between PCLPSO with the dynamic strategy for ***c*** and those with different fixed ***c*** on the 50-*D* CEC 2017 benchmark set.

**Table 9.** Comparison results between PCLPSO with and without the dynamic strategy for ***c*** on the 50-*D* CEC 2017 benchmark problems.

| ***F*** | **Dynamic-c** | **c = 0.8** | **c = 1.0** | **c = 1.2** | **c = 1.4** | **c = 1.6** | **c = 1.8** | **c = 2.0** |
|---|---|---|---|---|---|---|---|---|
| $F_1$ | $4.19 \times 10^{3}$ | $1.50 \times 10^{8}$ | $2.13 \times 10^{8}$ | $3.67 \times 10^{6}$ | $1.86 \times 10^{7}$ | **$1.06 \times 10^{3}$** | $2.50 \times 10^{3}$ | $5.93 \times 10^{3}$ |
| $F_3$ | $1.64 \times 10^{4}$ | $9.44 \times 10^{4}$ | $6.68 \times 10^{4}$ | $7.91 \times 10^{4}$ | $4.96 \times 10^{4}$ | $3.96 \times 10^{4}$ | $3.02 \times 10^{4}$ | **$1.35 \times 10^{4}$** |
| $F_4$ | **$3.53 \times 10^{2}$** | $4.75 \times 10^{2}$ | $5.85 \times 10^{2}$ | $4.38 \times 10^{2}$ | $4.16 \times 10^{2}$ | $3.69 \times 10^{2}$ | $3.92 \times 10^{2}$ | $4.02 \times 10^{2}$ |
| $F_5$ | **$5.17 \times 10^{1}$** | $1.09 \times 10^{2}$ | $9.86 \times 10^{1}$ | $9.60 \times 10^{1}$ | $8.23 \times 10^{1}$ | $6.69 \times 10^{1}$ | $8.42 \times 10^{1}$ | $9.64 \times 10^{1}$ |
| $F_6$ | **$8.50 \times 10^{1}$** | $7.12 \times 10^{0}$ | $4.87 \times 10^{0}$ | $3.37 \times 10^{0}$ | $2.92 \times 10^{0}$ | $2.02 \times 10^{0}$ | $1.58 \times 10^{0}$ | $1.96 \times 10^{0}$ |
| $F_7$ | $2.11 \times 10^{2}$ | $1.96 \times 10^{2}$ | $1.82 \times 10^{2}$ | $1.74 \times 10^{2}$ | **$1.55 \times 10^{2}$** | $1.72 \times 10^{2}$ | $2.07 \times 10^{2}$ | $2.19 \times 10^{2}$ |
| $F_8$ | **$5.39 \times 10^{1}$** | $1.08 \times 10^{2}$ | $9.07 \times 10^{1}$ | $8.67 \times 10^{1}$ | $7.04 \times 10^{1}$ | $7.45 \times 10^{1}$ | $8.66 \times 10^{1}$ | $1.17 \times 10^{+02}$ |
| $F_9$ | **$5.26 \times 10^{1}$** | $5.73 \times 10^{2}$ | $2.67 \times 10^{2}$ | $1.41 \times 10^{2}$ | $8.89 \times 10^{1}$ | $6.04 \times 10^{1}$ | $6.05 \times 10^{1}$ | $6.21 \times 10^{1}$ |
| $F_{10}$ | $1.09 \times 10^{4}$ | $8.61 \times 10^{3}$ | **$8.42 \times 10^{3}$** | $1.01 \times 10^{4}$ | $9.94 \times 10^{3}$ | $1.01 \times 10^{4}$ | $1.03 \times 10^{4}$ | $1.04 \times 10^{4}$ |
| $F_{11}$ | **$1.97 \times 10^{2}$** | $4.60 \times 10^{2}$ | $7.06 \times 10^{2}$ | $4.59 \times 10^{2}$ | $4.39 \times 10^{2}$ | $4.38 \times 10^{2}$ | $2.98 \times 10^{2}$ | $2.83 \times 10^{2}$ |
| $F_{12}$ | **$2.66 \times 10^{6}$** | $1.43 \times 10^{8}$ | $6.40 \times 10^{7}$ | $1.06 \times 10^{7}$ | $3.71 \times 10^{6}$ | $4.32 \times 10^{6}$ | $5.01 \times 10^{6}$ | $7.61 \times 10^{6}$ |
| $F_{13}$ | $3.90 \times 10^{3}$ | $1.41 \times 10^{5}$ | $2.98 \times 10^{4}$ | $1.53 \times 10^{4}$ | $8.39 \times 10^{3}$ | $8.84 \times 10^{3}$ | $4.74 \times 10^{3}$ | **$3.66 \times 10^{3}$** |
| $F_{14}$ | $5.50 \times 10^{4}$ | $1.10 \times 10^{5}$ | $8.56 \times 10^{4}$ | **$3.92 \times 10^{4}$** | $4.18 \times 10^{4}$ | $4.85 \times 10^{4}$ | $8.55 \times 10^{4}$ | $7.89 \times 10^{4}$ |
| $F_{15}$ | **$1.43 \times 10^{3}$** | $7.49 \times 10^{3}$ | $5.26 \times 10^{3}$ | $3.34 \times 10^{3}$ | $4.30 \times 10^{3}$ | $4.05 \times 10^{3}$ | $3.83 \times 10^{3}$ | $2.62 \times 10^{3}$ |
| $F_{16}$ | $8.76 \times 10^{2}$ | $8.37 \times 10^{2}$ | $7.80 \times 10^{2}$ | $7.92 \times 10^{2}$ | **$7.63 \times 10^{2}$** | $8.26 \times 10^{2}$ | $9.53 \times 10^{2}$ | $1.15 \times 10^{3}$ |
| $F_{17}$ | $8.16 \times 10^{2}$ | $7.62 \times 10^{2}$ | $6.27 \times 10^{2}$ | $6.45 \times 10^{2}$ | **$6.16 \times 10^{2}$** | $7.06 \times 10^{2}$ | $7.69 \times 10^{2}$ | $8.65 \times 10^{2}$ |
| $F_{18}$ | $2.20 \times 10^{5}$ | $1.53 \times 10^{6}$ | $1.40 \times 10^{6}$ | $9.63 \times 10^{5}$ | $5.95 \times 10^{5}$ | $3.35 \times 10^{5}$ | $2.72 \times 10^{5}$ | **$1.41 \times 10^{5}$** |
| $F_{19}$ | $1.52 \times 10^{4}$ | $1.57 \times 10^{4}$ | $1.49 \times 10^{4}$ | $1.52 \times 10^{4}$ | **$1.35 \times 10^{4}$** | $1.46 \times 10^{4}$ | $1.47 \times 10^{4}$ | $1.43 \times 10^{4}$ |
| $F_{20}$ | $6.50 \times 10^{2}$ | $4.16 \times 10^{2}$ | $3.49 \times 10^{2}$ | $3.61 \times 10^{2}$ | **$3.04 \times 10^{2}$** | $4.59 \times 10^{2}$ | $5.88 \times 10^{2}$ | $6.80 \times 10^{2}$ |
| $F_{21}$ | **$2.53 \times 10^{2}$** | $2.87 \times 10^{2}$ | $2.82 \times 10^{2}$ | $2.66 \times 10^{2}$ | $2.69 \times 10^{2}$ | $2.67 \times 10^{2}$ | $2.77 \times 10^{2}$ | $2.94 \times 10^{2}$ |
| $F_{22}$ | **$2.59 \times 10^{3}$** | $3.44 \times 10^{3}$ | $4.56 \times 10^{3}$ | $2.95 \times 10^{3}$ | $3.97 \times 10^{3}$ | $3.83 \times 10^{3}$ | $4.49 \times 10^{3}$ | $3.30 \times 10^{3}$ |
| $F_{23}$ | **$4.98 \times 10^{2}$** | $5.87 \times 10^{2}$ | $5.49 \times 10^{2}$ | $5.32 \times 10^{2}$ | $5.38 \times 10^{2}$ | $5.33 \times 10^{2}$ | $5.33 \times 10^{2}$ | $5.43 \times 10^{2}$ |
| $F_{24}$ | **$5.83 \times 10^{2}$** | $6.19 \times 10^{2}$ | $6.31 \times 10^{2}$ | $6.05 \times 10^{2}$ | $6.11 \times 10^{2}$ | $6.14 \times 10^{2}$ | $6.13 \times 10^{2}$ | $6.09 \times 10^{2}$ |
| $F_{25}$ | **$6.49 \times 10^{2}$** | $7.81 \times 10^{2}$ | $8.92 \times 10^{2}$ | $8.17 \times 10^{2}$ | $8.04 \times 10^{2}$ | $7.26 \times 10^{2}$ | $6.77 \times 10^{2}$ | $6.61 \times 10^{2}$ |
| $F_{26}$ | $1.77 \times 10^{3}$ | $3.34 \times 10^{3}$ | $2.07 \times 10^{3}$ | **$1.70 \times 10^{3}$** | $1.88 \times 10^{3}$ | $1.91 \times 10^{3}$ | $1.95 \times 10^{3}$ | $2.02 \times 10^{3}$ |
| $F_{27}$ | $8.74 \times 10^{2}$ | $8.61 \times 10^{2}$ | $8.69 \times 10^{2}$ | **$8.22 \times 10^{2}$** | $8.69 \times 10^{2}$ | $8.50 \times 10^{2}$ | $8.92 \times 10^{2}$ | $8.70 \times 10^{2}$ |
| $F_{28}$ | **$6.96 \times 10^{2}$** | $1.08 \times 10^{3}$ | $1.21 \times 10^{3}$ | $1.14 \times 10^{3}$ | $9.82 \times 10^{2}$ | $8.50 \times 10^{2}$ | $7.85 \times 10^{2}$ | $7.21 \times 10^{2}$ |
| $F_{29}$ | **$6.39 \times 10^{2}$** | $1.22 \times 10^{3}$ | $9.32 \times 10^{2}$ | $8.78 \times 10^{2}$ | $7.42 \times 10^{2}$ | $7.15 \times 10^{2}$ | $7.11 \times 10^{2}$ | $7.08 \times 10^{2}$ |
| $F_{30}$ | **$2.19 \times 10^{6}$** | $5.18 \times 10^{7}$ | $1.83 \times 10^{7}$ | $1.22 \times 10^{7}$ | $2.57 \times 10^{6}$ | $2.25 \times 10^{6}$ | $2.29 \times 10^{6}$ | $2.93 \times 10^{6}$ |
| *Rank* | **2.72** | 6.66 | 6.14 | 4.38 | 3.79 | 3.52 | 4.31 | 4.48 |

From Table 9, the following observations can be attained:

(1) In view of the Friedman test results, the PCLPSO with the proposed dynamic strategy achieves the lowest rank (2.72) and such a rank is much smaller than those of the PCLPSO (at least 3.52) with the fixed values. This verifies that the PCLPSO with the proposed dynamic strategy obtains the best overall performance and presents its great dominance to the PCLPSO with the fixed ***c***. This verifies the great superiority of the proposed dynamic strategy to the fixed one.

(2) Taking further observations, we find that the PCLPSO with the dynamic strategy obtains the best optimization results on 16 problems, while those with the fixed values obtain the best performance on at most 4 problems. Moreover, the results obtained by PCLPSO with the dynamic strategy on the other 13 problems are very similar to the best results obtained by PCLPSO with the associated optimal ***c***.

Based on the above experiments, the effectiveness of the proposed dynamic strategy is demonstrated. Such a strategy helps PCLPSO achieve promising performance because it can generate diversified values of ***c***, which is beneficial for further improving the search diversity of particles.

To summarize, the abovementioned experiments have comprehensively demonstrated the effectiveness of the proposed PCLPSO in solving optimization problems. In particular, PCLPSO performs much better than the compared peer methods in tackling complex problems, such as the multimodal problems, the hybrid problems, and the composition problems. The superiority of PCLPSO mainly profits from the proposed PCL strategy and the devised dynamic strategies for the learning step ***F*** and the acceleration coefficient ***c***, whose effectiveness was also verified by the experiments.

## 5. Conclusions

This paper devised a predominant cognitive learning particle swarm optimization (PCLPSO) to tackle optimization problems. Instead of letting particles learn from their own cognitive experience and the social experience of the entire swarm, the proposed PCLPSO constructs an effective guiding exemplar by the devised predominant cognitive learning (PCL) strategy to update each particle. Specifically, the guiding exemplar for each particle is constructed by letting its ***pbest*** learn from a predominant ***pbest*** randomly selected from those which are better than ***pbest*** of the updated particle. In this way, the constructed exemplar for each particle is expectedly more promising than its ***pbest***, and thus the search effectiveness of particles is expectedly improved. Moreover, due to the random selection of the predominant positions, the constructed guiding exemplars to update different particles are likely different, and thus the search diversity of particles is expectedly promoted as well. To further promote the search diversity and get rid of the sensitivity of PCLPSO to the related parameters, two dynamic strategies are particularly designed for the learning step in the exemplar construction and the acceleration coefficient in the velocity update. The proposed PCL and the devised dynamic strategies collaborate cohesively to help PCLPSO compromise the search effectiveness and the search diversity of particles well to search the solution space to obtain satisfactory performance.

Comparative experiments were carried out on the commonly adopted CEC 2017 benchmark set with three settings of dimensionality (30-*D*, 50-*D*, and 100-*D*) to compare the proposed PCLPSO with seven representative and state-of-the-art PSOs. Experimental results substantiated the great effectiveness of the devised PCLPSO and demonstrated that PCLPSO preserves a good scalability to solve different kinds of optimization problems. In particular, it was verified that the proposed PCLPSO preserves a good ability in tackling complex optimization problems, such as the multimodal problems, the hybrid problems, and the composition problems. To determine what contributes to the good performance of PCLPSO, deep investigations on PCLPSO were also carried out. The experimental results demonstrated that the proposed techniques contribute a lot to assisting PCLPSO to obtain good performance.

In the future, we aim to employ the proposed PCLPSO to tackle real-world optimization problems. Since PCLPSO is mainly designed for low-dimensional continuous optimization problems and it is independent of the mathematical properties of optimization problems, we mainly intend to use PCLPSO to solve continuous optimization problems in academic research and real-world engineering.

**Author Contributions:** Q.Y.: Conceptualization, supervision, methodology, formal analysis, and writing—original draft preparation. Y.J.: Implementation, formal analysis, and writing—original draft preparation. X.G.: Methodology, and writing—review and editing. D.X.: Methodology, and writing—review and editing. Z.L.: Writing—review and editing, and funding acquisition. S.-W.J.: Writing—review and editing. J.Z.: Conceptualization and writing—review and editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the National Natural Science Foundation of China under Grant 62006124 and U20B2061, in part by the Natural Science Foundation of Jiangsu Province under Project BK20200811, in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 20KJB520006, in part by the National Research Foundation of Korea (NRF-2021H1D3A2A01082705), and in part by the Startup Foundation for Introducing Talent of NUIST.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## References

1. Qiao, K.; Yu, K.; Qu, B.; Liang, J.; Song, H.; Yue, C. An Evolutionary Multitasking Optimization Framework for Constrained Multi-objective Optimization Problems. *IEEE Trans. Evol. Comput.* **2022**, *26*, 263–277. [CrossRef]
2. Yang, Q.; Chen, W.-N.; Zhang, J. Probabilistic Multimodal Optimization. In *Metaheuristics for Finding Multiple Solutions*; Preuss, M., Epitropakis, M.G., Li, X., Fieldsend, J.E., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 191–228.
3. Wan, X.; Cao, J.; Zhou, S.; Wang, J.; Zheng, N. Tracking Beyond Detection: Learning a Global Response Map for End-to-End Multi-Object Tracking. *IEEE Trans. Image Process.* **2021**, *30*, 8222–8235. [CrossRef] [PubMed]
4. Shen, J.; Yu, D.; Deng, L.; Dong, X. Fast Online Tracking with Detection Refinement. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 162–173. [CrossRef]
5. Xu, L.; Xu, H.; Li, X.; Pan, M. A Defect Inspection for Explosive Cartridge Using an Improved Visual Attention and Image-Weighted Eigenvalue. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 1191–1204. [CrossRef]
6. Zhou, X.; Wang, Y.; Zhu, Q.; Mao, J.; Xiao, C.; Lu, X.; Zhang, H. A Surface Defect Detection Framework for Glass Bottle Bottom Using Visual Attention Model and Wavelet Transform. *IEEE Trans. Ind. Informatics* **2019**, *16*, 2189–2201. [CrossRef]
7. Hu, L.; Naeem, W.; Rajabally, E.; Watson, G.; Mills, T.; Bhuiyan, Z.; Raeburn, C.; Salter, I.; Pekcan, C. A Multiobjective Optimization Approach for COLREGs-Compliant Path Planning of Autonomous Surface Vehicles Verified on Networked Bridge Simulators. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1167–1179. [CrossRef]
8. Wang, L.; Liu, L.; Qi, J.; Peng, W. Improved Quantum Particle Swarm Optimization Algorithm for Offline Path Planning in AUVs. *IEEE Access* **2020**, *8*, 143397–143411. [CrossRef]
9. Chen, W.-N.; Tan, D.-Z.; Yang, Q.; Gu, T.; Zhang, J. Ant Colony Optimization for the Control of Pollutant Spreading on Social Networks. *IEEE Trans. Cybern.* **2019**, *50*, 4053–4065. [CrossRef]
10. Yang, Q.; Chen, W.-N.; Gu, T.; Jin, H.; Mao, W.; Zhang, J. An Adaptive Stochastic Dominant Learning Swarm Optimizer for High-Dimensional Optimization. *IEEE Trans. Cybern.* **2022**, *52*, 1960–1976. [CrossRef]
11. Wei, F.-F.; Chen, W.-N.; Yang, Q.; Deng, J.; Luo, X.-N.; Jin, H.; Zhang, J. A Classifier-Assisted Level-Based Learning Swarm Optimizer for Expensive Optimization. *IEEE Trans. Evol. Comput.* **2020**, *25*, 219–233. [CrossRef]
12. Liang, J.J.; Qu, B.; Gong, D.; Yue, C. Problem Definitions and Evaluation Criteria for the CEC 2019 Special Session on Multimodal Multiobjective Optimization. *Comput. Intell. Lab. Zhengzhou Univ.* **2019**. [CrossRef]
13. Bisio, I.; Estatico, C.; Fedeli, A.; Lavagetto, F.; Pastorino, M.; Randazzo, A.; Sciarrone, A. Brain Stroke Microwave Imaging by Means of a Newton-Conjugate-Gradient Method in $L^p$ Banach Spaces. *IEEE Trans. Microw. Theory Tech.* **2018**, *66*, 3668–3682. [CrossRef]
14. Jain, P.; Kakade, S.M.; Kidambi, R.; Netrapalli, P.; Sidford, A. Accelerating Stochastic Gradient Descent for Least Squares Regression. In Proceedings of the 31st Conference on Learning Theory, Stockholm, Sweden, 6–9 July 2018; PMLR, Inc.: Brookline, MA, USA; pp. 545–604.

15. Lera, D.; Posypkin, M.; Sergeyev, Y.D. Space-filling curves for numerical approximation and visualization of solutions to systems of nonlinear inequalities with applications in robotics. *Appl. Math. Comput.* **2020**, *390*, 125660. [CrossRef]
16. Strongin, R.G.; Sergeyev, Y.D. Global multidimensional optimization on parallel computer. *Parallel Comput.* **1992**, *18*, 1259–1273. [CrossRef]
17. Chang, D.; Sun, S.; Zhang, C. An Accelerated Linearly Convergent Stochastic L-BFGS Algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3338–3346. [CrossRef]
18. Karimi, S.; Vavasis, S. IMRO: A Proximal Quasi-Newton Method for Solving $\ell$1-Regularized Least Squares Problems. *SIAM J. Optim.* **2017**, *27*, 583–615. [CrossRef]
19. Yu, P.; Pong, T.K.; Lu, Z. Convergence Rate Analysis of a Sequential Convex Programming Method with Line Search for a Class of Constrained Difference-of-Convex Optimization Problems. *SIAM J. Optim.* **2021**, *31*, 2024–2054. [CrossRef]
20. Malik, M.; Mamat, M.; Abas, S.S. Convergence Analysis of a New Coefficient Conjugate Gradient Method under Exact Line Search. *Int. J. Adv. Sci. Technol.* **2020**, *29*, 187–198.
21. Vaswani, S.; Mishkin, A.; Laradji, I.; Schmidt, M.; Gidel, G.; Lacoste-Julien, S. Painless Stochastic Gradient: Interpolation, Line-search, and Convergence Rates. In Proceedings of the 33rd Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32, pp. 1–14.
22. Babaie-Kafaki, S.; Rezaee, S. A randomized nonmonotone adaptive trust region method based on the simulated annealing strategy for unconstrained optimization. *Int. J. Intell. Comput. Cybern.* **2019**, *12*, 389–399. [CrossRef]
23. Shani, L.; Efroni, Y.; Mannor, S. Adaptive Trust Region Policy Optimization: Global Convergence and Faster Rates for Regularized MDPs. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 5668–5675.
24. Yang, Q.; Bian, Y.-W.; Gao, X.-D.; Xu, D.-D.; Lu, Z.-Y.; Jeon, S.-W.; Zhang, J. Stochastic Triad Topology Based Particle Swarm Optimization for Global Numerical Optimization. *Mathematics* **2022**, *10*, 1032. [CrossRef]
25. Yang, Q.; Hua, L.; Gao, X.; Xu, D.; Lu, Z.; Jeon, S.-W.; Zhang, J. Stochastic Cognitive Dominance Leading Particle Swarm Optimization for Multimodal Problems. *Mathematics* **2022**, *10*, 761. [CrossRef]
26. Yang, Q.; Xie, H.; Chen, W.; Zhang, J. Multiple Parents Guided Differential Evolution for Large Scale Optimization. In Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, 24–29 July 2016; pp. 3549–3556.
27. Yu, W.-J.; Ji, J.-Y.; Gong, Y.-J.; Yang, Q.; Zhang, J. A tri-objective differential evolution approach for multimodal optimization. *Inf. Sci.* **2018**, *423*, 1–23. [CrossRef]
28. Seo, J.-H.; Im, C.-H.; Heo, C.-G.; Kim, J.-K.; Jung, H.-K.; Lee, C.-G. Multimodal function optimization based on particle swarm optimization. *IEEE Trans. Magn.* **2006**, *42*, 1095–1098. [CrossRef]
29. Ji, X.; Zhang, Y.; Gong, D.; Sun, X.; Guo, Y. Multisurrogate-Assisted Multitasking Particle Swarm Optimization for Expensive Multimodal Problems. *IEEE Trans. Cybern.* **2021**, 1–15. [CrossRef]
30. Yang, Q.; Chen, W.-N.; Yu, Z.; Gu, T.; Li, Y.; Zhang, H.; Zhang, J. Adaptive Multimodal Continuous Ant Colony Optimization. *IEEE Trans. Evol. Comput.* **2016**, *21*, 191–205. [CrossRef]
31. Yang, Q.; Chen, W.-N.; Li, Y.; Chen, C.L.P.; Xu, X.-M.; Zhang, J. Multimodal Estimation of Distribution Algorithms. *IEEE Trans. Cybern.* **2016**, *47*, 636–650. [CrossRef]
32. Yue, C.; Qu, B.; Liang, J. A Multiobjective Particle Swarm Optimizer Using Ring Topology for Solving Multimodal Multiobjective Problems. *IEEE Trans. Evol. Comput.* **2017**, *22*, 805–817. [CrossRef]
33. Qu, B.; Li, G.; Yan, L.; Liang, J.; Yue, C.; Yu, K.; Crisalle, O.D. A grid-guided particle swarm optimizer for multimodal multi-objective problems. *Appl. Soft Comput.* **2022**, *117*, 108381. [CrossRef]
34. Jones, D.R. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *J. Glob. Optim.* **2001**, *21*, 345–383. [CrossRef]
35. Kvasov, D.E.; Pizzuti, C.; Sergeyev, Y.D. Local tuning and partition strategies for diagonal GO methods. *Numer. Math.* **2003**, *94*, 93–106. [CrossRef]
36. Sergeyev, Y.D.; Grishagin, V.A. Parallel Asynchronous Global Search and the Nested Optimization Scheme. *J. Comput. Anal. Appl.* **2001**, *3*, 123–145. [CrossRef]
37. Remigijus, P.; Sergeev, Y.; Kvasov, D.; Julius, Ž. Globally-biased BIRECT Algorithm with Local Accelerators for Expensive Global Optimization. *Expert Syst. Appl.* **2020**, *144*, 113052.
38. Vikhar, P.A. Evolutionary Algorithms: A Critical Review and Its Future Prospects. In Proceedings of the International Conference on Global Trends in Signal Processing, Information Computing and Communication, Jalgaon, India, 22–24 December 2016; pp. 261–265.
39. Sloss, A.N.; Gustafson, S. 2019 Evolutionary Algorithms Review. In *Genetic Programming Theory and Practice XVII, Banzhaf, W., Goodman, E., Sheneman, L., Trujillo, L., Worzel, B., Eds.*; Springer International Publishing: Cham, Switzerland, 2020; pp. 307–344.
40. Coello, C.A.C.; Lamont, G.B.; Van Veldhuizen, D.A. *Evolutionary Algorithms for Solving Multi-Objective Problems*; Springer: Berlin, Germany, 2007; Volume 5.
41. Slowik, A.; Kwasnicka, H. Evolutionary algorithms and their applications to engineering problems. *Neural Comput. Appl.* **2020**, *32*, 12363–12379. [CrossRef]
42. Kvasov, D.; Sergeyev, Y. Deterministic approaches for solving practical black-box global optimization problems. *Adv. Eng. Softw.* **2015**, *80*, 58–66. [CrossRef]

43. Sergeyev, Y.D.; Kvasov, D.E.; Mukhametzhanov, M. On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget. *Sci. Rep.* **2018**, *8*, 453. [CrossRef]
44. Sergeyev, Y.D.; Kvasov, D.E. A deterministic global optimization using smooth diagonal auxiliary functions. *Commun. Nonlinear Sci. Numer. Simul.* **2014**, *21*, 99–111. [CrossRef]
45. Zhang, J.; Yang, F.; Weng, X. An Evolutionary Scatter Search Particle Swarm Optimization Algorithm for the Vehicle Routing Problem with Time Windows. *IEEE Access* **2018**, *6*, 63468–63485. [CrossRef]
46. Yang, Q.; Chen, W.-N.; Gu, T.; Zhang, H.; Yuan, H.; Kwong, S.; Zhang, J. A Distributed Swarm Optimizer with Adaptive Communication for Large-Scale Optimization. *IEEE Trans. Cybern.* **2020**, *50*, 3393–3408. [CrossRef]
47. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
48. Eberhart, R.; Kennedy, J. A New Optimizer Using Particle Swarm Theory. In Proceedings of the International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
49. Tang, J.; Liu, G.; Pan, Q. A Review on Representative Swarm Intelligence Algorithms for Solving Optimization Problems: Applications and Trends. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 1627–1643. [CrossRef]
50. Okulewicz, M.; Mańdziuk, J. The impact of particular components of the PSO-based algorithm solving the Dynamic Vehicle Routing Problem. *Appl. Soft Comput.* **2017**, *58*, 586–604. [CrossRef]
51. Han, H.; Lu, W.; Hou, Y.; Qiao, J. An Adaptive-PSO-Based Self-Organizing RBF Neural Network. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 104–117. [CrossRef]
52. Guo, L. Research on Anomaly Detection in Massive Multimedia Data Transmission Network Based on Improved PSO Algorithm. *IEEE Access* **2020**, *8*, 95368–95377. [CrossRef]
53. Wu, J.; Song, C.; Ma, J.; Wu, J.; Han, G. Reinforcement Learning and Particle Swarm Optimization Supporting Real-Time Rescue Assignments for Multiple Autonomous Underwater Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2021**, 1–14. [CrossRef]
54. Wang, P.; Lei, Y.; Agbedanu, P.R.; Zhang, Z. Makespan-Driven Workflow Scheduling in Clouds Using Immune-Based PSO Algorithm. *IEEE Access* **2020**, *8*, 29281–29290. [CrossRef]
55. Gong, Y.-J.; Li, J.-J.; Zhou, Y.; Li, Y.; Chung, H.S.-H.; Shi, Y.-H.; Zhang, J. Genetic Learning Particle Swarm Optimization. *IEEE Trans. Cybern.* **2015**, *46*, 2277–2290. [CrossRef]
56. 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]
57. Zhan, Z.; Zhang, J.; Li, Y.; Shi, Y. Orthogonal Learning Particle Swarm Optimization. *IEEE Trans. Evol. Comput.* **2011**, *15*, 832–847. [CrossRef]
58. Mendes, R.; Kennedy, J.; Neves, J. The Fully Informed Particle Swarm: Simpler, Maybe Better. *IEEE Trans. Evol. Comput.* **2004**, *8*, 204–210. [CrossRef]
59. Zhang, X.; Liu, H.; Zhang, T.; Wang, Q.; Wang, Y.; Tu, L. Terminal crossover and steering-based particle swarm optimization algorithm with disturbance. *Appl. Soft Comput.* **2019**, *85*, 105841. [CrossRef]
60. Yang, Q.; Chen, W.-N.; Gu, T.; Zhang, H.; Deng, J.D.; Li, Y.; Zhang, J. Segment-Based Predominant Learning Swarm Optimizer for Large-Scale Optimization. *IEEE Trans. Cybern.* **2016**, *47*, 2896–2910. [CrossRef]
61. Zhang, K.; Huang, Q.; Zhang, Y. Enhancing comprehensive learning particle swarm optimization with local optima topology. *Inf. Sci.* **2018**, *471*, 1–18. [CrossRef]
62. Lynn, N.; Suganthan, P. Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. *Swarm Evol. Comput.* **2015**, *24*, 11–24. [CrossRef]
63. Wang, Z.J.; Zhan, Z.H.; Du, K.J.; Yu, Z.W.; Zhang, J. Orthogonal Learning Particle Swarm Optimization with Variable Relocation for Dynamic Optimization. In Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, 24–29 July 2016; pp. 594–600.
64. Guo, Q.; Ba, J.; Luo, C.; Xiao, S. Stability-enhanced prestack seismic inversion using hybrid orthogonal learning particle swarm optimization. *J. Pet. Sci. Eng.* **2020**, *192*, 107313. [CrossRef]
65. Lin, A.; Sun, W.; Yu, H.; Wu, G.; Tang, H. Global genetic learning particle swarm optimization with diversity enhancement by ring topology. *Swarm Evol. Comput.* **2018**, *44*, 571–583. [CrossRef]
66. Chen, Y.; Li, L.; Peng, H.; Xiao, J.; Yang, Y.; Shi, Y. Particle swarm optimizer with two differential mutation. *Appl. Soft Comput.* **2017**, *61*, 314–330. [CrossRef]
67. Shen, Y.; Wei, L.; Zeng, C.; Chen, J. Particle Swarm Optimization with Double Learning Patterns. *Comput. Intell. Neurosci.* **2016**, *2016*, 3049632. [CrossRef]
68. Zhang, X.; Wang, X.; Kang, Q.; Cheng, J. Differential mutation and novel social learning particle swarm optimization algorithm. *Inf. Sci.* **2018**, *480*, 109–129. [CrossRef]
69. 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.* **2018**, *23*, 718–731. [CrossRef]
70. Song, G.W.; Yang, Q.; Gao, X.D.; Ma, Y.Y.; Lu, Z.Y.; Zhang, J. An Adaptive Level-Based Learning Swarm Optimizer for Large-Scale Optimization. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne, Australia, 17–20 October 2021; pp. 152–159.

71. Han-Yu, X.; Yang, Q.; Xiao-Min, H.; Chen, W.N. Cross-generation Elites Guided Particle Swarm Optimization for large scale optimization. In Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016; pp. 1–8.
72. Yang, Q.; Li, Y.; Gao, X.-D.; Ma, Y.-Y.; Lu, Z.-Y.; Jeon, S.-W.; Zhang, J. An Adaptive Covariance Scaling Estimation of Distribution Algorithm. *Mathematics* **2021**, *9*, 3207. [CrossRef]
73. Awad, N.H.; Ali, M.Z.; Liang, J.J.; Qu, B.Y.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrain Real-Parameter Numerical Optimization*; Nanyang Technological University: Singapore, November 2016.
74. Sennan, S.; Ramasubbareddy, S.; Balasubramaniyam, S.; Nayyar, A.; Abouhawwash, M.; Hikal, N.A. T2FL-PSO: Type-2 Fuzzy Logic-Based Particle Swarm Optimization Algorithm Used to Maximize the Lifetime of Internet of Things. *IEEE Access* **2021**, *9*, 63966–63979. [CrossRef]
75. Meng, Z.; Zhong, Y.; Mao, G.; Liang, Y. PSO-sono: A novel PSO variant for single-objective numerical optimization. *Inf. Sci.* **2021**, *586*, 176–191. [CrossRef]
76. Wang, S.; Liu, G.; Gao, M.; Cao, S.; Guo, A.; Wang, J. Heterogeneous comprehensive learning and dynamic multi-swarm particle swarm optimizer with two mutation operators. *Inf. Sci.* **2020**, *540*, 175–201. [CrossRef]
77. Lin, A.; Sun, W.; Yu, H.; Wu, G.; Tang, H. Adaptive comprehensive learning particle swarm optimization with cooperative archive. *Appl. Soft Comput.* **2019**, *77*, 533–546. [CrossRef]
78. Liu, R.; Wang, L.; Ma, W.; Mu, C.; Jiao, L. Quadratic interpolation based orthogonal learning particle swarm optimization algorithm. *Nat. Comput.* **2013**, *13*, 17–37. [CrossRef]
79. Xia, X.; Gui, L.; Yu, F.; Wu, H.; Wei, B.; Zhang, Y.-L.; Zhan, Z.-H. Triple Archives Particle Swarm Optimization. *IEEE Trans. Cybern.* **2019**, *50*, 4862–4875. [CrossRef]
80. Zou, J.; Deng, Q.; Zheng, J.; Yang, S. A close neighbor mobility method using particle swarm optimizer for solving multimodal optimization problems. *Inf. Sci.* **2020**, *519*, 332–347. [CrossRef]
81. Blackwell, T.; Kennedy, J. Impact of Communication Topology in Particle Swarm Optimization. *IEEE Trans. Evol. Comput.* **2018**, *23*, 689–702. [CrossRef]
82. Liu, Q.; Wei, W.; Yuan, H.; Zhan, Z.-H.; Li, Y. Topology selection for particle swarm optimization. *Inf. Sci.* **2016**, *363*, 154–173. [CrossRef]
83. Wang, L.; Yang, B.; Orchard, J. Particle swarm optimization using dynamic tournament topology. *Appl. Soft Comput.* **2016**, *48*, 584–596. [CrossRef]
84. Liu, W.; Wang, Z.; Yuan, Y.; Zeng, N.; Hone, K.; Liu, X. A Novel Sigmoid-Function-Based Adaptive Weighted Particle Swarm Optimizer. *IEEE Trans. Cybern.* **2019**, *51*, 1085–1093. [CrossRef]
85. Varna, F.T.; Husbands, P. HIDMS-PSO: A New Heterogeneous Improved Dynamic Multi-Swarm PSO Algorithm. In Proceedings of the IEEE Symposium Series on Computational Intelligence, Canberra, Australia, 1–4 December 2020; pp. 473–480.
86. Zhang, H.; Yuan, M.; Liang, Y.; Liao, Q. A novel particle swarm optimization based on prey–predator relationship. *Appl. Soft Comput.* **2018**, *68*, 202–218. [CrossRef]