



Article

Stochastic Triad Topology Based Particle Swarm Optimization for Global Numerical Optimization

Qiang Yang¹ , Yu-Wei Bian¹, Xu-Dong Gao^{1,*}, Dong-Dong Xu¹, Zhen-Yu Lu¹, Sang-Woon Jeon² 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.); 201983460001@nuist.edu.cn (Y.-W.B.); 001601@nuist.edu.cn (D.-D.X.); 001114@nuist.edu.cn (Z.-Y.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: 003103@nuist.edu.cn

Abstract: Particle swarm optimization (PSO) has exhibited well-known feasibility in problem optimization. However, its optimization performance still encounters challenges when confronted with complicated optimization problems with many local areas. In PSO, the interaction among particles and utilization of the communication information play crucial roles in improving the learning effectiveness and learning diversity of particles. To promote the communication effectiveness among particles, this paper proposes a stochastic triad topology to allow each particle to communicate with two random ones in the swarm via their personal best positions. Then, unlike existing studies that employ the personal best positions of the updated particle and the neighboring best position of the topology to direct its update, this paper adopts the best one and the mean position of the three personal best positions in the associated triad topology as the two guiding exemplars to direct the update of each particle. To further promote the interaction diversity among particles, an archive is maintained to store the obsolete personal best positions of particles and is then used to interact with particles in the triad topology. To enhance the chance of escaping from local regions, a random restart strategy is probabilistically triggered to introduce initialized solutions to the archive. To alleviate sensitivity to parameters, dynamic adjustment strategies are designed to dynamically adjust the associated parameter settings during the evolution. Integrating the above mechanism, a stochastic triad topology-based PSO (STTPSO) is developed to effectively search complex solution space. With the above techniques, the learning diversity and learning effectiveness of particles are largely promoted and thus the developed STTPSO is expected to explore and exploit the solution space appropriately to find high-quality solutions. Extensive experiments conducted on the commonly used CEC 2017 benchmark problem set with different dimension sizes substantiate that the proposed STTPSO achieves highly competitive or even much better performance than state-of-the-art and representative PSO variants.

Keywords: particle swarm optimization; stochastic triad topology; guiding exemplar; multimodal problems; global optimization

MSC: 68-04; 65-04



Citation: 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. <https://doi.org/10.3390/math10071032>

Academic Editor: Janez Žerovnik

Received: 21 February 2022

Accepted: 21 March 2022

Published: 24 March 2022

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



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

1. Introduction

Particle swarm optimization (PSO) has witnessed tremendous success in solving optimization problems, especially non-differentiable ones [1–5], since its advent in 1995 [6,7]. Specifically, it maintains a swarm of particles, each of which represents a feasible solution, to iteratively search the solution space to find the global optima. Due to its good

interpretability and great convenience of implementation [8–10], PSO has been widely applied to solve various real-world problems in daily life and industrial engineering, such as supply chain network design [11], control of pollutant spreading on social networks [12], and industrial power load forecasting [13].

In the classical PSO [6,7], a fully connected topology with all particles is utilized to select guiding exemplars for particles to update, leading to the global best position (usually denoted as *gbest*) discovered by the whole swarm being shared by all particles. As a result, the learning diversity of particles is limited and thus the swarm falls into local areas when dealing with multimodal problems. To alleviate this limitation, researchers have paid extensive attention to designing novel learning strategies [14–19] to improve the optimization effectiveness of PSO.

In essence, the key to the learning strategies in PSO lies in the selection of guiding exemplars to direct the update of particles [17,20]. Broadly speaking, existing exemplar selection mechanisms can be classified into two categories, namely topology-based exemplar selection methods [15,16,21–23], and exemplar construction approaches [17–19,24,25].

Topology-based exemplar selection methods have been widely utilized in the research of PSO. In most cases, these methods aim to determine a less greedy exemplar to replace the social exemplar, namely *gbest*, in the classical PSO [6,7]. Based on different topologies, an abundance of remarkable PSO variants have been developed [26–28], such as ring topology structure [26], pyramid topology structure [27], Von Neumann topology structure [29], random topology [22], etc. Different topologies usually preserve different characteristics and merits. Therefore, a natural idea is to hybridize them to ensemble the merits of different topologies to improve the optimization performance of PSO. Along this line, many PSO variants [28,30–33] have been developed based on different methods of hybridization. In addition, to alleviate the shortcoming of static topologies where each particle can only interact with fixed peers, some researchers further proposed dynamic topologies [34–36] to dynamically change the topologies (the topology type or the topology size) based on the evolution state of the swarm.

Although topology-based PSO variants have shown to be highly promising in solving optimization performance, the guiding exemplars selected by different topologies to direct the update of particles are all the historical promising positions found by particles. Therefore, the learning effectiveness of particles is limited by the historical positions [14,18,37]. Once all the historical positions converge to local areas, it is difficult for the swarm to jump out of the local basin. To alleviate this issue, researchers have attempted to develop novel PSOs from another perspective, namely constructing new guiding exemplars for particles to learn from [14,38].

Different from topology-based exemplar selection methods, exemplar construction methods generally build new guiding exemplars for particles by combining dimensions of historical positions. In general, it is highly possible that the built exemplars are not visited by particles. The most representative PSO variant in this direction is the comprehensive learning PSO (CLPSO) [17], which constructs a new guiding exemplar dimension by dimension from the personal best positions of all particles. Inspired from this method of constructing new exemplars, researchers have developed many other construction approaches, such as orthogonal learning PSO (OLPSO) [37], and genetic learning PSO (GLPSO) [18].

Although most existing PSO variants have shown very promising performances in simple optimization problems, such as unimodal problems and simple multimodal problems, their performance is confronted with great challenges or even deteriorates dramatically when dealing with complicated optimization problems, such as multimodal problems with many wide and flat local areas [39,40], and composition problems with many interacting variables. However, in the era of big data and Internet of Things (IoT) [41], optimization problems become increasingly complicated, which are ubiquitous in every field in daily life and industrial engineering [11,13]. As a consequence, the optimization ability of PSO to solve complicated optimization problems warrants urgent and careful research, rendering ongoing research into PSO an important frontier in evolutionary community [42,43].

To improve the optimization effectiveness of PSO in solving complicated problems, this paper proposes a stochastic triad topology-based PSO (STTPSO). Specifically, during the evolution, for each particle, two personal best positions are first randomly selected from those of the rest particles. Then, the personal best position of the particle to be updated and the two random best positions form a triad topology. Based on this topology, the best one in the topology and the mean position of the topology are employed as the guiding exemplars to direct the update of the particle. In this way, each particle likely preserves different guiding exemplars and thus the learning diversity of particles can be largely improved, which is beneficial for the swarm to escape from local areas.

Overall, the main components of the proposed STTPSO are summarized as follows:

- (1) A stochastic triad topology is employed to connect the personal best position of each particle and two different personal best positions randomly selected from those of the rest particles to select guiding exemplars for particles to update. Different from existing studies [22,37], which only utilize the topologies to determine the best position to replace the social exemplar, namely *gbest*, in the classical PSO (with another guiding exemplar as the personal best position of the particle), the proposed STTPSO utilizes the stochastic triad topology to select the best one and computes the mean position of the triad best positions as the two guiding exemplars to direct the update of each particle. Since the topology is stochastic, it is likely that different particles preserve different guiding exemplars. As a result, the learning diversity of particles can be largely promoted, and thus the probability of the swarm escaping from local areas can be promoted.
- (2) An archive is maintained to store the obsolete personal best positions and then is combined with the personal best positions of all particles in the current generation to form the triad topologies for particles. In this way, valuable historical information can be utilized to direct the update of particles, which is helpful for improving swarm diversity.
- (3) A random restart strategy is designed by randomly initializing a solution with a small probability. However, instead of employing this restart strategy on the swarm, we utilize it on the archive. That is to say, a randomly initialized solution is inserted into the archive with a small probability. In this way, the swarm diversity can be promoted without significant sacrifice of convergence speed.
- (4) A dynamic strategy for the acceleration coefficients is devised to alleviate the sensitivity of STTPSO. Instead of utilizing fixed values for the two acceleration coefficients, this paper randomly samples the two acceleration coefficients based on the Gaussian distribution with the mean value set as the classical setting of the two coefficients and a small deviation. With this dynamic strategy, different particles can have different settings, and thus the learning diversity can be further promoted.

The above four components collaborate cohesively to help STTPSO explore and exploit the solution appropriately to locate the optima of optimization problems. In order to verify the effectiveness of the proposed STTPSO, extensive experiments are conducted on the widely used CEC 2017 benchmark problem set [44] with three different dimension sizes by comparing STTPSO with several state-of-the-art and popular PSO variants. In addition, deep investigations on the devised STTPSO are also conducted to observe the influence of each component in STTPSO.

The rest of this paper is organized as follows. Closely related studies are reviewed briefly in Section 2. In Section 3, the proposed STTPSO is elucidated in detail. Then, extensive comparison experiments and analysis of the associated results are conducted and discussed in Section 4. Finally, the conclusion of this paper is provided in Section 5.

2. Related Works

2.1. Basic PSO

PSO is a heuristic search algorithm and was first proposed in 1995 by Kennedy and Eberhart [6,7]. Specifically, PSO maintains a population of particles to search the solution

space, and each particle is represented by a position vector x_i and a velocity vector v_i , where the position vector represents a feasible solution to the problem, while the velocity vector represents the moving direction of the particle. Moreover, each particle memorizes its own historical best position $pbest_i$, and the global best position $gbest$ of the whole population is also memorized during the evolution. Then, each particle is updated by cognitively learning from its own experience, namely its personal best position $pbest_i$, and socially learning from the experience of the whole swarm, namely the global best position $gbest$. Specifically, each particle is updated as follows:

$$v_i = wv_i + c_1r_1(pbest_i - x_i) + c_2r_2(gbest - x_i) \quad (1)$$

$$x_i = x_i + v_i \quad (2)$$

where w is the inertia weight, c_1 and c_2 are called acceleration coefficients, and r_1 and r_2 are two real random numbers uniformly sampled within $[0, 1]$.

In Equation (1), the first part in the right hand is the “inertia part”, which controls the memory of the velocity of each particle in the last generation. The second part is the “self-cognition” part, where each particle learns from its own experience. The third part is the social part, where each particle learns from the experience of the whole swarm.

As for the inertia weight w , in the literature [17,18,37,45], a linear decay method is widely utilized in PSO variants, which is presented below:

$$w = 0.9 - 0.5 \times \frac{fes}{FE_{max}} \quad (3)$$

In the literature [10,17,18,37,45–47], it is widely recognized that the learning strategy in Equation (1) plays the most important role in helping PSO achieve satisfactory performance. As a result, researchers have been devoted to designing novel effective learning strategies for PSO to improve its optimization ability.

2.2. Advanced Learning Strategies for PSO

To improve the optimization performance of PSO, many researchers have designed an ocean of effective learning strategies to improve the learning diversity and the learning effectiveness of particles [18,48–51]. As far as we are concerned, existing learning strategies for PSO could be classified into two main categories as shown in Table 1, namely topology-based learning strategies [22,26,52], and exemplar construction-based learning strategies [14,17,37,38,53].

Topology-based learning strategies [21,22,26,54,55], mainly utilize a specific topology to interact with particles to select guiding exemplars to update particles. In fact, the classical PSO [6,7] introduced above is a topology-based learning PSO, where the topology is the full topology connecting all particles. Such a full topology usually leads to an excessively greedy guiding exemplar (namely the global best position $gbest$), which likely attracts particles into local areas. To alleviate this issue, many researchers have developed many kinds of local topologies to select less greedy guiding exemplars to direct the update of particles. For instance, in [26], the ring topology was utilized to organize particles into a ring, and then each particle interacts with its two neighbors to select one guiding exemplar to replace $gbest$ in Equation (1). In [27], the pyramid topology with a three-dimensional wire-frame triangle was used to select the guiding exemplars for particles. In [29,55], the star topology was employed for particles to interact with others. In this topology, the central node shares information with other particles, and other particles also share information with the central node. Therefore, the communication is a two-way information exchange. In [29], the Von Neumann topology which is a two-dimensional lattice, was adopted to select guiding exemplars. Specifically, this topology connects the top, bottom, left and right neighbors of each point to form a neighborhood topology of size five. Such a topology can be regarded as a “two-dimensional” ring topology derived from a one-dimensional line into a two-dimensional plane.

Table 1. The rough classification of existing PSO variants.

Category		Methods	Characteristics
Topology-based Methods	Static Topology	Full Topology	PSO [6,7]
		Ring Topology	MRTPSO [26], GGL-PSOD [56]
		Pyramid Topology	PMKPSO [27]
		Star Topology	PSO-Star [29,55]
		Von Neumann Topology	PSO-Von-Neumann [29]
	Hybrid Topology	XPSO [23]	
	Dynamic Topology	Dynamic Topology	DNPSO [28], DMSPSO [15], SPSO [22]
Exemplar construction-based Methods	Random Construction	CLPSO_LS [14], CLPSO [17], HCLPSO [25], TCSPSO [19]	Randomly recombine dimensions of personal best positions. The exemplar construction efficiency is low, but it consumes no fitness evaluations in exemplar construction.
	Operator-based Construction	MPSOEG [24], GLPSO [18]	Recombine dimensions of personal best positions based evolutionary operators in other EAs. The exemplar construction efficiency is high, but it consumes many fitness evaluations in exemplar construction
	Orthogonal Recombination	OLPSO [37]	Recombine dimensions of personal best positions based on orthogonal experimental design. The exemplar construction efficiency is high, but it consumes a lot of fitness evaluation in exemplar construction

All topologies mentioned above are static topologies. In these topologies, each particle interacts with fixed peers during the evolution, and thus they bear limitations in improving the learning effectiveness of particles. To compensate for this shortcoming, researchers have attempted to develop dynamic topologies to select guiding exemplars for particles. Along this line, Liang and Suganthan [15] designed a random topology, which connects each particle with several randomly selected particles. Their experimental results demonstrated that the randomly constructed topological structure exhibits the best performance when its size is equal to three. In [22], each particle interacts with k particles randomly selected from the swarm. As for the setting of k , it is set between one and the population size. In particular, it can be the same for all particles and can also be different for different particles. In [16], the authors proposed adaptive adjustment of the size of the topology based on the evolution state. Specifically, in the early stage, a small size is maintained to preserve high search diversity, so that particles focus on exploring the solution space. Whereas, at the late stage, a large topology size is maintained to guarantee the convergence, so that particles focus on exploiting the solution space. In [23], the authors combined the global topology and the local topology to select guiding exemplars for particles, so that a good balance between exploration and exploitation could be maintained.

The aforementioned topology-based learning strategies mainly select guiding exemplars for particles based on existing personal best positions found by all particles. To further promote the learning effectiveness of particles, some researchers have attempted to

construct novel exemplars, which might not be visited by particles, for particles, leading to exemplar construction-based learning strategies [14,17,19,24,37,38].

Different from topology-based learning strategies, exemplar construction-based learning strategies construct new exemplars that are not visited by particles based on the personal best positions of all particles. In this direction, the most representative algorithm is the comprehensive learning PSO (CLPSO) [17]. Specifically, this algorithm uses the binary tournament selection mechanism to select a learning object for each dimension of the current particle. From a macro perspective, CLPSO constructs a new position for each particle dimension by dimension that does not exist in the current population. Since CLPSO randomly recombines dimensions of different personal best positions, it ignores the correlation between variables and thus cannot effectively integrate useful evolutionary information together. To further improve the optimization performance of CLPSO, a heterogeneous CLPSO (HCLPSO) was proposed in [25]. This algorithm divides the population into two sub-populations, with one sub-population used for exploration, which is updated by the original CL strategy, and another sub-population used for exploitation, which is guided by the global best position.

Although the above introduced CLPSO variants have achieved good performance in solving multimodal problems, the construction of promising exemplars for particle is inefficient since the recombination of dimensions is totally random. To improve the exemplar construction effectiveness and efficiency, in [37], Zhan et al. proposed an orthogonal learning PSO (OLPSO) by using the orthogonal experimental design to seek useful dimension combinations of the historical positions found by particles. Specifically, OLPSO adopts an orthogonal matrix to evaluate the effectiveness of the dimension combinations and then combines the most useful dimension combinations to construct promising exemplars. Though the exemplar construction efficiency is improved, it consumes many fitness evaluations in the exemplar construction stage. To reduce fitness evaluation consumption in the exemplar construction, in [18], Gong et al. employed the genetic operators such as crossover, mutation, and selection, to construct guiding exemplars for particles, leading to a genetic learning PSO (GLPSO). With these operators, GLPSO is expected to generate diversified and high-quality exemplars for particles. To further improve its optimization effectiveness, in [56], a global GLPSO with a ring topology (GGL-PSOD) was devised, where the ring topology is adopted to generate diversified exemplars based on neighbor particles. Though the constructed exemplars are promising, GLPSO and its variants still consume many fitness evaluations in the exemplar construction. To further construct diversified but promising guiding exemplars for particles, in [19], terminal crossover and steering-based PSO with distribution (TCSPSO) was proposed by devising a new crossover mechanism to construct exemplars. Meanwhile, a global disturbance was designed to improve the population diversity to escape from local areas. In [24], a modified particle swarm optimization with effective guides (MPSOEG) was devised by generating two types of guiding exemplars with an optimal guide creation (OGC) module. In particular, a global exemplar is constructed by the OGC module to guide the swarm towards promising regions, whereas a local exemplar is constructed for each particle to escape from local areas.

Except for the abovementioned learning strategies, some researchers have even attempted utilizing multiple learning strategies to direct the evolution of the swarm in PSO. For instance, in [20], the concept of evolutionary game theory was introduced, and four classical learning strategies were taken as game strategies in the game theory. Then, the swarm adaptively selects the most suitable learning strategy based on the current evolution state. In [28], a dynamic-neighborhood-based switching PSO (DNSPSO) algorithm was proposed by adjusting the personal best position and the global best position based on a distance-based dynamic neighborhood and hybridizing the differential evolution algorithm to alleviate premature convergence.

Although many of the original limitations and shortcomings of PSO have been greatly improved since its introduction, its optimization performance in solving complex optimization problems with many interacting variables and a wide saddle-point region still

encounters great challenges. Therefore, methods to improve the optimization performance of PSO in solving widely emerging complicated problems remains an open issue and deserves careful attention, which results in the research of PSO remaining a highly popular and frontier topic in the evolutionary computation community. To improve the learning effectiveness of particles in complicated environments, this paper proposes a stochastic triad topology-based PSO (STTPSO), which will be elucidated in the following section.

3. Stochastic Triad Topology-Based Particle Swarm Optimization

The most crucial aspect of PSO is the interaction among particles to select guiding exemplars to direct the update of particles [22,26,27,29]. Most existing topology-based PSO variants [21,22,55] only adopt the topologies to select one exemplar to replace the social exemplar (namely *gbest*) in the classical PSO (shown in Equation (1)). Such utilization of topologies is limited since it only changes one exemplar in Equation (1), which results in limited improvement in the learning effectiveness and learning diversity of particles. To alleviate this predicament, this paper proposes a stochastic triad topology-based PSO (STTPSO), which utilizes a stochastic triad topology for each particle to select two novel guiding exemplars to replace the two ones in the classical PSO to promote the learning effectiveness and learning diversity of particles.

3.1. Stochastic Triad Topology

During the evolution, given that *PS* particles are maintained in the swarm, then for each particle x_i ($0 \leq i \leq PS$), a stochastic triad topology is employed to connect the personal best position (*pbest_i*) of this particle and two different personal best positions, which are randomly selected from those of other particles. Given that the two randomly selected personal best positions are *pbest_{r1}* and *pbest_{r2}*, respectively, this paper utilizes the best one among the triad *pbests*, (*pbest_i*, *pbest_{r1}*, and *pbest_{r2}*) and the mean position of these triad *pbests* as the two guiding exemplars to replace the two ones in Equation (1) to update each particle. Specifically, the velocity of each particle is updated as follows:

$$v_i = wv_i + c_1r_1(tpbest_i - x_i) + c_2r_2(\overline{tmean}_i - x_i) \quad (4)$$

where *tpbest_i* is the best one among the triad *pbests*, which is determined as follows:

$$tpbest_i = \operatorname{argmin}\{f(pbest_{r_1}), f(pbest_{r_2}), f(pbest_i)\} \quad (5)$$

\overline{tmean}_i represents the mean position of the triad *pbests*, which is calculated as follows:

$$\overline{tmean}_i = \frac{1}{3}(pbest_{r_1} + pbest_{r_2} + pbest_i) \quad (6)$$

Upon deep observation of Equation (4), we discover the following findings: (1) The first guiding exemplar (*tpbest_i*) is likely different for different particles. This is because the triad topology of each particle is constructed by randomly selecting two different personal best positions (*pbest_{r1}* and *pbest_{r2}*) from those of other particles along with the personal best position (*pbest_i*) of this particle. Due to such randomness, the diversity of the first exemplar could be largely promoted. (2) The second guiding exemplar (\overline{tmean}_i) is also likely different due to the random construction of the triad topology. Therefore, the diversity of the second exemplar is also promoted to a large extent. Along with high diversity of the first exemplar, we can see that the learning diversity of particles is high, which is of great benefit for particles to search the solution space dispersedly and thus is helpful for the swarm to escape from local areas. (3) The first exemplar is expectedly better than the personal best position of the particle to be updated. As a result, the learning effectiveness and efficiency of particles is expected to be promoted, which is beneficial for the swarm to rapidly converge to promising areas. (4) As for the second exemplar, it can be considered as a kind of distribution estimation of the triad *pbests*. Utilizing it as one guiding exemplar is also expected to direct the updated particle to promising areas fast. (5) However, compared with the first exemplar, the quality

of the second exemplar is uncertain. In particular, we can consider that the first exemplar is responsible for convergence, while the second is in charge of swarm diversity. Therefore, in Equation (4), a promising balance between fast convergence and high diversity is implicitly maintained in the update of particles.

As for the triad topology structure, to guarantee the learning effectiveness of particles, instead of frequently changing the topology structure, we first keep the structure fixed for each particle. That is to say, the indexes ($pbest_{r1}$ and $pbest_{r2}$) of the randomly selected two personal best positions for each particle are not changed. Then, we observe the change of the personal best position of each particle. For particle x_i ($0 \leq i \leq PS$), if its personal best position $pbest_i$ keeps unchanged for continuous $stop_{max}$ times, this indicates that the learning effectiveness of this particle under the triad structure degrades. In this situation, to improve the learning effectiveness of this particle, we randomly reselect two different personal best positions from those of other particles to rebuild the triad topology structure. In this way, the learning effectiveness and learning diversity of particles can be largely promoted.

In Section 4.3, investigative experiments are conducted to verify the effectiveness of the adaption strategy for the triad topology structure of each particle. Experimental results show that $stop_{max} = 30$ helps STTPSO achieve the best overall performance, and thus in this paper, we set $stop_{max} = 30$ for STTPSO.

Remark

In essence, the proposed stochastic triad topology belongs to a kind of random topology. In the literature, many studies [21,22,52,57] have adopted random topologies to select guiding exemplars for particles to learn from. Compared with these existing studies, this paper distinguishes itself from them in the following ways:

- (1) Unlike existing studies that use the random topologies to determine only one guiding exemplar to replace the social exemplar ($gbest$) in the classical PSO [6,7], the proposed STTPSO utilizes the stochastic triad topology for each particle to select the best one among the triad personal best positions and computes the mean position of these $pbests$ as the two guiding exemplars to direct the update of this particle. In this way, due to the randomness of the triad topology, not only the diversity of the first exemplar is promoted largely, but also the diversity of the second exemplar is promoted to a large extent. Therefore, the learning diversity of particles is improved, which is beneficial for enhancing the chance of escaping from local areas for the swarm.
- (2) Unlike existing studies that change the random topology structure every generation, this paper adaptively changes the triad topology structure based on the evolution state of each particle. In particular, we record stagnation times of each particle (x_i), which is actually the number of continuous generations where the personal best position ($pbest_i$) of the particle remains unchanged. When such a number exceeds a predefined threshold $stop_{max}$, the triad topology structure is reconstructed by randomly reselecting two different personal best positions from those of other particles. In this way, the triad topology structure of each particle is changed asynchronously, which guarantees the learning effectiveness of particles.

3.2. Dynamic Acceleration Coefficients

As for the parameters in Equation (4), with respect to the inertia weight w , we directly adopt the widely used dynamic strategy as shown in Equation (3) to dynamically adjust w during the evolution.

As for the acceleration coefficients c_1 and c_2 , in the classical PSO, a large body of research has recommended to set them as $c_1 = c_2 = 1.49618$ [18]. Such a setting makes all particles share the same setting, which, as far as we are concerned, is not beneficial for improving the learning diversity of particles. Therefore, to alleviate this issue and to further

enhance the learning diversity of particles, we first randomly samples two values v_1 and v_2 from the Gaussian distribution $Gaussian(1.49618, 0.1)$ as follows:

$$\begin{cases} v_1 = Gaussian(1.49618, 0.1) \\ v_2 = Gaussian(1.49618, 0.1) \end{cases} \quad (7)$$

Then, we set c_1 and c_2 based on the sampled two values as follows:

$$\begin{cases} c_1 = \max(v_1, v_2) \\ c_2 = \min(v_1, v_2) \end{cases} \quad (8)$$

First, the Gaussian distribution $Gaussian(1.49618, 0.1)$ with the mean value set as the classical setting of c_1 and c_2 and the standard deviation set as a small value allows the sampled two values to be close to the classical setting but with a small difference. In this way, the diversity of the settings of c_1 and c_2 is slightly increased, resulting in a slight improvement in the learning diversity of particles.

Second, between the two sampled values, the larger one is set to c_1 , while the smaller one is set to c_2 . This is because, as aforementioned in Equation (4), the first guiding exemplar (the best one among the triad *pbests*) is expectedly better than the second guiding exemplar (the mean position of the triad *pbests*), and thus we can consider that the first exemplar is responsible for the convergence, while the second exemplar takes charge of the diversity. Since the second exemplar is the mean position of the triad *pbests*, which is expectedly different from the first exemplar, we set c_1 with the larger sampled value and c_2 with the smaller value to guarantee that the updated particle learns more from the first exemplar, so that it can approach promising areas fast without serious loss of diversity by learning slightly less from the second exemplar.

Lastly, experiments conducted in Section 4.3 will demonstrate the effectiveness of the proposed dynamic acceleration coefficient strategy.

3.3. Historical Information Utilization

In PSO, the obsolete historical information may also contain useful evolutionary information. As a consequence, many studies [2,58] have maintained an additional archive to store historical information to evolve the swarm. Inspired from this, this paper also maintains an archive of size $PS/2$ to store the obsolete personal best positions of particles.

Specifically, during the evolution, once a particle discovers a better position, its old personal best position is first inserted into the archive and then is replaced by the new better position. Once the archive is full, namely when its size exceeds $PS/2$, an obsolete personal best position is inserted into the archive by randomly replacing a solution in the archive.

During the update of particles, the archive along with the personal best positions of all particles are used to construct the triad topology structure of each particle. In particular, when the stagnation times of the personal best position of one particle exceeds $stop_{max}$, two different personal best positions are randomly selected from the archive and those of the other particles to rebuild the triad topology structure of this particle. In this way, the historical evolutionary information is employed to evolve the swarm.

Due to the utilization of historical information, the number of candidates used to build the triad topology of each particle is increased and thus the learning diversity of particles can be improved largely, which is beneficial for the swarm to escape from local areas. Experiments conducted in Section 4.3 will demonstrate the effectiveness of this additional archive.

3.4. Random Restart Strategy

To further enhance the chance of escaping from local areas for the swarm, this paper further proposes a random restart strategy to introduce initial solutions. Specifically, given a small restart probability pm , during the evolution, in each generation, when a uniformly

sampled value within $[0, 1]$ is smaller than pm , a feasible solution is randomly initialized within the range of variables. Then, instead of inserting this solution into the current swarm as noted in existing studies [59–63], this paper inserts this initialized solution into the archive. If the archive is full, it randomly replaces a solution in the archive.

In particular, such a restart strategy with the initialized solution inserted in the archive could not seriously break the convergence of the swarm but could improve the learning diversity of particles effectively. Most existing studies [60–63] only replace one particle in the current swarm with the initialized solution. Such a strategy usually leads to a very small improvement in the learning diversity and learning effectiveness of particles. This is because the personal best positions of all particles remain unchanged, leading to the learning effectiveness of most particles not improving. However, in our strategy, the randomly initialized solution is inserted into the archive, which is then used to build the triad topology structure of each particle. Therefore, we can see that once the initialized solution is selected to build the triad structure of one particle, at least the second exemplar (namely the mean position of the triad positions in the topology) is changed. Consequently, the learning diversity of particles can be effectively improved, which is beneficial for the swarm to escape from local areas. Experiments conducted in Section 4.3 will demonstrate the effectiveness of this restart strategy.

3.5. Overall Procedure

Integrating the above components together, the overall procedure of the developed STTPSO is shown in Algorithm 1. Specifically, as shown in Lines 1 to 4, the triad topology is constructed for each of PS particles after they are randomly initialized and evaluated. Moreover, the stagnation time of each particle is initialized as 0. Then, the algorithm goes into the main iteration (Lines 5–27). First, for each particle x_i , the inertia weight w is computed (Line 7) and then the acceleration coefficients c_1 and c_2 are set based on Gaussian distribution (Lines 8–11). Subsequently, the particle is updated, and then the personal best position ($pbest_i$) of this particle is updated with its stagnation time $stop_i$ updated as well (Lines 12–19). Once the stagnation time of particle x_i reaches the allowed maximum stagnation time $stop_{max}$, two different personal best positions are randomly selected from those of all particles and the archive to rebuild the triad topology structure (Lines 20–22). After all particles are updated, the random restart strategy is conditionally triggered to randomly insert an initialized solution into the archive (Lines 24–26). The above main iteration proceeds until the maximum number of fitness evaluations is exhausted and at the end of the program, the global best position is obtained as the final output.

From Algorithm 1, it can be observed that during each iteration, $O(PS)$ is needed to compute the parameters such as w , c_1 and c_2 . Following this, $O(PS)$ is needed to obtain the best one among the triad $pbests$ and $O(PS \times D)$ to calculate the mean position of the triad $pbests$ for all particles. Then, $O(PS \times D)$ is used to update particles. During the update of the archive, $O(PS \times D)$ is needed in each generation. Overall, the time complexity of STTPSO is $O(PS \times D)$, which is the same as the classical PSO. Therefore, we can see that STTPSO remains as efficient as the classical PSO.

Algorithm 1: The pseudocode of STTPSO

Input: swarm size PS , maximum fitness evaluations $FEmax$, maximum stagnation times $stop_{max}$, restart probability pm ;

- 1: Initialize PS particles randomly and calculate their fitness;
- 2: Set $fes = PS$, and set the archive empty;
- 3: Randomly select two different personal best positions ($pbest_{r1}$ and $pbest_{r2}$) from the personal best positions of other particles and the archive for each particle to form the associated triad topology;
- 4: Set the stagnation time $stop_i = 0$ ($1 \leq i \leq PS$) for each particle;
- 5: **While** ($fes \leq FEmax$) **do**
- 6: **For** $i = 1:PS$ **do**
- 7: Compute w according to Equation (3);
- 8: Randomly sample c_1 and c_2 from Gaussian(1.49618,0.1);
- 9: **If** $c_1 < c_2$ **then**
- 10: Swap c_1 and c_2 ;
- 11: **End If**
- 12: Update x_i and v_i according to Equations (2) and (4);
- 13: Calculate the fitness of the updated x_i : $f(x_i)$ and $fes += 1$;
- 14: **If** $f(x_i) < f(pbest_i)$ **then**
- 15: Put $pbest_i$ in the archive and set $stop_i = 0$;
- 16: $pbest_i = x_i$;
- 17: **Else**
- 18: $stop_i += 1$;
- 19: **End If**
- 20: **If** $stop_i \geq stop_{max}$ **then**
- 21: Reselect two different personal best positions ($pbest_{r1}$ and $pbest_{r2}$) from those of other particles and the archive for x_i to form the associated triad topology;
- 22: **End If**
- 23: **End For**
- 24: **If** $\text{rand}(0, 1) < pm$ **then**
- 25: Randomly initialize a solution and store it into the archive;
- 26: **End If**
- 27: **End While**
- 28: Obtain the global best solution $gbest$ and its fitness $f(gbest)$;

Output: $f(gbest)$ and $gbest$

4. Experiments

This section mainly verifies the effectiveness of the proposed STTPSO by extensive experiments conducted on the widely used CEC 2017 benchmark function set [44]. Specifically, this benchmark set contains 29 optimization problems with four categories, namely unimodal functions, simple multimodal functions, hybrid functions, and composition functions. Compared with the former two categories of optimization problems, the latter two kinds of optimization problems are more difficult to optimize.

4.1. Experimental Setup

Firstly, in order to verify the effectiveness of STTPSO effectively, we select seven most advanced PSO variants as the compared methods, namely DNSPSO [28], XPSO [23], DPLPSO [45], TCSPSO [19], GLPSO [18], HCLPSO [25] and CLPSO [17]. DNSPSO, XPSO and DPLPSO are state-of-the-art topology-based PSO variants, while TCSPSO, GLPSO, HCLPSO and CLPSO are state-of-the-art exemplar construction based PSO variants.

Secondly, in order to verify the optimization performance of STTPSO in a comprehensive way, we conduct comparative experiments on the CEC 2017 benchmark set with three dimension sizes, namely 30- D , 50- D and 100- D respectively. For the sake of fairness, the maximum number of function evaluation times is set as $10,000 \times D$ for all algorithms.

Thirdly, for fair comparisons, except for the population size, we adopt the parameter settings for all key parameters in the compared PSO variants as recommended in the associated papers. As for the population size, due to its sensitivity to optimization problems, we fine-tune its settings for all compared PSO variants. After preliminary parameter fine-tuning experiments, Table 2 lists the specific parameter settings of all algorithms.

Table 2. Parameter settings of the proposed STTPSO and the compared algorithms.

Algorithm	<i>D</i>	Parameter Settings	
STTPSO	30	<i>PS</i> = 300	<i>AS</i> = <i>PS</i> /2; <i>w</i> = 0.9~0.4; <i>c</i> ~N(1.49618,0.1); <i>pm</i> = 0.01; <i>stop_{max}</i> = 30
	50	<i>PS</i> = 300	
	100	<i>PS</i> = 300	
DNSPSO	30	<i>PS</i> = 50	<i>w</i> = 0.4~0.9; <i>k</i> = 5; <i>F</i> = 0.5; <i>CR</i> = 0.9;
	50	<i>PS</i> = 50	
	100	<i>PS</i> = 60	
XPSO	30	<i>PS</i> = 100	<i>η</i> = 0.2; <i>Stag_{max}</i> = 5; <i>p</i> = 0.5; <i>σ</i> = 0.1
	50	<i>PS</i> = 150	
	100	<i>PS</i> = 150	
TCPSO	30	<i>PS</i> = 50	<i>w</i> = 0.9~0.4; <i>c</i> ₁ = <i>c</i> ₂ = 2
	50	<i>PS</i> = 50	
	100	<i>PS</i> = 50	
GLPSO	30	<i>PS</i> = 40	<i>w</i> = 0.7298; <i>c</i> = 1.49618; <i>pm</i> = 0.1; <i>sg</i> = 7
	50	<i>PS</i> = 40	
	100	<i>PS</i> = 50	
HCLPSO	30	<i>PS</i> = 160	<i>w</i> = 0.99~0.2; <i>c</i> ₁ = 2.5~0.5; <i>c</i> ₂ = 0.5~2.5; <i>c</i> = 3~1.5
	50	<i>PS</i> = 180	
	100	<i>PS</i> = 180	
DPLPSO	30	<i>PS</i> = 40	<i>c</i> ₁ = <i>c</i> ₂ = 2; <i>L</i> = 50
	50	<i>PS</i> = 40	
	100	<i>PS</i> = 40	
CLPSO	30	<i>PS</i> = 40	<i>Pc</i> = 0.05~0.5
	50	<i>PS</i> = 40	
	100	<i>PS</i> = 40	

Finally, in order to comprehensively evaluate the optimization performance of all algorithms, we independently execute each algorithm 30 times and use the median, the mean and the standard deviation to measure the optimization performance of each algorithm. To distinguish the statistical significance with respect to the performance difference between two algorithms, the Wilcoxon rank sum test at the significance level of 0.05 is conducted. Furthermore, to obtain the overall performance of each algorithm on the whole benchmark set, the Friedman test is conducted to obtain the overall ranks of all algorithms on the whole benchmark set.

4.2. Comparison with State-of-the-Art PSO Variants

In this section, we conduct extensive comparative experiments on the CEC 2017 benchmark set with the three dimension sizes to compare STTPSO with the seven state-of-art PSO variants. Tables 3–5, respectively, show the comparison results between the proposed STTPSO and the seven PSO variants on the 30-*D*, the 50-*D* and the 100-*D* CEC 2017

benchmark functions. In these tables, the symbols '+', '-' and '=' indicate that the proposed STTPSO is significantly superior to, significantly inferior to and roughly equivalent to the associated compared algorithm on the associated functions. As shown in the second to last row of each table, 'w/t/l' count the number of functions where the proposed STTPSO achieves significantly better performance, obtains roughly equivalent performance, and exhibits significantly worse performance than the compared algorithms, respectively. In fact, they are the numbers of '+', '=' and '-'. In the last row of each table, the average rank of each algorithm obtained by the Friedman test is displayed. Moreover, the statistical comparison results between the proposed STTPSO and the seven state-of-the-art PSO variants on the CEC 2017 benchmark set with different dimension sizes in terms of "w/t/l" are summarized in Table 6.

As shown in Table 3, the comparison results on the 30-D CEC 2017 benchmark functions are summarized below:

- (1) According to the Friedman test results as shown in the last row, STTPSO achieves the lowest rank among all eight algorithms and its rank value (1.86) is much smaller than those (at least 2.55) of the seven compared algorithms. This demonstrates that STTPSO achieves the best overall performance on the 30-D CEC 2017 benchmark functions, and presents significant superiority over the seven compared algorithms.
- (2) The second last row of Table 2 shows that STTPSO is significantly superior to the compared algorithms on at least 21 problems except for XPSO, and only presents inferior performance on, at most, five problems. Compared with XPSO, STTPSO obtains significantly better performance on 18 problems, while only performing worse than XPSO on three problems.
- (3) In terms of the comparison results on different types of optimization problems, STTPSO achieves highly competitive performance with all the compared algorithms on the two unimodal problems. In particular, it shows significant dominance to DNSPSO and DPLPSO both on the two problems. In terms of the six simple multimodal problems, except for DNSPSO, STTPSO shows significantly better performance than the other six compared algorithms on all these problems. Compared with DNSPSO, STTPSO presents significant superiority on five problems and shows inferiority on only one problem. Regarding the 10 hybrid problems, STTPSO shows much better performance than DPLPSO on all 10 problems. Compared with DNSPSO, TCPSO, and HCLPSO, STTPSO obtains significantly better performance on seven, six, and seven problems, respectively, and only shows inferiority to them on, at most, two problems. In comparison with XPSO, GLPSO, and CLPSO, STTPSO achieves no worse performance on at least seven problems and displays inferiority to them on, at most, three problems. Concerning the 11 composition problems, STTPSO outperforms the seven compared algorithms on at least nine problems, and only shows inferiority on, at most, two problems. In particular, STTPSO significantly outperforms both TCPSO and GLPSO on all these problems and obtains much better performance than both HCLPSO and DPLPSO on 10 problems with no inferiority to them on all the 11 problems. Overall, it is demonstrated that STTPSO shows promise in solving various kinds of problems and particularly obtains good performance on complicated problems, such as multimodal problems, hybrid problems, and composition problems.

Table 3. Comparison results between the proposed STTPSO and the 7 state-of-the-art and popular PSO variants on the 30-D CEC 2017 benchmark functions. The highlighted *p*-values means that the proposed STTPSO is significantly better than the associated compared algorithms on the corresponding problems.

<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCSPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₁	Unimodal Functions	Median	1.19×10^3	1.95×10^5	2.26×10^3	3.20×10^3	2.30×10^3	5.49×10^3	2.64×10^9	1.52×10^2
		Mean	2.10×10^3	2.11×10^5	4.05×10^3	3.66×10^3	3.06×10^3	8.65×10^3	2.87×10^9	3.88×10^2
		Std	2.28×10^3	1.30×10^5	4.72×10^3	4.08×10^3	2.42×10^3	7.34×10^3	1.11×10^9	7.31×10^2
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$2.17 \times 10^{-1=}$	$1.95 \times 10^{-1=}$	$5.85 \times 10^{-2=}$	$8.55 \times 10^{-5+}$	$1.83 \times 10^{-6+}$	$7.84 \times 10^{-5-}$
<i>f</i> ₃		Median	1.52×10^4	1.51×10^5	6.26×10^{-2}	9.94×10^3	1.14×10^{-13}	4.54×10^1	3.91×10^4	4.30×10^4
		Mean	1.53×10^4	1.54×10^5	7.91×10^{-1}	1.15×10^4	1.33×10^{-13}	6.87×10^1	3.87×10^4	4.41×10^4
		Std	4.39×10^3	3.37×10^4	2.00×10^0	3.68×10^3	5.36×10^{-14}	8.17×10^1	8.36×10^3	1.00×10^4
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6-}$	$2.51 \times 10^{-4-}$	$1.83 \times 10^{-6-}$	$1.83 \times 10^{-6-}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₁₋₃	<i>w/t/l</i>	-	2/0/0	0/1/1	0/1/1	0/1/1	1/0/1	2/0/0	1/0/1	
<i>f</i> ₄		Median	8.47×10^1	2.54×10^1	1.24×10^2	1.30×10^2	1.47×10^2	8.56×10^1	7.62×10^2	9.09×10^1
		Mean	8.48×10^1	2.56×10^1	1.20×10^2	1.32×10^2	1.48×10^2	8.73×10^1	7.99×10^2	9.12×10^1
		Std	3.74×10^{-1}	1.07×10^0	2.71×10^1	4.84×10^1	4.27×10^1	7.92×10^0	1.86×10^2	1.57×10^0
		<i>p</i> -value	-	$1.83 \times 10^{-6-}$	$3.56 \times 10^{-5+}$	$3.89 \times 10^{-5+}$	$4.97 \times 10^{-6+}$	$7.84 \times 10^{-5+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$
<i>f</i> ₅	Simple Multimodal Functions	Median	4.98×10^0	2.04×10^2	4.18×10^1	8.56×10^1	6.77×10^1	6.45×10^1	2.00×10^2	7.66×10^1
		Mean	4.71×10^0	2.03×10^2	4.38×10^1	8.92×10^1	6.68×10^1	6.77×10^1	1.93×10^2	7.52×10^1
		Std	1.96×10^0	1.25×10^1	1.58×10^1	2.54×10^1	1.96×10^1	1.67×10^1	3.28×10^1	6.69×10^0
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₆		Median	1.14×10^{-13}	1.87×10^{-1}	3.82×10^{-3}	8.01×10^{-1}	6.34×10^{-3}	1.62×10^{-4}	2.97×10^1	2.66×10^{-6}
		Mean	1.12×10^{-7}	1.91×10^{-1}	1.50×10^{-2}	1.04×10^0	9.69×10^{-3}	2.41×10^{-3}	2.98×10^1	2.86×10^{-6}
		Std	2.91×10^{-7}	5.92×10^{-2}	3.82×10^{-2}	1.15×10^0	8.77×10^{-3}	5.85×10^{-3}	5.82×10^0	1.90×10^{-6}
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$

Table 3. Cont.

<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO	
<i>f</i> ₇	Median		3.44×10^1	2.36×10^2	7.94×10^1	1.45×10^2	9.75×10^1	1.06×10^2	2.90×10^2	9.23×10^1	
	Mean		3.46×10^1	2.33×10^2	8.17×10^1	1.42×10^2	9.86×10^1	1.01×10^2	2.88×10^2	9.05×10^1	
	Std		1.12×10^0	1.69×10^1	1.81×10^1	2.87×10^1	1.53×10^1	1.86×10^1	2.45×10^1	7.88×10^0	
	<i>p</i> -value		-	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₈	Median		3.98×10^0	2.01×10^2	3.83×10^1	9.55×10^1	5.97×10^1	5.66×10^1	1.94×10^2	8.09×10^1	
	Mean		4.15×10^0	2.02×10^2	3.98×10^1	9.33×10^1	6.08×10^1	6.35×10^1	1.90×10^2	8.18×10^1	
	Std		1.67×10^0	1.03×10^1	1.37×10^1	2.22×10^1	1.65×10^1	2.08×10^1	3.20×10^1	9.86×10^0	
	<i>p</i> -value		-	$1.82 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	
<i>f</i> ₉	Median		5.69×10^{-14}	1.50×10^0	1.45×10^0	3.01×10^2	5.98×10^1	4.90×10^1	1.27×10^3	6.58×10^2	
	Mean		5.69×10^{-14}	2.09×10^0	2.73×10^0	3.85×10^2	7.13×10^1	8.90×10^1	1.50×10^3	6.76×10^2	
	Std		5.69×10^{-14}	1.39×10^0	3.44×10^0	3.36×10^2	4.83×10^1	1.49×10^2	6.80×10^2	2.80×10^2	
	<i>p</i> -value		-	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	
<i>f</i> ₄₋₉	<i>w/t/l</i>		-	5/0/1	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	
<i>f</i> ₁₀	Median		1.81×10^3	6.21×10^3	2.80×10^3	2.98×10^3	3.26×10^3	2.87×10^3	6.39×10^3	3.00×10^3	
	Mean		2.82×10^3	5.90×10^3	2.62×10^3	2.97×10^3	3.23×10^3	2.90×10^3	6.33×10^3	2.94×10^3	
	Std		1.82×10^3	1.01×10^3	6.13×10^2	4.22×10^2	8.64×10^2	5.17×10^2	4.47×10^2	2.77×10^2	
	<i>p</i> -value		-	$4.50 \times 10^{-6+}$	$5.37 \times 10^{-1=}$	$7.89 \times 10^{-1=}$	$2.41 \times 10^{-1=}$	$8.69 \times 10^{-1=}$	$1.83 \times 10^{-6+}$	$8.53 \times 10^{-1=}$	
<i>f</i> ₁₁	Median	Hybrid Functions		1.79×10^1	9.12×10^1	8.01×10^1	1.16×10^2	7.24×10^1	1.09×10^2	4.10×10^2	1.21×10^2
	Mean			2.79×10^1	9.19×10^1	8.63×10^1	1.18×10^2	7.82×10^1	1.08×10^2	4.23×10^2	1.16×10^2
	Std			2.33×10^1	8.54×10^0	4.60×10^1	4.27×10^1	3.83×10^1	4.49×10^1	1.19×10^2	1.72×10^1
	<i>p</i> -value			-	$1.83 \times 10^{-6+}$	$9.77 \times 10^{-6+}$	$4.50 \times 10^{-6+}$	$1.68 \times 10^{-5+}$	$3.03 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$2.02 \times 10^{-6+}$
<i>f</i> ₁₂	Median		5.38×10^4	5.52×10^7	2.64×10^4	1.86×10^5	1.13×10^6	2.39×10^5	1.55×10^8	1.74×10^6	
	Mean		6.09×10^4	6.25×10^7	1.93×10^5	5.12×10^5	3.35×10^6	2.55×10^5	1.77×10^8	2.01×10^6	
	Std		3.97×10^4	2.90×10^7	5.57×10^5	6.68×10^5	4.39×10^6	1.70×10^5	9.46×10^7	1.11×10^6	
	<i>p</i> -value		-	$1.83 \times 10^{-6+}$	$2.17 \times 10^{-1=}$	$1.36 \times 10^{-5+}$	$1.72 \times 10^{-5+}$	$8.88 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	

Table 3. Cont.

<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₁₃	Median		5.18×10^3	1.32×10^6	7.54×10^3	8.24×10^3	7.23×10^3	6.10×10^4	6.71×10^6	3.39×10^3
	Mean		1.10×10^4	1.52×10^6	9.83×10^3	3.28×10^5	1.19×10^4	3.80×10^4	2.47×10^7	3.40×10^3
	Std		1.11×10^4	7.16×10^5	1.02×10^4	1.14×10^6	1.45×10^4	2.64×10^4	7.42×10^7	1.42×10^3
	<i>p</i> -value		-	$1.83 \times 10^{-6}+$	$9.34 \times 10^{-1}=\$	$4.59 \times 10^{-1}=\$	$1.00 \times 10^0=\$	$2.26 \times 10^{-5}+$	$1.83 \times 10^{-6}+$	$1.14 \times 10^{-2}-$
<i>f</i> ₁₄	Median		3.89×10^3	1.90×10^2	3.75×10^3	3.49×10^4	1.66×10^3	1.43×10^4	1.20×10^5	4.42×10^4
	Mean		6.63×10^3	1.93×10^2	5.36×10^3	5.20×10^4	3.38×10^4	1.55×10^4	1.66×10^5	4.93×10^4
	Std		6.41×10^3	2.22×10^1	4.38×10^3	7.90×10^4	7.81×10^4	9.99×10^3	2.35×10^5	3.41×10^4
	<i>p</i> -value		-	$1.83 \times 10^{-6}-$	$2.49 \times 10^{-1}=\$	$5.93 \times 10^{-4}+$	$9.51 \times 10^{-1}=\$	$2.51 \times 10^{-4}+$	$4.08 \times 10^{-6}+$	$4.50 \times 10^{-6}+$
<i>f</i> ₁₅	Median		3.91×10^3	4.24×10^4	1.61×10^3	1.08×10^4	5.46×10^3	8.96×10^3	1.28×10^4	4.08×10^2
	Mean		7.84×10^3	4.70×10^4	3.59×10^3	1.33×10^4	8.80×10^3	1.35×10^4	2.61×10^4	4.59×10^2
	Std		8.32×10^3	2.04×10^4	4.74×10^3	1.04×10^4	8.48×10^3	1.22×10^4	3.16×10^4	2.45×10^2
	<i>p</i> -value		-	$1.83 \times 10^{-6}+$	$8.04 \times 10^{-2}=\$	$3.41 \times 10^{-2}+$	$7.89 \times 10^{-1}=\$	$4.38 \times 10^{-2}+$	$2.33 \times 10^{-3}+$	$3.34 \times 10^{-6}-$
<i>f</i> ₁₆	Median		2.22×10^1	1.88×10^3	5.70×10^2	8.65×10^2	8.49×10^2	7.46×10^2	1.57×10^3	6.61×10^2
	Mean		5.93×10^1	1.87×10^3	5.33×10^2	8.54×10^2	8.21×10^2	7.11×10^2	1.52×10^3	6.24×10^2
	Std		6.79×10^1	1.61×10^2	1.96×10^2	2.56×10^2	2.22×10^2	2.08×10^2	2.55×10^2	1.64×10^2
	<i>p</i> -value		-	$1.83 \times 10^{-6}+$	$1.83 \times 10^{-6}+$	$1.83 \times 10^{-6}+$	$1.83 \times 10^{-6}+$	$1.83 \times 10^{-6}+$	$1.83 \times 10^{-6}+$	$1.83 \times 10^{-6}+$
<i>f</i> ₁₇	Median		4.48×10^1	8.55×10^2	1.64×10^2	3.18×10^2	2.02×10^2	3.12×10^2	4.36×10^2	1.96×10^2
	Mean		4.69×10^1	8.68×10^2	1.45×10^2	2.96×10^2	2.28×10^2	3.22×10^2	4.31×10^2	1.88×10^2
	Std		1.01×10^1	1.21×10^2	8.08×10^1	1.44×10^2	1.38×10^2	1.54×10^2	1.50×10^2	6.59×10^1
	<i>p</i> -value		-	$1.83 \times 10^{-6}+$	$4.97 \times 10^{-6}+$	$1.83 \times 10^{-6}+$	$3.34 \times 10^{-6}+$	$1.83 \times 10^{-6}+$	$1.83 \times 10^{-6}+$	$1.83 \times 10^{-6}+$
<i>f</i> ₁₈	Median		1.87×10^5	1.88×10^5	9.24×10^4	1.41×10^5	1.68×10^4	1.90×10^5	8.67×10^5	1.87×10^5
	Mean		2.64×10^5	2.16×10^5	1.46×10^5	2.78×10^5	1.23×10^5	2.05×10^5	1.03×10^6	2.46×10^5
	Std		2.51×10^5	8.08×10^4	1.27×10^5	2.93×10^5	4.87×10^5	1.47×10^5	8.56×10^5	1.55×10^5
	<i>p</i> -value		-	$1.00 \times 10^0=\$	$5.58 \times 10^{-2}=\$	$9.18 \times 10^{-1}=\$	$1.20 \times 10^{-4}-$	$9.34 \times 10^{-1}=\$	$6.60 \times 10^{-5}+$	$9.18 \times 10^{-1}=\$

Table 3. Cont.

<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₁₉		Median	5.77×10^3	1.95×10^3	3.65×10^3	7.66×10^3	2.99×10^3	1.31×10^4	1.52×10^4	1.05×10^2
		Mean	1.10×10^4	2.25×10^3	4.56×10^3	1.49×10^4	7.89×10^3	1.63×10^4	3.47×10^4	1.35×10^2
		Std	1.32×10^4	1.04×10^3	4.83×10^3	1.58×10^4	1.05×10^4	1.76×10^4	7.74×10^4	8.36×10^1
		<i>p</i> -value	-	2.86×10^{-3} –	3.78×10^{-2} –	3.24×10^{-1} =	3.34×10^{-1} =	2.94×10^{-1} =	1.80×10^{-2}+	1.83×10^{-6} –
<i>f</i> _{10–19}	<i>w/t/l</i>		-	7/1/2	3/6/1	6/4/0	4/5/1	7/3/0	10/0/0	5/2/3
<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₂₀		Median	3.72×10^1	3.50×10^2	1.74×10^2	3.84×10^2	1.96×10^2	2.13×10^2	3.66×10^2	1.94×10^2
		Mean	4.65×10^1	3.87×10^2	1.84×10^2	3.70×10^2	2.16×10^2	2.09×10^2	4.00×10^2	1.89×10^2
		Std	3.35×10^1	1.20×10^2	6.75×10^1	1.39×10^2	1.01×10^2	1.04×10^2	1.28×10^2	6.57×10^1
		<i>p</i> -value	-	1.83×10^{-6}+	2.48×10^{-6}+	1.83×10^{-6}+	3.69×10^{-6}+	5.48×10^{-6}+	1.83×10^{-6}+	3.69×10^{-6}+
<i>f</i> ₂₁	Composition Functions	Median	2.12×10^2	4.04×10^2	2.35×10^2	2.81×10^2	2.66×10^2	2.75×10^2	4.02×10^2	2.88×10^2
		Mean	2.13×10^2	4.04×10^2	2.38×10^2	2.84×10^2	2.67×10^2	2.76×10^2	4.00×10^2	2.83×10^2
		Std	3.76×10^0	9.54×10^0	1.05×10^1	2.25×10^1	2.07×10^1	1.25×10^1	2.25×10^1	2.53×10^1
		<i>p</i> -value	-	1.83×10^{-6}+	1.83×10^{-6}+	1.83×10^{-6}+	1.82×10^{-6}+	1.82×10^{-6}+	1.82×10^{-6}+	2.47×10^{-6}+
<i>f</i> ₂₂		Median	1.00×10^2	6.38×10^3	1.00×10^2	1.05×10^2	1.00×10^2	1.02×10^2	5.79×10^2	2.06×10^2
		Mean	1.00×10^2	6.29×10^3	4.30×10^2	1.70×10^3	1.02×10^2	8.37×10^2	5.92×10^2	8.25×10^2
		Std	0.00×10^0	7.48×10^2	9.91×10^2	1.75×10^3	3.08×10^0	1.49×10^3	1.44×10^2	1.18×10^3
		<i>p</i> -value	-	1.82×10^{-6}+	3.82×10^{-3}+	1.43×10^{-4}+	1.63×10^{-3}+	4.67×10^{-4}+	1.83×10^{-6}+	1.83×10^{-6}+
<i>f</i> ₂₃		Median	3.85×10^2	5.83×10^2	3.99×10^2	4.44×10^2	4.26×10^2	4.53×10^2	6.77×10^2	4.46×10^2
		Mean	3.86×10^2	5.87×10^2	3.98×10^2	4.47×10^2	4.33×10^2	4.51×10^2	6.84×10^2	4.45×10^2
		Std	7.70×10^0	3.62×10^1	1.15×10^1	2.85×10^1	3.02×10^1	2.02×10^1	4.57×10^1	1.09×10^1
		<i>p</i> -value	-	1.83×10^{-6}+	6.89×10^{-4}+	1.82×10^{-6}+	2.24×10^{-6}+	1.83×10^{-6}+	1.83×10^{-6}+	1.83×10^{-6}+

Table 3. Cont.

<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₂₄	Median		4.60×10^2	6.68×10^2	4.70×10^2	5.37×10^2	4.88×10^2	5.35×10^2	7.32×10^2	5.60×10^2
	Mean		4.61×10^2	6.82×10^2	4.73×10^2	5.38×10^2	4.99×10^2	5.39×10^2	7.37×10^2	5.60×10^2
	Std		8.66×10^0	4.48×10^1	2.40×10^1	5.08×10^1	3.90×10^1	2.34×10^1	3.79×10^1	1.66×10^1
	<i>p</i> -value		-	$1.83 \times 10^{-6}+$	$1.28 \times 10^{-2}+$	$2.24 \times 10^{-6}+$	$2.06 \times 10^{-5}+$	$1.83 \times 10^{-6}+$	$1.82 \times 10^{-6}+$	$1.83 \times 10^{-6}+$
<i>f</i> ₂₅	Median		3.87×10^2	3.79×10^2	3.91×10^2	4.14×10^2	4.09×10^2	3.89×10^2	6.00×10^2	3.89×10^2
	Mean		3.87×10^2	3.78×10^2	3.92×10^2	4.12×10^2	4.04×10^2	3.89×10^2	6.18×10^2	3.89×10^2
	Std		2.06×10^{-1}	1.04×10^0	4.86×10^0	1.56×10^1	1.24×10^1	8.23×10^0	8.24×10^1	5.62×10^{-1}
	<i>p</i> -value		-	$1.77 \times 10^{-6}-$	$6.65 \times 10^{-6}+$	$1.83 \times 10^{-6}+$	$1.82 \times 10^{-6}+$	$1.74 \times 10^{-2}+$	$1.83 \times 10^{-6}+$	$1.78 \times 10^{-6}+$
<i>f</i> ₂₆	Median		1.47×10^3	3.28×10^3	3.00×10^2	2.32×10^3	1.94×10^3	2.04×10^3	1.61×10^3	1.85×10^3
	Mean		1.49×10^3	3.45×10^3	6.97×10^2	2.23×10^3	1.92×10^3	1.82×10^3	1.95×10^3	1.58×10^3
	Std		1.09×10^2	4.89×10^2	6.06×10^2	6.97×10^2	4.77×10^2	6.36×10^2	1.02×10^3	4.70×10^2
	<i>p</i> -value		-	$1.83 \times 10^{-6}+$	$3.56 \times 10^{-5}-$	$1.97 \times 10^{-4}+$	$1.67 \times 10^{-4}+$	$1.52 \times 10^{-2}+$	$1.92 \times 10^{-1}=-$	$1.88 \times 10^{-1}=-$
<i>f</i> ₂₇	Median		5.13×10^2	5.00×10^2	5.36×10^2	5.61×10^2	5.48×10^2	5.14×10^2	8.09×10^2	5.11×10^2
	Mean		5.18×10^2	5.00×10^2	5.35×10^2	5.61×10^2	5.50×10^2	5.16×10^2	8.19×10^2	5.10×10^2
	Std		1.40×10^1	0.00×10^0	1.10×10^1	1.95×10^1	1.28×10^1	1.59×10^1	5.73×10^1	4.56×10^0
	<i>p</i> -value		-	$2.24 \times 10^{-6}-$	$9.31 \times 10^{-5}+$	$2.87 \times 10^{-6}+$	$2.48 \times 10^{-6}+$	$8.05 \times 10^{-1}=-$	$1.82 \times 10^{-6}+$	$1.70 \times 10^{-2}-$
<i>f</i> ₂₈	Median		4.08×10^2	5.00×10^2	4.03×10^2	4.40×10^2	4.72×10^2	4.55×10^2	8.24×10^2	4.74×10^2
	Mean		3.79×10^2	5.00×10^2	3.86×10^2	4.51×10^2	4.51×10^2	4.48×10^2	8.80×10^2	4.83×10^2
	Std		5.79×10^1	0.00×10^0	6.85×10^1	5.21×10^1	7.00×10^1	3.48×10^1	1.61×10^2	2.88×10^1
	<i>p</i> -value		-	$1.81 \times 10^{-6}+$	$7.76 \times 10^{-1}=-$	$3.73 \times 10^{-4}+$	$1.72 \times 10^{-3}+$	$1.54 \times 10^{-4}+$	$1.83 \times 10^{-6}+$	$2.02 \times 10^{-6}+$
<i>f</i> ₂₉	Median		4.84×10^2	1.66×10^3	5.66×10^2	8.62×10^2	7.74×10^2	6.92×10^2	1.34×10^3	6.58×10^2
	Mean		5.11×10^2	1.56×10^3	5.88×10^2	9.05×10^2	8.17×10^2	6.98×10^2	1.34×10^3	6.46×10^2
	Std		7.29×10^1	2.89×10^2	9.03×10^1	1.86×10^2	2.37×10^2	1.56×10^2	2.29×10^2	7.20×10^1
	<i>p</i> -value		-	$1.83 \times 10^{-6}+$	$2.95 \times 10^{-4}+$	$1.83 \times 10^{-6}+$	$1.72 \times 10^{-5}+$	$1.30 \times 10^{-5}+$	$1.83 \times 10^{-6}+$	$7.33 \times 10^{-6}+$

Table 3. Cont.

<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₃₀		Median	4.19×10^3	4.00×10^4	8.04×10^3	1.20×10^4	9.45×10^3	7.39×10^3	1.96×10^6	1.27×10^4
		Mean	5.03×10^3	6.20×10^4	9.78×10^3	1.80×10^4	2.08×10^4	8.59×10^3	2.55×10^6	1.37×10^4
		Std	2.02×10^3	5.12×10^4	6.72×10^3	1.77×10^4	2.96×10^4	4.52×10^3	1.97×10^6	3.99×10^3
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$4.36 \times 10^{-4+}$	$4.08 \times 10^{-6+}$	$1.88 \times 10^{-5+}$	$1.54 \times 10^{-3+}$	$1.83 \times 10^{-6+}$	$2.02 \times 10^{-6+}$
<i>f</i> _{20–30}	<i>w/t/l</i>		-	9/0/2	9/1/1	11/0/0	11/0/0	10/1/0	10/1/0	9/1/1
	<i>w/t/l</i>		-	23/1/5	18/8/3	23/5/1	21/6/2	24/4/1	28/1/0	21/3/5
	<i>rank</i>		1.86	6.00	2.55	5.48	4.00	4.41	7.45	4.24

Table 4. Comparison results between the proposed STTPSO and the 7 state-of-the-art and popular PSO variants on the 50-D CEC 2017 benchmark functions. The highlighted *p*-values means that the proposed STTPSO is significantly better than the associated compared algorithms on the corresponding problems.

<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₁	Unimodal Functions	Median	2.89×10^3	5.13×10^3	8.51×10^2	5.47×10^3	3.11×10^3	9.83×10^3	1.92×10^{10}	2.06×10^3
		Mean	4.33×10^3	8.53×10^3	3.66×10^3	2.84×10^6	2.76×10^6	7.15×10^7	1.84×10^{10}	2.59×10^3
		Std	4.38×10^3	1.13×10^4	5.11×10^3	1.52×10^7	1.37×10^7	2.72×10^8	4.36×10^9	1.88×10^3
		<i>p</i> -value	-	$2.02 \times 10^{-1=}$	$4.97 \times 10^{-1=}$	$2.02 \times 10^{-1=}$	$7.69 \times 10^{-2=}$	$2.51 \times 10^{-4+}$	$1.83 \times 10^{-6+}$	$2.67 \times 10^{-1=}$
<i>f</i> ₃		Median	5.80×10^4	3.84×10^5	4.35×10^3	5.55×10^4	$4.55E^{-13}$	4.71×10^3	1.23×10^5	1.31×10^5
		Mean	5.85×10^4	3.82×10^5	4.72×10^3	5.83×10^4	$3.92E^{-12}$	5.11×10^3	1.22×10^5	1.31×10^5
		Std	8.60×10^3	6.15×10^4	1.64×10^3	9.39×10^3	$1.49E^{-11}$	2.41×10^3	1.62×10^4	2.22×10^4
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6-}$	$5.51 \times 10^{-1=}$	$1.83 \times 10^{-6-}$	$1.83 \times 10^{-6-}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> _{1–3}	<i>w/t/l</i>		-	1/1/0	0/1/1	0/2/0	0/1/1	1/0/1	2/0/0	1/1/0
<i>f</i> ₄	Simple Multimodal Functions	Median	1.75×10^2	4.57×10^1	2.46×10^2	2.88×10^2	3.22×10^2	1.61×10^2	3.81×10^3	1.90×10^2
		Mean	1.69×10^2	5.24×10^1	2.33×10^2	2.93×10^2	3.29×10^2	1.48×10^2	4.02×10^3	1.87×10^2
		Std	3.35×10^1	2.36×10^1	5.15×10^1	9.09×10^1	8.97×10^1	5.23×10^1	1.06×10^3	2.00×10^1
		<i>p</i> -value	-	$2.24 \times 10^{-6-}$	$5.08 \times 10^{-5+}$	$6.65 \times 10^{-6+}$	$2.02 \times 10^{-6+}$	$9.99 \times 10^{-2=}$	$1.83 \times 10^{-6+}$	$1.90 \times 10^{-2+}$
<i>f</i> ₅		Median	8.96×10^0	4.11×10^2	8.21×10^1	1.87×10^2	1.47×10^2	1.65×10^2	4.58×10^2	2.02×10^2
		Mean	9.09×10^0	4.12×10^2	8.33×10^1	1.91×10^2	1.42×10^2	1.66×10^2	4.46×10^2	1.98×10^2
		Std	2.70×10^0	1.88×10^1	1.50×10^1	3.82×10^1	3.83×10^1	3.23×10^1	4.64×10^1	1.55×10^1
		<i>p</i> -value	-	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$

Table 4. Cont.

<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₆		Median	1.22×10^{-6}	1.02×10^{-1}	5.59×10^{-2}	3.01×10^0	1.19×10^{-2}	1.85×10^{-3}	5.28×10^1	1.23×10^{-8}
		Mean	4.48×10^{-6}	1.16×10^{-1}	1.53×10^{-1}	3.93×10^0	2.00×10^{-2}	2.65×10^{-3}	5.18×10^1	2.45×10^{-3}
		Std	8.10×10^{-6}	4.66×10^{-2}	2.59×10^{-1}	3.68×10^0	2.08×10^{-2}	2.37×10^{-3}	4.60×10^0	1.32×10^{-2}
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₇		Median	6.27×10^1	4.70×10^2	1.51×10^2	3.18×10^2	2.27×10^2	1.94×10^2	7.71×10^2	2.11×10^2
		Mean	6.33×10^1	4.70×10^2	1.53×10^2	3.35×10^2	2.36×10^2	2.02×10^2	7.70×10^2	2.10×10^2
		Std	3.22×10^0	1.85×10^1	2.57×10^1	6.19×10^1	3.89×10^1	3.06×10^1	6.79×10^1	1.44×10^1
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₈		Median	8.96×10^0	3.99×10^2	8.81×10^1	1.97×10^2	1.36×10^2	1.56×10^2	4.60×10^2	1.96×10^2
		Mean	9.35×10^0	4.00×10^2	9.29×10^1	2.09×10^2	1.41×10^2	1.56×10^2	4.45×10^2	1.97×10^2
		Std	3.22×10^0	1.86×10^1	2.34×10^1	6.12×10^1	3.13×10^1	2.68×10^1	4.48×10^1	1.61×10^1
		<i>p</i> -value	-	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.82 \times 10^{-6+}$
<i>f</i> ₉		Median	4.99×10^{-1}	1.72×10^1	1.55×10^1	2.85×10^3	4.51×10^2	1.25×10^3	1.08×10^4	3.95×10^3
		Mean	8.56×10^{-1}	3.56×10^1	4.97×10^1	3.34×10^3	5.98×10^2	1.26×10^3	1.10×10^4	4.22×10^3
		Std	1.05×10^0	5.56×10^1	8.96×10^1	1.87×10^3	5.23×10^2	5.35×10^2	1.90×10^3	1.20×10^3
		<i>p</i> -value	-	$3.34 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> _{4–9}	<i>w/t/l</i>	-	5/0/1	6/0/0	6/0/0	6/0/0	6/0/0	5/1/0	6/0/0	6/0/0
<i>f</i> ₁₀		Median	3.60×10^3	1.16×10^4	5.17×10^3	5.57×10^3	5.59×10^3	5.58×10^3	1.23×10^4	6.66×10^3
		Mean	3.49×10^3	1.15×10^4	5.00×10^3	5.56×10^3	6.27×10^3	5.57×10^3	1.23×10^4	6.59×10^3
		Std	5.89×10^2	1.43×10^3	8.85×10^2	6.71×10^2	1.85×10^3	6.10×10^2	5.33×10^2	4.43×10^2
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$6.04 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$2.02 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.82 \times 10^{-6+}$
<i>f</i> ₁₁	Hybrid Functions	Median	6.11×10^1	2.04×10^2	1.58×10^2	2.15×10^2	2.90×10^2	2.08×10^2	2.34×10^3	1.93×10^2
		Mean	6.12×10^1	2.02×10^2	1.56×10^2	2.36×10^2	3.81×10^2	2.07×10^2	2.38×10^3	1.92×10^2
		Std	1.18×10^1	2.05×10^1	3.37×10^1	9.86×10^1	3.27×10^2	6.30×10^1	5.66×10^2	4.10×10^1
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₁₂		Median	8.59×10^5	2.78×10^7	5.18×10^5	1.83×10^6	1.19×10^6	2.94×10^6	3.95×10^9	1.86×10^7
		Mean	8.89×10^5	3.23×10^7	9.85×10^5	8.82×10^6	5.72×10^6	3.61×10^6	4.04×10^9	1.96×10^7
		Std	5.26×10^5	1.68×10^7	1.13×10^6	2.40×10^7	1.24×10^7	2.48×10^6	1.72×10^9	9.49×10^6
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$7.11 \times 10^{-1=}$	$2.97 \times 10^{-5+}$	$1.52 \times 10^{-2+}$	$2.24 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$

Table 4. Cont.

<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₁₃	Median		1.88×10^4	2.62×10^6	2.30×10^3	3.81×10^3	3.32×10^3	2.15×10^4	3.11×10^8	1.11×10^4
	Mean		1.63×10^4	2.73×10^6	4.40×10^3	7.76×10^3	5.58×10^3	2.29×10^4	5.22×10^8	1.13×10^4
	Std		1.12×10^4	1.72×10^6	4.74×10^3	9.15×10^3	6.22×10^3	1.32×10^4	7.74×10^8	3.04×10^3
	<i>p</i> -value		-	$1.83 \times 10^{-6+}$	$1.30 \times 10^{-4-}$	$1.44 \times 10^{-2-}$	$1.54 \times 10^{-4-}$	$9.99 \times 10^{-2=}$	$1.83 \times 10^{-6+}$	$4.60 \times 10^{-2-}$
<i>f</i> ₁₄	Median		5.52×10^4	7.90×10^3	3.86×10^4	4.09×10^4	4.01×10^4	8.86×10^4	1.56×10^6	4.64×10^5
	Mean		7.09×10^4	8.09×10^3	3.97×10^4	2.30×10^5	1.43×10^5	1.22×10^5	2.21×10^6	5.29×10^5
	Std		5.64×10^4	3.51×10^3	2.64×10^4	5.29×10^5	2.20×10^5	1.15×10^5	2.01×10^6	2.65×10^5
	<i>p</i> -value		-	$8.07 \times 10^{-6-}$	$1.36 \times 10^{-2-}$	$3.65 \times 10^{-1=}$	$7.42 \times 10^{-1=}$	$4.38 \times 10^{-2+}$	$2.02 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₁₅	Median		1.18×10^4	4.55×10^5	2.69×10^3	7.12×10^3	3.89×10^3	1.81×10^4	2.93×10^6	8.15×10^2
	Mean		1.32×10^4	4.80×10^5	4.24×10^3	1.46×10^4	5.84×10^3	1.92×10^4	1.79×10^7	9.31×10^2
	Std		9.68×10^3	1.58×10^5	4.08×10^3	2.54×10^4	6.28×10^3	1.01×10^4	2.92×10^7	4.49×10^2
	<i>p</i> -value		-	$1.83 \times 10^{-6+}$	$4.71 \times 10^{-4-}$	$2.76 \times 10^{-1=}$	$1.07 \times 10^{-3-}$	$6.41 \times 10^{-2=}$	$1.83 \times 10^{-6+}$	$6.65 \times 10^{-6-}$
<i>f</i> ₁₆	Median		4.55×10^2	3.79×10^3	9.27×10^2	1.63×10^3	1.60×10^3	1.52×10^3	3.13×10^3	1.45×10^3
	Mean		4.23×10^2	3.78×10^3	9.56×10^2	1.70×10^3	1.57×10^3	1.53×10^3	3.06×10^3	1.41×10^3
	Std		1.90×10^2	2.33×10^2	3.27×10^2	4.35×10^2	4.06×10^2	3.57×10^2	5.41×10^2	2.00×10^2
	<i>p</i> -value		-	$1.83 \times 10^{-6+}$	$6.04 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₁₇	Median		2.45×10^2	2.41×10^3	8.83×10^2	1.15×10^3	1.03×10^3	1.30×10^3	1.85×10^3	1.06×10^3
	Mean		3.11×10^2	2.40×10^3	8.56×10^2	1.17×10^3	1.05×10^3	1.20×10^3	1.83×10^3	1.04×10^3
	Std		1.43×10^2	2.00×10^2	2.56×10^2	2.97×10^2	2.23×10^2	3.24×10^2	2.86×10^2	1.94×10^2
	<i>p</i> -value		-	$1.82 \times 10^{-6+}$	$3.69 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$2.02 \times 10^{-6+}$
<i>f</i> ₁₈	Median		3.71×10^5	3.27×10^6	1.71×10^5	3.12×10^6	1.09×10^6	3.18×10^5	7.61×10^6	1.14×10^6
	Mean		4.28×10^5	3.47×10^6	3.94×10^5	5.58×10^6	3.42×10^6	4.33×10^5	1.04×10^7	1.31×10^6
	Std		2.34×10^5	1.61×10^6	4.88×10^5	5.71×10^6	5.02×10^6	3.40×10^5	9.88×10^6	7.63×10^5
	<i>p</i> -value		-	$1.83 \times 10^{-6+}$	$2.58 \times 10^{-1=}$	$4.50 \times 10^{-6+}$	$1.90 \times 10^{-3+}$	$6.81 \times 10^{-1=}$	$1.83 \times 10^{-6+}$	$2.24 \times 10^{-6+}$
<i>f</i> ₁₉	Median		1.60×10^3	2.67×10^4	9.19×10^3	1.30×10^4	1.42×10^4	1.03×10^4	1.44×10^6	3.36×10^2
	Mean		4.33×10^3	3.18×10^4	1.12×10^4	1.51×10^4	1.74×10^4	1.48×10^4	3.65×10^6	5.26×10^2
	Std		6.31×10^3	1.73×10^4	8.07×10^3	1.38×10^4	1.07×10^4	1.36×10^4	8.78×10^6	5.03×10^2
	<i>p</i> -value		-	$1.83 \times 10^{-6+}$	$1.42 \times 10^{-4+}$	$1.54 \times 10^{-3+}$	$3.26 \times 10^{-5+}$	$1.90 \times 10^{-3+}$	$1.83 \times 10^{-6+}$	$1.10 \times 10^{-4-}$
<i>f</i> _{10–19}	<i>w/t/l</i>		-	9/0/1	5/2/3	7/2/1	7/1/2	7/3/0	10/0/0	7/0/3

Table 4. Cont.

<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₂₀		Median	9.66×10^1	1.66×10^3	4.66×10^2	9.58×10^2	7.45×10^2	9.48×10^2	1.44×10^3	5.92×10^2
		Mean	1.76×10^2	1.59×10^3	4.70×10^2	9.01×10^2	7.43×10^2	8.97×10^2	1.38×10^3	6.14×10^2
		Std	1.38×10^2	3.96×10^2	2.19×10^2	2.96×10^2	2.66×10^2	2.33×10^2	2.58×10^2	1.30×10^2
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$2.97 \times 10^{-5+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₂₁		Median	2.22×10^2	6.01×10^2	2.79×10^2	3.77×10^2	3.37×10^2	3.86×10^2	6.59×10^2	4.21×10^2
		Mean	2.22×10^2	6.00×10^2	2.80×10^2	4.02×10^2	3.46×10^2	3.80×10^2	6.53×10^2	4.21×10^2
		Std	3.65×10^0	2.02×10^1	1.96×10^1	6.26×10^1	4.16×10^1	2.91×10^1	3.35×10^1	1.52×10^1
		<i>p</i> -value	-	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$
<i>f</i> ₂₂		Median	3.04×10^3	1.28×10^4	5.63×10^3	6.39×10^3	6.49×10^3	6.24×10^3	1.26×10^4	7.13×10^3
		Mean	3.10×10^3	1.27×10^4	4.80×10^3	6.14×10^3	5.88×10^3	5.76×10^3	1.13×10^4	7.14×10^3
		Std	1.34×10^3	9.78×10^2	2.20×10^3	1.52×10^3	3.61×10^3	1.63×10^3	3.48×10^3	2.75×10^2
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$1.01 \times 10^{-2+}$	$1.18 \times 10^{-5+}$	$1.33 \times 10^{-3+}$	$8.55 \times 10^{-5+}$	$2.48 \times 10^{-6+}$	$2.02 \times 10^{-6+}$
<i>f</i> ₂₃	Composition Functions	Median	5.06×10^2	8.64×10^2	5.27×10^2	6.43×10^2	6.65×10^2	6.66×10^2	1.22×10^3	6.66×10^2
		Mean	5.09×10^2	8.88×10^2	5.24×10^2	6.50×10^2	6.81×10^2	6.65×10^2	1.22×10^3	6.66×10^2
		Std	1.71×10^1	6.87×10^1	3.11×10^1	5.85×10^1	8.50×10^1	3.99×10^1	6.75×10^1	1.90×10^1
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$4.38 \times 10^{-2+}$	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₂₄		Median	5.81×10^2	1.06×10^3	5.90×10^2	7.06×10^2	7.38×10^2	7.27×10^2	1.31×10^3	8.04×10^2
		Mean	5.83×10^2	1.08×10^3	6.06×10^2	7.09×10^2	7.49×10^2	7.23×10^2	1.33×10^3	8.05×10^2
		Std	1.63×10^1	1.26×10^2	5.78×10^1	7.38×10^1	9.38×10^1	3.46×10^1	9.27×10^1	2.65×10^1
		<i>p</i> -value	-	$1.82 \times 10^{-6+}$	$3.41 \times 10^{-2+}$	$1.83 \times 10^{-6+}$	$2.02 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.82 \times 10^{-6+}$
<i>f</i> ₂₅		Median	4.80×10^2	4.31×10^2	5.98×10^2	6.75×10^2	6.61×10^2	4.80×10^2	2.60×10^3	5.31×10^2
		Mean	5.06×10^2	4.41×10^2	5.97×10^2	6.76×10^2	6.66×10^2	5.01×10^2	2.77×10^3	5.30×10^2
		Std	3.55×10^1	2.14×10^1	2.43×10^1	6.58×10^1	7.00×10^1	3.54×10^1	6.68×10^2	6.29×10^0
		<i>p</i> -value	-	$3.14 \times 10^{-6-}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$2.02 \times 10^{-6+}$	$2.33 \times 10^{-1=}$	$1.82 \times 10^{-6+}$	$5.91 \times 10^{-4+}$
<i>f</i> ₂₆		Median	2.23×10^3	6.70×10^3	9.43×10^2	3.98×10^3	2.96×10^3	3.70×10^3	7.22×10^3	3.60×10^3
		Mean	2.27×10^3	7.10×10^3	1.15×10^3	4.07×10^3	3.04×10^3	3.64×10^3	6.89×10^3	3.52×10^3
		Std	1.23×10^2	1.63×10^3	8.75×10^2	1.02×10^3	6.50×10^2	3.13×10^2	2.09×10^3	3.37×10^2
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$1.42 \times 10^{-5-}$	$1.18 \times 10^{-5+}$	$6.87 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$2.02 \times 10^{-6+}$
<i>f</i> ₂₇		Median	7.00×10^2	5.00×10^2	7.19×10^2	9.01×10^2	8.35×10^2	6.54×10^2	1.98×10^3	6.35×10^2
		Mean	6.94×10^2	5.00×10^2	7.35×10^2	9.06×10^2	8.39×10^2	6.89×10^2	1.99×10^3	6.33×10^2
		Std	5.44×10^1	0.00×10^0	8.89×10^1	9.24×10^1	8.57×10^1	1.06×10^2	1.76×10^2	2.76×10^1
		<i>p</i> -value	-	$1.82 \times 10^{-6-}$	$7.03 \times 10^{-2=}$	$1.83 \times 10^{-6+}$	$7.69 \times 10^{-6+}$	$7.50 \times 10^{-1=}$	$1.83 \times 10^{-6+}$	$7.84 \times 10^{-5-}$

Table 4. Cont.

<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₂₈		Median	5.08×10^2	5.00×10^2	5.38×10^2	6.68×10^2	7.00×10^2	4.92×10^2	2.91×10^3	1.71×10^3
		Mean	9.06×10^2	5.00×10^2	5.47×10^2	6.77×10^2	6.97×10^2	4.92×10^2	3.04×10^3	1.79×10^3
		Std	1.14×10^3	0.00×10^0	3.99×10^1	6.99×10^1	5.82×10^1	3.40×10^1	5.38×10^2	4.45×10^2
		<i>p</i> -value	-	$6.21 \times 10^{-2} =$	$7.35 \times 10^{-2} =$	$1.52 \times 10^{-2} -$	$1.52 \times 10^{-2} -$	$9.56 \times 10^{-2} =$	$3.69 \times 10^{-6} +$	$4.71 \times 10^{-4} +$
<i>f</i> ₂₉		Median	5.58×10^2	3.32×10^3	8.90×10^2	1.43×10^3	1.03×10^3	1.19×10^3	3.44×10^3	1.01×10^3
		Mean	5.90×10^2	3.27×10^3	8.66×10^2	1.43×10^3	1.08×10^3	1.16×10^3	3.52×10^3	1.02×10^3
		Std	1.75×10^2	2.69×10^2	1.87×10^2	2.49×10^2	2.75×10^2	3.49×10^2	5.07×10^2	1.49×10^2
		<i>p</i> -value	-	$1.83 \times 10^{-6} +$	$2.07 \times 10^{-5} +$	$1.83 \times 10^{-6} +$	$7.33 \times 10^{-6} +$	$2.02 \times 10^{-6} +$	$1.83 \times 10^{-6} +$	$2.48 \times 10^{-6} +$
<i>f</i> ₃₀		Median	8.17×10^5	1.79×10^6	1.93×10^6	2.00×10^6	2.00×10^6	1.16×10^6	1.38×10^8	7.30×10^5
		Mean	8.55×10^5	2.22×10^6	1.92×10^6	2.36×10^6	2.40×10^6	1.23×10^6	1.38×10^8	7.40×10^5
		Std	1.54×10^5	1.12×10^6	3.32×10^5	8.05×10^5	1.44×10^6	4.15×10^5	5.12×10^7	7.50×10^4
		<i>p</i> -value	-	$4.50 \times 10^{-6} +$	$1.83 \times 10^{-6} +$	$1.83 \times 10^{-6} +$	$1.83 \times 10^{-6} +$	$4.36 \times 10^{-4} +$	$1.83 \times 10^{-6} +$	$1.15 \times 10^{-3} -$
<i>f</i> ₂₀₋₃₀	<i>w/t/l</i>		-	8/1/2	8/2/1	10/0/1	10/0/1	8/3/0	11/0/0	9/0/2
	<i>w/t/l</i>		-	23/2/4	19/5/5	23/4/2	23/2/4	21/7/1	29/0/0	23/1/5
	<i>rank</i>		2.17	5.72	2.45	5.14	4.31	4.14	7.79	4.28

Table 5. Comparison results between the proposed STTPSO and the 7 state-of-the-art and popular PSO variants on the 100D CEC 2017 benchmark functions. The highlighted *p*-values means that the proposed STTPSO is significantly better than the associated compared algorithms on the corresponding problems.

<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₁	Unimodal Functions	Median	2.53×10^3	3.31×10^3	3.50×10^3	2.62×10^3	6.15×10^3	1.37×10^4	1.10×10^{11}	1.63×10^9
		Mean	4.29×10^3	7.08×10^3	8.27×10^3	6.22×10^3	1.21×10^4	1.93×10^7	1.10×10^{11}	1.78×10^9
		Std	4.43×10^3	1.04×10^4	9.67×10^3	7.24×10^3	1.60×10^4	1.04×10^8	1.17×10^{10}	1.48×10^9
		<i>p</i> -value	-	$7.89 \times 10^{-1} =$	$2.41 \times 10^{-1} =$	$4.34 \times 10^{-1} =$	$5.15 \times 10^{-3} +$	$7.20 \times 10^{-5} +$	$1.83 \times 10^{-6} +$	$4.08 \times 10^{-6} +$
<i>f</i> ₃		Median	2.22×10^5	1.07×10^6	7.04×10^4	2.55×10^5	7.98×10^1	8.06×10^4	3.82×10^5	5.10×10^5
		Mean	2.23×10^5	1.04×10^6	6.94×10^4	2.53×10^5	3.93×10^3	8.23×10^4	3.73×10^5	5.12×10^5
		Std	2.17×10^4	1.25×10^5	9.80×10^3	3.05×10^4	9.85×10^3	2.40×10^4	4.19×10^4	3.83×10^4
		<i>p</i> -value	-	$1.83 \times 10^{-6} +$	$1.83 \times 10^{-6} -$	$1.74 \times 10^{-4} +$	$1.83 \times 10^{-6} -$	$1.82 \times 10^{-6} -$	$1.83 \times 10^{-6} +$	$1.82 \times 10^{-6} +$
<i>f</i> ₁₋₃	<i>w/t/l</i>		-	1/1/0	0/1/1	1/1/0	1/0/1	1/0/1	2/0/0	2/0/0

Table 5. Cont.

<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₁₈		Median	3.35×10^5	3.02×10^7	3.87×10^5	2.76×10^6	3.39×10^5	1.17×10^6	2.20×10^7	7.77×10^6
		Mean	3.70×10^5	3.11×10^7	4.91×10^5	3.38×10^6	5.25×10^5	1.63×10^6	2.45×10^7	7.50×10^6
		Std	1.72×10^5	1.18×10^7	3.58×10^5	2.09×10^6	4.73×10^5	1.25×10^6	1.25×10^7	2.41×10^6
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$1.62 \times 10^{-1=}$	$1.83 \times 10^{-6+}$	$4.97 \times 10^{-1=}$	$2.74 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₁₉		Median	5.57×10^3	4.75×10^3	3.13×10^3	2.30×10^3	4.20×10^9	1.28×10^4	6.56×10^8	1.79×10^3
		Mean	7.46×10^3	7.38×10^3	4.46×10^3	4.86×10^3	4.12×10^9	1.93×10^4	6.13×10^8	1.97×10^3
		Std	7.38×10^3	6.43×10^3	5.99×10^3	6.41×10^3	7.32×10^8	1.52×10^4	2.78×10^8	7.13×10^2
		<i>p</i> -value	-	$9.67 \times 10^{-1=}$	$1.33 \times 10^{-1=}$	$8.04 \times 10^{-2=}$	$1.83 \times 10^{-6+}$	$3.06 \times 10^{-3+}$	$1.83 \times 10^{-6+}$	$2.18 \times 10^{-3-}$
<i>f</i> ₁₀₋₁₉	<i>w/t/l</i>		-	8/2/0	5/5/0	7/3/0	7/3/0	10/0/0	10/0/0	9/0/1
<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₂₀		Median	7.56×10^2	5.57×10^3	2.10×10^3	2.88×10^3	5.26×10^3	2.86×10^3	4.69×10^3	2.29×10^3
		Mean	7.88×10^2	5.20×10^3	2.16×10^3	2.83×10^3	5.23×10^3	2.76×10^3	4.69×10^3	2.34×10^3
		Std	2.72×10^2	9.09×10^2	3.96×10^2	4.93×10^2	1.93×10^2	3.59×10^2	4.05×10^2	2.09×10^2
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₂₁		Median	2.98×10^2	1.22×10^3	4.64×10^2	7.91×10^2	6.06×10^2	8.46×10^2	1.60×10^3	9.66×10^2
		Mean	2.96×10^2	1.22×10^3	4.66×10^2	7.87×10^2	6.26×10^2	8.42×10^2	1.60×10^3	9.62×10^2
		Std	9.91×10^0	3.32×10^1	5.19×10^1	8.43×10^1	7.42×10^1	6.64×10^1	8.73×10^1	3.02×10^1
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₂₂	Composition Functions	Median	8.16×10^3	3.08×10^4	1.36×10^4	1.46×10^4	1.80×10^4	1.41×10^4	3.08×10^4	2.25×10^4
		Mean	8.11×10^3	3.02×10^4	1.22×10^4	1.46×10^4	1.81×10^4	1.42×10^4	3.07×10^4	2.23×10^4
		Std	1.17×10^3	1.39×10^3	4.91×10^3	1.35×10^3	3.93×10^3	1.27×10^3	1.24×10^3	6.52×10^2
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$7.05 \times 10^{-3+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₂₃		Median	7.10×10^2	1.84×10^3	8.13×10^2	1.04×10^3	1.11×10^3	8.83×10^2	2.82×10^3	9.01×10^2
		Mean	7.18×10^2	1.89×10^3	8.09×10^2	1.07×10^3	1.12×10^3	8.89×10^2	2.85×10^3	9.01×10^2
		Std	2.95×10^1	2.76×10^2	4.95×10^1	1.09×10^2	1.66×10^2	4.44×10^1	2.15×10^2	2.66×10^1
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$2.48 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$
<i>f</i> ₂₄		Median	1.14×10^3	3.00×10^3	1.20×10^3	1.52×10^3	1.68×10^3	1.50×10^3	4.68×10^3	1.50×10^3
		Mean	1.14×10^3	3.14×10^3	1.25×10^3	1.55×10^3	1.64×10^3	1.51×10^3	4.69×10^3	1.49×10^3
		Std	5.71×10^1	6.83×10^2	1.17×10^2	1.46×10^2	2.19×10^2	6.83×10^1	4.23×10^2	2.50×10^1
		<i>p</i> -value	-	$1.83 \times 10^{-6+}$	$1.67 \times 10^{-4+}$	$1.83 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$	$1.83 \times 10^{-6+}$	$1.82 \times 10^{-6+}$

Table 5. Cont.

<i>f</i>	Category	Quality	STTPSO	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
<i>f</i> ₂₅		Median	8.21×10^2	7.62×10^2	1.09×10^3	1.29×10^3	1.38×10^3	7.63×10^2	1.10×10^4	9.02×10^2
		Mean	7.97×10^2	7.67×10^2	1.10×10^3	1.35×10^3	1.37×10^3	7.80×10^2	1.10×10^4	9.08×10^2
		Std	5.15×10^1	5.19×10^1	7.56×10^1	2.93×10^2	2.10×10^2	6.35×10^1	1.35×10^3	4.80×10^1
		<i>p</i> -value	-	3.00×10^{-2} –	1.83×10^{-6}+	1.83×10^{-6}+	1.83×10^{-6}+	1.68×10^{-1} =	1.82×10^{-6}+	1.83×10^{-6}+
<i>f</i> ₂₆		Median	6.53×10^3	2.88×10^4	5.34×10^3	1.06×10^4	8.54×10^3	1.13×10^4	2.89×10^4	1.09×10^4
		Mean	6.55×10^3	2.92×10^4	3.90×10^3	1.14×10^4	8.61×10^3	1.12×10^4	2.85×10^4	1.10×10^4
		Std	4.55×10^2	5.67×10^3	2.58×10^3	2.40×10^3	1.57×10^3	5.79×10^2	2.39×10^3	3.16×10^2
		<i>p</i> -value	-	1.83×10^{-6}+	4.97×10^{-6} –	1.83×10^{-6}+	4.08×10^{-6}+	1.83×10^{-6}+	1.83×10^{-6}+	1.83×10^{-6}+
<i>f</i> ₂₇		Median	7.39×10^2	5.00×10^2	8.54×10^2	1.12×10^3	1.02×10^3	8.05×10^2	4.00×10^3	7.58×10^2
		Mean	7.53×10^2	5.00×10^2	8.75×10^2	1.12×10^3	1.01×10^3	8.17×10^2	4.05×10^3	7.59×10^2
		Std	4.28×10^1	0.00×10^0	7.68×10^1	1.75×10^2	8.73×10^1	8.19×10^1	4.27×10^2	2.21×10^1
		<i>p</i> -value	-	1.83×10^{-6} –	3.34×10^{-6}+	1.83×10^{-6}+	1.82×10^{-6}+	1.15×10^{-3}+	1.83×10^{-6}+	3.99×10^{-1} =
<i>f</i> ₂₈		Median	5.85×10^2	5.00×10^2	8.26×10^2	1.33×10^3	1.27×10^3	5.85×10^2	1.43×10^4	1.28×10^4
		Mean	4.99×10^3	5.00×10^2	8.26×10^2	1.37×10^3	1.32×10^3	1.12×10^3	1.41×10^4	1.28×10^4
		Std	5.82×10^3	0.00×10^0	4.81×10^1	3.31×10^2	1.61×10^2	2.21×10^3	1.54×10^3	5.96×10^1
		<i>p</i> -value	-	1.82×10^{-6} –	3.88×10^{-1} =	3.88×10^{-1} =	3.88×10^{-1} =	2.33×10^{-1} =	3.34×10^{-6}+	1.13×10^{-5}+
<i>f</i> ₂₉		Median	1.76×10^3	6.76×10^3	3.08×10^3	3.91×10^3	3.53×10^3	3.95×10^3	1.03×10^4	3.30×10^3
		Mean	1.82×10^3	6.79×10^3	3.10×10^3	3.91×10^3	3.71×10^3	3.94×10^3	1.05×10^4	3.30×10^3
		Std	3.67×10^2	3.36×10^2	5.11×10^2	5.41×10^2	7.04×10^2	6.27×10^2	1.15×10^3	2.71×10^2
		<i>p</i> -value	-	1.83×10^{-6}+	1.83×10^{-6}+	1.83×10^{-6}+	1.83×10^{-6}+	1.82×10^{-6}+	1.83×10^{-6}+	1.82×10^{-6}+
<i>f</i> ₃₀		Median	4.41×10^3	7.97×10^2	2.59×10^4	1.04×10^5	2.24×10^5	1.13×10^4	2.38×10^9	5.76×10^4
		Mean	4.74×10^3	8.61×10^2	3.28×10^4	1.46×10^5	6.40×10^5	1.50×10^4	2.49×10^9	7.32×10^4
		Std	1.39×10^3	2.29×10^2	2.36×10^4	1.34×10^5	1.01×10^6	1.62×10^4	7.38×10^8	5.20×10^4
		<i>p</i> -value	-	1.83×10^{-6} –	1.83×10^{-6}+	1.83×10^{-6}+	1.83×10^{-6}+	1.07×10^{-5}+	1.83×10^{-6}+	1.83×10^{-6}+
<i>f</i> _{20–30}	<i>w/t/l</i>		-	7/0/4	9/1/1	10/1/0	10/1/0	9/2/0	11/0/0	10/1/0
	<i>w/t/l</i>		-	21/4/4	20/7/2	24/5/0	24/4/1	26/2/1	29/0/0	27/1/1
	<i>rank</i>		1.52	5.31	2.72	4.83	4.83	4.00	7.72	5.07

Table 6. Statistical comparison results between the proposed STTPSO and the 7 state-of-the-art and popular PSO variants on the CEC 2017 benchmark set with different dimensions in terms of “ $w/t/l$ ”.

Category	D	DNPSO	XPSO	TCPSO	GLPSO	HCLPSO	DPLPSO	CLPSO
Unimodal Functions	30	2/0/0	0/1/1	0/1/1	0/1/1	1/0/1	2/0/0	1/0/1
	50	1/1/0	0/1/1	0/2/0	0/1/1	1/0/1	2/0/0	1/1/0
	100	1/1/0	0/1/1	1/1/0	1/0/1	1/0/1	2/0/0	2/0/0
Simple Multimodal Functions	30	5/0/1	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0
	50	5/0/1	6/0/0	6/0/0	6/0/0	5/1/0	6/0/0	6/0/0
	100	5/1/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0
Hybrid Functions	30	7/1/2	3/6/1	6/4/0	4/5/1	7/3/0	10/0/0	5/2/3
	50	9/0/1	5/2/3	7/2/1	7/1/2	7/3/0	10/0/0	7/0/3
	100	8/2/0	5/5/0	7/3/0	7/3/0	10/0/0	10/0/0	9/0/1
Composition Functions	30	9/0/2	9/1/1	11/0/0	11/0/0	10/1/0	10/1/0	9/1/1
	50	8/1/2	8/2/1	10/0/1	10/0/1	8/3/0	11/0/0	9/0/2
	100	7/0/4	9/1/1	10/1/0	10/1/0	9/2/0	11/0/0	10/1/0
Whole Set	30	23/1/5	18/8/3	23/5/1	21/6/2	24/4/1	28/1/0	21/3/5
	50	23/2/4	19/5/5	23/4/2	23/2/4	21/7/1	29/0/0	23/1/5
	100	21/4/4	20/7/2	24/5/0	24/4/1	26/2/1	29/0/0	27/1/1

As shown in Table 4, the comparison results on the 50-*D* CEC 2017 benchmark problems are summarized below:

- (1) According to the Friedman test results shown in the last row, STTPSO achieves the lowest rank. This indicates that STTPSO still achieves the best overall performance on the whole 50-*D* CEC 2017 benchmark set. In particular, except for XPSO, its rank value (2.17) is much smaller than those (at least 4.14) of the other six compared algorithms. This demonstrates that STTPSO displays significantly better overall performance than the six compared algorithms.
- (2) From the perspective of the Wilcoxon rank sum test, as shown in the second to last row, STTPSO achieves significantly better performance than the seven compared algorithms on at least 19 problems and shows inferiority to them on, at most, five problems. In particular, compared with DNSPSO, TCSPSO, GLPSO, and CLPSO, STTPSO significantly dominates them all on 23 problems. In comparison with DPLPSO, STTPSO presents significant superiority on all the 29 problems.
- (3) In terms of different types of optimization problems, STTPSO achieves highly competitive performance with the seven compared state-of-the-art PSO variants regarding the two unimodal problems. Particularly, STTPSO defeats DPLPSO concerning these two problems. On the six simple multimodal problems, STTPSO performs much better than the seven compared algorithms on at least five problems. In particular, STTPSO presents significant dominance to XPSO, TCSPSO, GLPSO, DPLPSO, and CLPSO on all the six problems. Regarding the 10 hybrid problems, except for XPSO, STTPSO is significantly superior to the seven compared algorithms on at least seven problems, and shows inferiority on, at most, three problems. In particular, STTPSO significantly outperforms DPLPSO on all the 10 problems and obtains significantly better performance than DNSPSO on nine problems. Concerning the 11 composition problems, STTPSO displays significantly better performance than the seven state-of-the-art PSO variants on at least eight problems, and performs worse than them on, at most, two problems. Particularly, STTPSO shows significant dominance to DPLPSO on all the 11 problems and obtains much better performance than both TCSPSO and GLPSO on 10 problems. Overall, it is still demonstrated that STTPSO is a promising approach for problem optimization and displays its sound optimization ability in solving complicated optimization problems, such as multimodal problems, hybrid problems, and composition problems.

As shown in Table 5, the comparison results on the 100-*D* CEC 2017 benchmark set are summarized below:

- (1) According to the Friedman test results, STTPSO achieves the lowest rank among all algorithms. This verifies that STTPSO still obtains the best overall performance on the 100-*D* CEC 2017 benchmark set. In particular, its rank value (1.52) is much smaller than those (at least 2.72) of the seven compared algorithms. This further demonstrates that STTPSO displays significant dominance to the seven compared algorithms. Together with the observations on the 30-*D* and 50-*D* CEC 2017 benchmark set, we can see that STTPSO consistently performs the best on the CEC 2017 benchmark set with different dimension sizes among all eight algorithms, and consistently presents its significant superiority to the seven compared algorithms on the benchmark set with the three dimension sizes. Therefore, it is demonstrated that STTPSO preserves a good scalability to solve optimization problems.
- (2) Regarding the Wilcoxon rank sum test, from the second to last row, it is observed that STTPSO achieves significantly better performance than the seven compared algorithms on at least 20 problems and shows inferiority to them on, at most, four problems. In particular, STTPSO outperforms DPLPSO significantly on all the 29 problems, and obtains much better performance than TCSPSO, GLPSO, HCLPSO, and CLPSO on 24, 24, 26, and 27 problems, respectively.
- (3) With respect to the optimization performance on different types of optimization problems, STTPSO obtains highly competitive or even much better performance than the

seven compared algorithms on the two unimodal problems. Particularly, STTPSO shows significant dominance to DPLPSO and CLPSO on the two problems. As for the six simple multimodal problems, except for DNSPSO, STTPSO exhibits significant superiority to the other six compared algorithms on all these six problems. Competed with DNSPSO, STTPSO also shows much better performance on five problems. In terms of the 10 hybrid problems, except for XPSO, STTPSO is significantly superior to the other six compared algorithms on at least seven problems. Compared with XPSO, STTPSO illustrates significantly better performance on five problems and does not show inferiority on any of the problems. In particular, it is discovered that STTPSO is significantly better than HCLPSO and DPLPSO on all the 10 problems. Regarding the 11 composition problems, except for DNSPSO, STTPSO achieves much better performance than the other six compared algorithms on at least nine problems. Compared with DNSPSO, it still performs much better on seven problems. Particularly, STTPSO shows significant superiority to DPLPSO on all the 11 problems, and obtains much better performance than TCPSO, GLPSO, and CLPSO on 10 problems and shows no inferiority to the three compared methods on these kinds of problems. Overall, it is demonstrated that STTPSO is still effective at solving optimization problems, especially complicated problems, such as multimodal problems, hybrid problems, and composition problems.

To summarize, as shown in Table 6, on the CEC 2017 benchmark set with different dimension sizes, we find that the proposed STTPSO not only shows highly competitive performance against the compared state-of-the-art PSO variants on simple optimization problems, such as unimodal problems, but also achieves much better performance on complicated optimization problems, such as multimodal problems, hybrid problems and composition problems. In particular, we find that the superiority of STTPSO to the compared state-of-the-art methods is far more conspicuous regarding complicated problems, such as hybrid problems and composition problems. On the other hand, it can be concluded that STTPSO preserves a good scalability to solve optimization problems, since it consistently achieves the best overall performance on the CEC 2017 set with the three dimension sizes. Moreover, it is found that as the dimensionality increases, the superiority of STTPSO to certain compared algorithms become much more evident.

The above extensive experiments have demonstrated the effectiveness of STTPSO in solving optimization problems. To further demonstrate its efficiency in tackling optimization problems, we conduct experiments on the 50D CEC 2017 benchmark set to form convergence comparisons between STTPSO and the seven compared algorithms. Figure 1 presents the comparison results on the 16 50D CEC 2017 problems of different categories.

From Figure 1, the following observations can be obtained. (1) At a first glance, STTPSO obtains much better performance in terms of both convergence speed and solution quality on 12 problems ($f_5, f_7, f_9, f_{11}, f_{12}, f_{16}, f_{17}, f_{20}, f_{21}, f_{23}, f_{24}$, and f_{29}). (2) On f_{19} and f_{26} , STTPSO shows clear dominance to six compared methods regarding both convergence speed and solution quality, and only presents inferiority to only one of the compared methods. (3) On f_1 and f_{14} , STTPSO displays conspicuously faster convergence speed and higher solution quality than five compared methods, and only presents slight inferiority to two compared methods. (4) Overall, it is demonstrated that STTPSO could solve optimization problems with both high effectiveness and efficiency.

The superiority of STTPSO mainly benefits from the proposed stochastic triad strategy along with the devised archive, the restart mechanism and the dynamic parameter adjustment strategy. These strategies cooperate cohesively to improve the learning diversity and learning effectiveness of particles, which help the swarm explore and exploit the solution space properly to find the optima of optimization problems. To investigate the influence of the four components, we will conduct a thorough investigation on STTPSO in the following subsection.

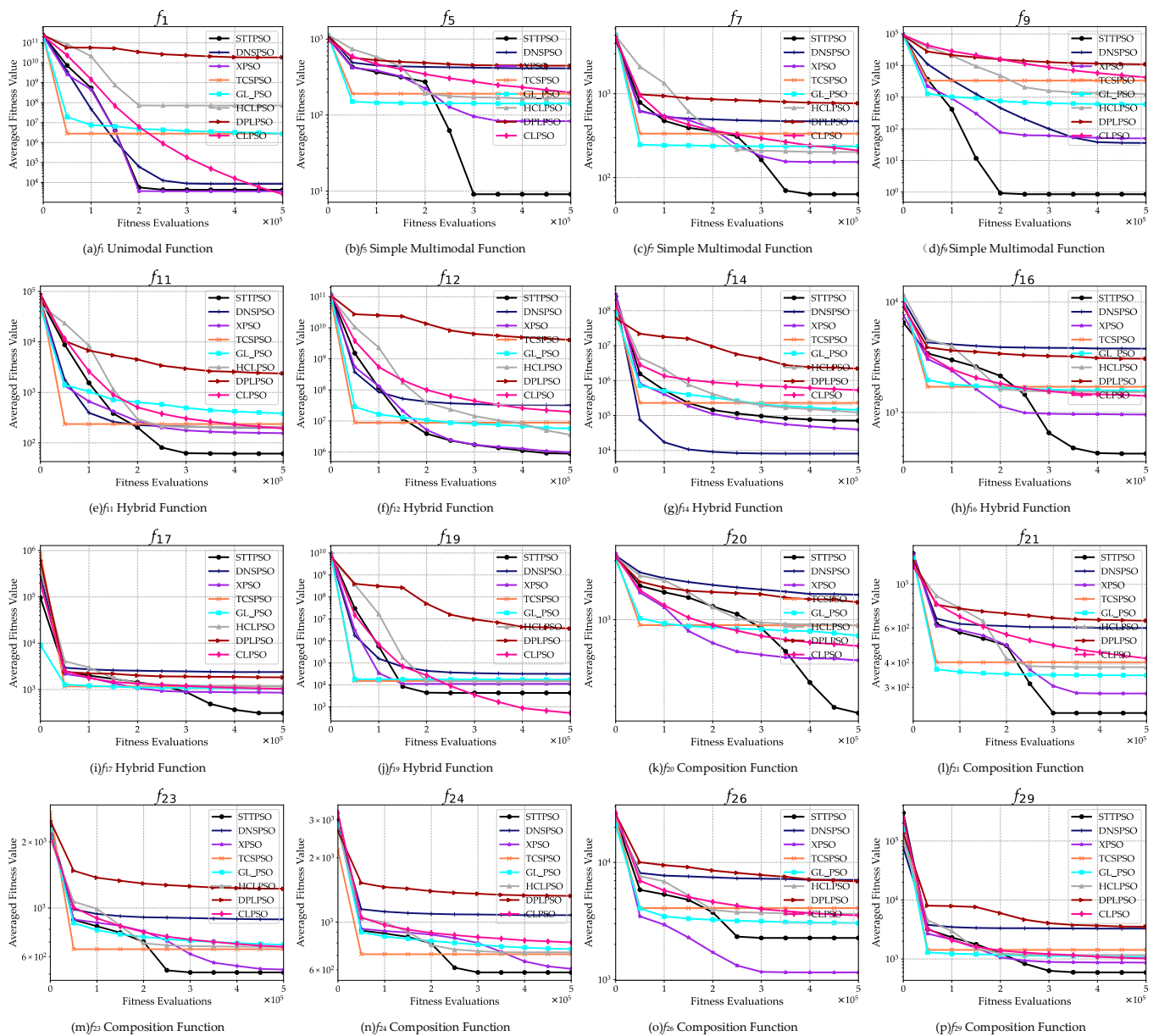


Figure 1. Convergence behavior comparisons between STTPSO and the seven compared algorithms on the 16 50D CEC 2017 benchmark problems.

4.3. Deep Investigation on STTPSO

In this section, we aim to verify the effectiveness of each component in STTPSO by conducting experiments on the 50-D CEC 2017 benchmark set.

4.3.1. Effectiveness of the Reformulation of the Stochastic Triad Topology

First, we conduct experiments to verify the effectiveness of the reformulation of the stochastic triad topology. In Section 3.1, we mentioned that in order to retain the learning effectiveness of particles, we let the triad topology structure remain fixed for each particle and then adjust it based on the evolution state of this particle. In particular, when the personal best position (*pbest*) of one particle keeps unchanged for *stop_{max}* times, we then reformulate the triad topology by randomly selecting two personal positions from those of the current swarm and the archive. In this way, both the learning effectiveness and learning diversity of particles can be guaranteed.

To verify the effectiveness of this strategy, we fix $stop_{max}$ as different values as shown in Table 7. It should be mentioned that the larger the value of $stop_{max}$, the less frequently the triad topology structure changes. In particular, when $stop_{max} = 0$, the triad structure is consequently changed in every generation. The comparison results among STTPSO with different settings of $stop_{max}$ is shown in Table 7.

From Table 7, it is found that (1) from the perspective of the Friedman test, $stop_{max} = 30$ helps STTPSO achieve the best overall performance on the 50-D CEC 2017 benchmark set. In particular, we find that the rank value (2.14) of STTPSO with $stop_{max} = 30$ is much smaller than those (at least 4.31) of STTPSO with other settings. This indicates that the superiority of $stop_{max} = 30$ is much more significant than the other settings. Moreover, we also find that STTPSO with small $stop_{max}$, such as $stop_{max} = 0$ and $stop_{max} = 5$ achieve a much worse performance. This demonstrates an absence of beneficial effects regarding STTPSO frequently changing the triad structure. (2) Through meticulous observation, we find that STTPSO with $stop_{max} = 30$ achieves the best performance on 22 problems. Concerning the other seven problems, its performance is extremely close to the STTPSO with the associated optimal settings of $stop_{max}$.

Based on the above observations, it is verified that the reformulation of the triad topology is very helpful for STTPSO to achieve promising performance. In particular, such reformulation should bear neither an excessively high frequency, nor an excessively low frequency.

4.3.2. Effectiveness of the Additional Archive and the Proposed Random Restart Strategy

Subsequently, we conduct experiments to verify the effectiveness of the additional archive and the random restart strategy. To this end, we first remove the additional archive from STTPSO, deriving a new version of STTPSO, which we name as STTPSO_WA. Then, we remove the restart strategy from STTPSO, deriving another version of STTPSO, which we denote as STTPSO_WR. Subsequently, we conduct experiments on the 50-D CEC 2017 benchmark set to compare the three versions of STTPSO. Table 8 displays the comparison results.

From Table 8, it is discovered that STTPSO with both the archive and the restart strategy achieves the best overall performance than the other two versions of STTPSO. In particular, we find that STTPSO_WR obtains the worst performance. This indicates that compared with the archive, the restart strategy is far more helpful. This is because compared with the archive, which stores the obsolete historical information, the restart strategy is more effective at improving the swarm diversity since it can introduce new solutions into the archive to promote the learning diversity of particles.

4.3.3. Effectiveness of the Dynamic Acceleration Coefficients

At last, we conduct experiments to verify the effectiveness of the devised dynamic acceleration coefficient strategy. In Section 3.2, instead of using fixed acceleration coefficients, the proposed STTPSO first randomly samples two different values based on the Gaussian distribution, and then the larger one between the two sampled values is utilized as c_1 , while the smaller one is utilized as c_2 . In this way, a promising balance between exploration and exploitation can be preserved. To validate this, we first denote the original strategy in this paper as "Dynamic". Then, we replace the settings of c_1 and c_2 with two other settings. The first is to directly utilize the sampled values as c_1 and c_2 without comparison, which we denote as "Dynamic2". The other is to utilize the smaller one between the two sampled values as c_1 , and the larger one as c_2 , which is a converse setting of the one used in this paper, which we denote as "Dynamic3". Lastly, as the baseline comparison, we adopt fixed settings for c_1 and c_2 by varying them from 1.0 to 2.0. Table 9 shows the comparison results between STTPSO with different settings of c_1 and c_2 on the 50-D CEC 2017 benchmark set.

Table 7. Comparison results between STTPSO with different settings of stagnation times on the 50D CEC 2017 functions. The best results are highlighted in bold.

<i>f</i>	<i>stop</i> _{max} = 0	<i>stop</i> _{max} = 5	<i>stop</i> _{max} = 10	<i>stop</i> _{max} = 15	<i>stop</i> _{max} = 20	<i>stop</i> _{max} = 25	<i>stop</i> _{max} = 30	<i>stop</i> _{max} = 35	<i>stop</i> _{max} = 40
<i>f</i> ₁	9.66 × 10 ⁶	1.07 × 10 ⁴	7.82 × 10 ³	6.64 × 10 ³	6.20 × 10 ³	9.07 × 10 ³	4.33 × 10³	7.55 × 10 ³	8.64 × 10 ³
<i>f</i> ₃	1.75 × 10 ⁵	7.17 × 10 ⁴	6.60 × 10 ⁴	6.66 × 10 ⁴	6.17 × 10 ⁴	6.53 × 10 ⁴	5.85 × 10⁴	6.16 × 10 ⁴	6.14 × 10 ⁴
<i>f</i> ₄	2.32 × 10 ²	1.97 × 10 ²	1.88 × 10 ²	1.91 × 10 ²	1.95 × 10 ²	1.83 × 10 ²	1.69 × 10²	1.74 × 10 ²	1.87 × 10 ²
<i>f</i> ₅	3.99 × 10 ²	9.49 × 10 ¹	1.50 × 10 ¹	1.47 × 10 ¹	1.77 × 10 ¹	1.67 × 10 ¹	9.09 × 10⁰	1.84 × 10 ¹	1.69 × 10 ¹
<i>f</i> ₆	6.74 × 10 ⁰	8.56 × 10⁻⁷	1.51 × 10 ⁻⁶	1.26 × 10 ⁻⁶	1.78 × 10 ⁻⁶	1.50 × 10 ⁻⁶	4.48 × 10 ⁻⁶	1.83 × 10 ⁻⁶	2.53 × 10 ⁻⁶
<i>f</i> ₇	4.76 × 10 ²	3.35 × 10 ²	1.16 × 10 ²	6.12 × 10¹	6.52 × 10 ¹	6.75 × 10 ¹	6.33 × 10 ¹	6.90 × 10 ¹	6.32 × 10 ¹
<i>f</i> ₈	3.97 × 10 ²	4.72 × 10 ¹	1.68 × 10 ¹	1.74 × 10 ¹	1.63 × 10 ¹	1.73 × 10 ¹	9.35 × 10⁰	1.80 × 10 ¹	1.81 × 10 ¹
<i>f</i> ₉	1.46 × 10 ²	6.68 × 10⁻¹	1.62 × 10 ⁰	1.39 × 10 ⁰	1.92 × 10 ⁰	1.93 × 10 ⁰	8.56 × 10 ⁻¹	1.19 × 10 ⁰	1.17 × 10 ⁰
<i>f</i> ₁₀	1.32 × 10 ⁴	1.25 × 10 ⁴	1.22 × 10 ⁴	1.13 × 10 ⁴	1.15 × 10 ⁴	1.14 × 10 ⁴	3.49 × 10³	1.09 × 10 ⁴	1.09 × 10 ⁴
<i>f</i> ₁₁	5.40 × 10 ²	4.67 × 10¹	5.27 × 10 ¹	5.62 × 10 ¹	5.81 × 10 ¹	6.07 × 10 ¹	6.12 × 10 ¹	5.77 × 10 ¹	6.07 × 10 ¹
<i>f</i> ₁₂	7.61 × 10 ⁷	2.82 × 10 ⁶	1.83 × 10 ⁶	2.12 × 10 ⁶	2.16 × 10 ⁶	1.64 × 10 ⁶	8.89 × 10⁵	1.64 × 10 ⁶	1.69 × 10 ⁶
<i>f</i> ₁₃	3.65 × 10 ⁴	3.01 × 10 ⁴	2.19 × 10 ⁴	3.04 × 10 ⁴	2.92 × 10 ⁴	2.72 × 10 ⁴	1.63 × 10⁴	2.70 × 10 ⁴	2.64 × 10 ⁴
<i>f</i> ₁₄	4.53 × 10 ⁵	2.17 × 10 ⁵	2.34 × 10 ⁵	1.77 × 10 ⁵	1.74 × 10 ⁵	2.21 × 10 ⁵	7.09 × 10⁴	2.19 × 10 ⁵	1.83 × 10 ⁵
<i>f</i> ₁₅	3.11 × 10 ⁴	3.13 × 10 ⁴	3.13 × 10 ⁴	2.98 × 10 ⁴	3.12 × 10 ⁴	3.11 × 10 ⁴	1.32 × 10⁴	3.00 × 10 ⁴	3.00 × 10 ⁴
<i>f</i> ₁₆	3.04 × 10 ³	1.71 × 10 ³	6.33 × 10 ²	4.70 × 10 ²	4.38 × 10 ²	5.28 × 10 ²	4.23 × 10²	5.27 × 10 ²	5.81 × 10 ²
<i>f</i> ₁₇	1.95 × 10 ³	1.04 × 10 ³	7.59 × 10 ²	5.10 × 10 ²	4.22 × 10 ²	3.80 × 10 ²	3.11 × 10²	5.28 × 10 ²	6.00 × 10 ²
<i>f</i> ₁₈	5.94 × 10 ⁶	2.78 × 10 ⁶	2.51 × 10 ⁶	2.00 × 10 ⁶	1.82 × 10 ⁶	1.82 × 10 ⁶	4.28 × 10⁵	1.51 × 10 ⁶	1.64 × 10 ⁶
<i>f</i> ₁₉	2.46 × 10 ³	2.29 × 10 ³	2.11 × 10 ³	2.02 × 10 ³	1.91 × 10 ³	1.90 × 10 ³	4.33 × 10 ³	1.69 × 10³	1.81 × 10 ³
<i>f</i> ₂₀	1.63 × 10 ³	1.29 × 10 ³	1.17 × 10 ³	1.01 × 10 ³	8.36 × 10 ²	8.01 × 10 ²	1.76 × 10²	6.15 × 10 ²	6.60 × 10 ²
<i>f</i> ₂₁	6.00 × 10 ²	2.79 × 10 ²	2.28 × 10 ²	2.28 × 10 ²	2.30 × 10 ²	2.29 × 10 ²	2.22 × 10²	2.32 × 10 ²	2.32 × 10 ²
<i>f</i> ₂₂	1.33 × 10 ⁴	1.21 × 10 ⁴	1.19 × 10 ⁴	1.05 × 10 ⁴	9.73 × 10 ³	1.02 × 10 ⁴	3.10 × 10³	9.29 × 10 ³	9.76 × 10 ³
<i>f</i> ₂₃	8.36 × 10 ²	5.13 × 10 ²	5.11 × 10 ²	5.14 × 10 ²	5.18 × 10 ²	5.18 × 10 ²	5.09 × 10²	5.21 × 10 ²	5.20 × 10 ²
<i>f</i> ₂₄	8.85 × 10 ²	5.98 × 10 ²	5.89 × 10 ²	5.90 × 10 ²	5.94 × 10 ²	5.91 × 10 ²	5.83 × 10²	5.96 × 10 ²	5.95 × 10 ²
<i>f</i> ₂₅	5.48 × 10 ²	4.81 × 10²	4.82 × 10 ²	4.83 × 10 ²	4.81 × 10 ²	4.82 × 10 ²	5.06 × 10 ²	4.84 × 10 ²	4.83 × 10 ²
<i>f</i> ₂₆	5.49 × 10 ³	2.37 × 10 ³	2.40 × 10 ³	2.45 × 10 ³	2.51 × 10 ³	2.51 × 10 ³	2.27 × 10³	2.54 × 10 ³	2.57 × 10 ³
<i>f</i> ₂₇	6.92 × 10²	7.72 × 10 ²	7.66 × 10 ²	7.52 × 10 ²	7.59 × 10 ²	7.56 × 10 ²	6.94 × 10 ²	7.59 × 10 ²	7.63 × 10 ²
<i>f</i> ₂₈	5.01 × 10 ³	4.99 × 10 ³	4.30 × 10 ³	5.00 × 10 ³	4.92 × 10 ³	4.96 × 10 ³	9.06 × 10²	4.77 × 10 ³	4.78 × 10 ³
<i>f</i> ₂₉	2.07 × 10 ³	1.32 × 10 ³	1.16 × 10 ³	1.00 × 10 ³	1.03 × 10 ³	9.13 × 10 ²	5.90 × 10²	9.35 × 10 ²	1.01 × 10 ³
<i>f</i> ₃₀	1.57 × 10 ⁶	1.52 × 10 ⁶	1.45 × 10 ⁶	1.36 × 10 ⁶	1.37 × 10 ⁶	1.34 × 10 ⁶	8.55 × 10⁵	1.15 × 10 ⁶	1.24 × 10 ⁶
<i>rank</i>	8.55	6.52	5.14	4.31	4.69	4.62	2.14	4.31	4.72

Table 8. Comparison results between STTPSO with and without the archive and the restart strategy on the 50-D CEC 2017 functions. The best results are highlighted in bold.

<i>f</i>	STTPSO	STTPSO_WR	STTPSO_WA
<i>f</i> ₁	4.33 × 10 ³	8.59 × 10 ³	5.75 × 10 ³
<i>f</i> ₃	5.85 × 10 ⁴	6.82 × 10 ⁴	6.08 × 10 ⁴
<i>f</i> ₄	1.69 × 10 ²	1.96 × 10 ²	1.96 × 10 ²
<i>f</i> ₅	9.09 × 10 ⁰	1.64 × 10 ¹	1.46 × 10 ¹
<i>f</i> ₆	4.48 × 10 ^{−6}	2.38 × 10 ^{−6}	3.85 × 10 ^{−6}
<i>f</i> ₇	6.33 × 10 ¹	9.71 × 10 ¹	5.95 × 10 ¹
<i>f</i> ₈	9.35 × 10 ⁰	1.56 × 10 ¹	1.45 × 10 ¹
<i>f</i> ₉	8.56 × 10 ^{−1}	1.59 × 10 ⁰	1.57 × 10 ⁰
<i>f</i> ₁₀	3.49 × 10 ³	1.21 × 10 ⁴	1.18 × 10 ⁴
<i>f</i> ₁₁	6.12 × 10 ¹	5.07 × 10 ¹	5.07 × 10 ¹
<i>f</i> ₁₂	8.89 × 10 ⁵	2.46 × 10 ⁶	1.85 × 10 ⁶
<i>f</i> ₁₃	1.63 × 10 ⁴	2.52 × 10 ⁴	2.46 × 10 ⁴
<i>f</i> ₁₄	7.09 × 10 ⁴	2.17 × 10 ⁵	1.88 × 10 ⁵
<i>f</i> ₁₅	1.32 × 10 ⁴	3.13 × 10 ⁴	2.70 × 10 ⁴
<i>f</i> ₁₆	4.23 × 10 ²	6.87 × 10 ²	6.41 × 10 ²
<i>f</i> ₁₇	3.11 × 10 ²	5.46 × 10 ²	3.78 × 10 ²
<i>f</i> ₁₈	4.28 × 10 ⁵	1.99 × 10 ⁶	1.63 × 10 ⁶
<i>f</i> ₁₉	4.33 × 10 ³	2.16 × 10 ³	2.03 × 10 ³
<i>f</i> ₂₀	1.76 × 10 ²	1.12 × 10 ³	8.45 × 10 ²
<i>f</i> ₂₁	2.22 × 10 ²	2.29 × 10 ²	2.27 × 10 ²
<i>f</i> ₂₂	3.10 × 10 ³	1.15 × 10 ⁴	1.01 × 10 ⁴
<i>f</i> ₂₃	5.09 × 10 ²	5.12 × 10 ²	5.16 × 10 ²
<i>f</i> ₂₄	5.83 × 10 ²	5.94 × 10 ²	5.98 × 10 ²
<i>f</i> ₂₅	5.06 × 10 ²	4.81 × 10 ²	4.82 × 10 ²
<i>f</i> ₂₆	2.27 × 10 ³	2.40 × 10 ³	2.47 × 10 ³
<i>f</i> ₂₇	6.94 × 10 ²	7.56 × 10 ²	7.47 × 10 ²
<i>f</i> ₂₈	9.06 × 10 ²	4.96 × 10 ³	4.34 × 10 ³
<i>f</i> ₂₉	5.90 × 10 ²	1.01 × 10 ³	8.27 × 10 ²
<i>f</i> ₃₀	8.55 × 10 ⁵	1.30 × 10 ⁶	1.12 × 10 ⁶
<i>rank</i>	1.31	2.62	2.07

From Table 9, it is observed that from the average rank obtained from the Friedman test, the proposed dynamic strategy for c_1 and c_2 helps STTPSO achieve the best overall performance among all setting versions of c_1 and c_2 . This demonstrates that the proposed dynamic strategy is extremely effective for STTPSO to achieve good performance. In particular, compared with the fixed settings, the proposed dynamic strategy helps STTPSO achieve much better performance than all the fixed settings. This demonstrates the dynamic sampling of c_1 and c_2 is far more effective than fixed ones. In comparison with the other two dynamic strategies, STTPSO with the proposed dynamic strategy obtains significantly better performance than those with the other two dynamic strategies. This demonstrates utilization of the larger one between the sampled values as c_1 and the smaller one as c_2 is far more effective. Together, we can observe that the proposed dynamic acceleration coefficient strategy is helpful in order for STTPSO to achieve good performance.

Table 9. Comparison results between STTPSO with different acceleration coefficient settings on the 50D CEC 2017 functions. The best results are highlighted in bold.

<i>f</i>	Dynamic	Dynamic1	Dynamic2	<i>c</i> ₁ = 1.0			<i>c</i> ₁ = 1.5			<i>c</i> ₁ = 2.0		
				<i>c</i> ₂ = 1.0	<i>c</i> ₂ = 1.5	<i>c</i> ₂ = 2.0	<i>c</i> ₂ = 1.0	<i>c</i> ₂ = 1.5	<i>c</i> ₂ = 2.0	<i>c</i> ₂ = 1.0	<i>c</i> ₂ = 1.5	<i>c</i> ₂ = 2.0
<i>f</i> ₁	4.33 × 10 ³	7.70 × 10 ³	8.72 × 10 ³	2.44 × 10³	4.08 × 10 ³	7.78 × 10 ³	6.41 × 10 ³	7.75 × 10 ³	1.74 × 10 ⁴	1.37 × 10 ⁴	1.93 × 10 ⁴	5.31 × 10 ⁷
<i>f</i> ₃	5.85 × 10 ⁴	6.84 × 10 ⁴	6.43 × 10 ⁴	6.42 × 10 ⁴	7.20 × 10 ⁴	7.39 × 10 ⁴	5.43 × 10⁴	6.76 × 10 ⁴	9.44 × 10 ⁴	6.18 × 10 ⁴	9.54 × 10 ⁴	1.46 × 10 ⁵
<i>f</i> ₄	1.69 × 10 ²	1.91 × 10 ²	1.90 × 10 ²	1.18 × 10²	1.75 × 10 ²	1.94 × 10 ²	1.91 × 10 ²	1.84 × 10 ²	1.93 × 10 ²	1.80 × 10 ²	1.93 × 10 ²	2.33 × 10 ²
<i>f</i> ₅	9.09 × 10 ⁰	1.68 × 10 ¹	1.82 × 10 ¹	9.78 × 10 ⁰	8.50 × 10⁰	1.08 × 10 ¹	1.72 × 10 ¹	1.71 × 10 ¹	2.90 × 10 ²	2.89 × 10 ¹	2.94 × 10 ²	3.90 × 10 ²
<i>f</i> ₆	4.48 × 10 ^{−6}	2.37 × 10 ^{−4}	2.39 × 10 ^{−6}	7.32 × 10 ^{−5}	5.44 × 10 ^{−6}	1.27 × 10^{−6}	8.07 × 10 ^{−4}	1.50 × 10 ^{−6}	3.54 × 10 ^{−4}	3.90 × 10 ^{−5}	3.99 × 10 ^{−3}	3.02 × 10 ⁰
<i>f</i> ₇	6.33 × 10 ¹	9.75 × 10 ¹	7.80 × 10 ¹	6.67 × 10 ¹	6.24 × 10 ¹	1.80 × 10 ²	6.07 × 10¹	9.58 × 10 ¹	3.61 × 10 ²	1.03 × 10 ²	3.65 × 10 ²	4.44 × 10 ²
<i>f</i> ₈	9.35 × 10 ⁰	1.70 × 10 ¹	1.76 × 10 ¹	8.57 × 10 ⁰	8.18 × 10⁰	1.13 × 10 ¹	1.68 × 10 ¹	1.77 × 10 ¹	2.85 × 10 ²	2.87 × 10 ¹	2.74 × 10 ²	3.88 × 10 ²
<i>f</i> ₉	8.56 × 10 ^{−1}	1.74 × 10 ⁰	1.60 × 10 ⁰	6.85 × 10 ^{−1}	3.14 × 10^{−1}	8.52 × 10 ^{−1}	1.22 × 10 ⁰	1.49 × 10 ⁰	2.31 × 10 ⁰	4.38 × 10 ⁰	3.46 × 10 ⁰	2.95 × 10 ²
<i>f</i> ₁₀	3.49 × 10³	1.22 × 10 ⁴	1.21 × 10 ⁴	3.52 × 10 ³	7.09 × 10 ³	1.22 × 10 ⁴	7.77 × 10 ³	1.23 × 10 ⁴	1.27 × 10 ⁴	1.14 × 10 ⁴	1.29 × 10 ⁴	1.28 × 10 ⁴
<i>f</i> ₁₁	6.12 × 10 ¹	5.16 × 10 ¹	5.39 × 10 ¹	7.72 × 10 ¹	5.91 × 10 ¹	4.65 × 10¹	5.81 × 10 ¹	5.04 × 10 ¹	1.72 × 10 ²	8.09 × 10 ¹	1.91 × 10 ²	4.75 × 10 ²
<i>f</i> ₁₂	8.89 × 10 ⁵	2.18 × 10 ⁶	2.40 × 10 ⁶	7.14 × 10⁵	1.23 × 10 ⁶	1.87 × 10 ⁶	1.46 × 10 ⁶	2.16 × 10 ⁶	5.52 × 10 ⁶	3.38 × 10 ⁶	6.65 × 10 ⁶	1.08 × 10 ⁸
<i>f</i> ₁₃	1.63 × 10 ⁴	3.26 × 10 ⁴	3.00 × 10 ⁴	9.33 × 10³	1.82 × 10 ⁴	2.72 × 10 ⁴	2.11 × 10 ⁴	3.14 × 10 ⁴	3.61 × 10 ⁴	3.65 × 10 ⁴	3.63 × 10 ⁴	3.77 × 10 ⁴
<i>f</i> ₁₄	7.09 × 10⁴	2.77 × 10 ⁵	2.14 × 10 ⁵	7.15 × 10 ⁴	1.02 × 10 ⁵	1.81 × 10 ⁵	1.93 × 10 ⁵	2.41 × 10 ⁵	2.84 × 10 ⁵	2.43 × 10 ⁵	2.40 × 10 ⁵	4.09 × 10 ⁵
<i>f</i> ₁₅	1.32 × 10 ⁴	3.12 × 10 ⁴	3.14 × 10 ⁴	3.71 × 10³	2.15 × 10 ⁴	2.97 × 10 ⁴	3.07 × 10 ⁴	3.12 × 10 ⁴	3.15 × 10 ⁴	3.15 × 10 ⁴	3.15 × 10 ⁴	3.19 × 10 ⁴
<i>f</i> ₁₆	4.23 × 10 ²	6.14 × 10 ²	5.99 × 10 ²	4.96 × 10 ²	4.02 × 10 ²	6.90 × 10 ²	3.92 × 10²	5.46 × 10 ²	2.26 × 10 ³	6.08 × 10 ²	2.16 × 10 ³	2.80 × 10 ³
<i>f</i> ₁₇	3.11 × 10 ²	7.20 × 10 ²	6.73 × 10 ²	3.01 × 10 ²	3.85 × 10 ²	7.63 × 10 ²	2.39 × 10²	5.93 × 10 ²	1.38 × 10 ³	4.69 × 10 ²	1.26 × 10 ³	1.78 × 10 ³
<i>f</i> ₁₈	4.28 × 10 ⁵	2.45 × 10 ⁶	2.39 × 10 ⁶	3.80 × 10⁵	6.79 × 10 ⁵	1.55 × 10 ⁶	1.64 × 10 ⁶	2.45 × 10 ⁶	3.21 × 10 ⁶	2.25 × 10 ⁶	3.06 × 10 ⁶	4.03 × 10 ⁶
<i>f</i> ₁₉	4.33 × 10 ³	2.09 × 10 ³	2.20 × 10 ³	5.91 × 10 ³	1.75 × 10 ³	1.86 × 10 ³	1.70 × 10³	2.14 × 10 ³	2.32 × 10 ³	2.37 × 10 ³	2.40 × 10 ³	2.48 × 10 ³
<i>f</i> ₂₀	1.76 × 10 ²	1.13 × 10 ³	1.12 × 10 ³	1.73 × 10²	4.35 × 10 ²	1.11 × 10 ³	4.18 × 10 ²	1.17 × 10 ³	1.32 × 10 ³	1.11 × 10 ³	1.36 × 10 ³	1.47 × 10 ³
<i>f</i> ₂₁	2.22 × 10 ²	2.27 × 10 ²	2.30 × 10 ²	2.32 × 10 ²	2.33 × 10 ²	2.22 × 10²	2.30 × 10 ²	2.30 × 10 ²	4.92 × 10 ²	2.43 × 10 ²	4.82 × 10 ²	5.92 × 10 ²
<i>f</i> ₂₂	3.10 × 10³	1.17 × 10 ⁴	1.18 × 10 ⁴	3.24 × 10 ³	4.42 × 10 ³	1.18 × 10 ⁴	4.83 × 10 ³	1.18 × 10 ⁴	1.25 × 10 ⁴	1.23 × 10 ⁴	1.28 × 10 ⁴	1.29 × 10 ⁴
<i>f</i> ₂₃	5.09 × 10 ²	5.18 × 10 ²	5.16 × 10 ²	5.36 × 10 ²	5.40 × 10 ²	5.04 × 10²	5.22 × 10 ²	5.15 × 10 ²	6.38 × 10 ²	5.27 × 10 ²	6.32 × 10 ²	8.21 × 10 ²
<i>f</i> ₂₄	5.83 × 10 ²	5.89 × 10 ²	5.87 × 10 ²	6.04 × 10 ²	6.06 × 10 ²	5.73 × 10²	5.90 × 10 ²	5.90 × 10 ²	7.90 × 10 ²	5.88 × 10 ²	7.45 × 10 ²	8.85 × 10 ²
<i>f</i> ₂₅	5.06 × 10 ²	4.83 × 10 ²	4.84 × 10 ²	5.51 × 10 ²	5.16 × 10 ²	4.80 × 10²	4.84 × 10 ²	4.82 × 10 ²	5.24 × 10 ²	5.16 × 10 ²	5.31 × 10 ²	5.69 × 10 ²
<i>f</i> ₂₆	2.27 × 10 ³	2.44 × 10 ³	2.49 × 10 ³	2.15 × 10³	2.34 × 10 ³	2.28 × 10 ³	2.40 × 10 ³	2.42 × 10 ³	2.65 × 10 ³	2.56 × 10 ³	2.67 × 10 ³	5.09 × 10 ³
<i>f</i> ₂₇	6.94 × 10²	7.43 × 10 ²	7.67 × 10 ²	7.46 × 10 ²	7.31 × 10 ²	7.07 × 10 ²	7.75 × 10 ²	7.53 × 10 ²	7.42 × 10 ²	8.28 × 10 ²	8.04 × 10 ²	7.80 × 10 ²
<i>f</i> ₂₈	9.06 × 10 ²	5.13 × 10 ³	5.10 × 10 ³	5.29 × 10 ²	5.27 × 10²	2.37 × 10 ³	4.60 × 10 ³	5.14 × 10 ³	5.30 × 10 ³	5.67 × 10 ³	5.66 × 10 ³	5.74 × 10 ³
<i>f</i> ₂₉	5.90 × 10 ²	1.14 × 10 ³	1.33 × 10 ³	5.62 × 10²	5.92 × 10 ²	8.11 × 10 ²	1.13 × 10 ³	1.13 × 10 ³	1.43 × 10 ³	1.80 × 10 ³	1.62 × 10 ³	1.80 × 10 ³
<i>f</i> ₃₀	8.55 × 10 ⁵	1.48 × 10 ⁶	1.38 × 10 ⁶	8.27 × 10⁵	9.18 × 10 ⁵	1.33 × 10 ⁶	1.29 × 10 ⁶	1.41 × 10 ⁶	1.53 × 10 ⁶	1.59 × 10 ⁶	1.54 × 10 ⁶	1.51 × 10 ⁶
rank	2.83	6.45	6.21	3.66	3.79	4.66	4.55	5.97	9.72	8.10	10.38	11.69

5. Conclusions

This paper has devised a stochastic triad topology-based particle swarm optimization (STTPSO) algorithm to solve optimization problems. Specifically, in this optimizer, for each particle, a triad topology is utilized to connect the personal best position of this particle and two other personal best positions randomly selected from those of particles in the current swarm and an additional archive used to store obsolete historical best positions. Then, the best one in the topology and the mean position of the connected triad personal best positions are employed to update each particle. In addition, during the evolution, the triad topology structure of each particle is dynamically updated based on its evolution state. In this way, the learning diversity and learning effectiveness of particles could be largely promoted, so that the swarm could explore and exploit the solution space appropriately. To further improve the swarm diversity, a random restart strategy is proposed by randomly initializing a feasible solution and then inserting into the archive. To alleviate the sensitivity of STTPSO to the acceleration coefficients, a dynamic acceleration coefficient strategy is devised based on the Gaussian distribution. With the above mechanisms, the proposed STTPSO is expected to search the solution space with proper intensification and diversification to achieve promising performance.

Extensive comparative experiments conducted on the CEC 2017 benchmark set with three different dimension sizes have demonstrated the effectiveness of STTPSO. Compared with seven state-of-the-art PSO variants, the proposed STTPSO consistently achieves the best overall performance on the CEC 2017 set with the three dimension sizes. In particular, we find that STTPSO exhibits much better performance than the compared methods regarding complicated optimization problems, such as hybrid problems and composition problems. In addition, the experimental results verified that STTPSO preserves a good scalability to solve optimization problems. In depth investigation on the proposed STTPSO was also conducted to validate the effectiveness of each component in STTPSO. Experimental results demonstrated each component as being of great benefit for STTPSO to achieve good performance.

However, from Tables 3–5, we can see that the results obtained by STTPSO on certain problems remain far from the true optima. Therefore, its optimization performance still requires improvement. In this paper, we adjusted the parameters in STTPSO dynamically without considering the evolution state of particles and the difference between particles. As a result, in future, we will mainly focus on devising adaptive parameter adjustment strategies by considering both the difference between particles and the evolution state of particles to further promote the optimization ability of STTPSO.

Author Contributions: Q.Y.: Conceptualization, supervision, methodology, formal analysis, and writing—original draft preparation. Y.-W.B.: Implementation, formal analysis, and writing—original draft preparation. X.-D.G.: Methodology, and writing—review and editing. D.-D.X.: Writing—review and editing. Z.-Y.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.

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

References

1. Li, L.; Chang, L.; Gu, T.; Sheng, W.; Wang, W. On the Norm of Dominant Difference for Many-Objective Particle Swarm Optimization. *IEEE Trans. Cybern.* **2021**, *51*, 2055–2067. [[CrossRef](#)] [[PubMed](#)]
2. Xia, X.; Gui, L.; Yu, F.; Wu, H.; Wei, B.; Zhang, Y.L.; Zhan, Z.H. Triple Archives Particle Swarm Optimization. *IEEE Trans. Cybern.* **2020**, *50*, 4862–4875. [[CrossRef](#)] [[PubMed](#)]
3. 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.* **2021**, *51*, 1085–1093. [[CrossRef](#)] [[PubMed](#)]
4. 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](#)]
5. Yang, Q.; Chen, W.-N.; Zhang, J. Probabilistic Multimodal Optimization. In *Metaheuristics for Finding Multiple Solutions*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 191–228.
6. 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.
7. Shi, Y.; Eberhart, R.C. Empirical Study of Particle Swarm Optimization. In Proceedings of the Congress on Evolutionary Computation, Washington, DC, USA, 6–9 July 1999; pp. 1945–1950.
8. Tsekouras, G.E.; Tsimikas, J.; Kalloniatis, C.; Gritzalis, S. Interpretability Constraints for Fuzzy Modeling Implemented by Constrained Particle Swarm Optimization. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 2348–2361. [[CrossRef](#)]
9. Lin, C.; Chen, C.; Lin, C. Efficient Self-Evolving Evolutionary Learning for Neurofuzzy Inference Systems. *IEEE Trans. Fuzzy Syst.* **2008**, *16*, 1476–1490.
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. Zhang, X.; Du, K.J.; Zhan, Z.H.; Kwong, S.; Gu, T.L.; Zhang, J. Cooperative Coevolutionary Bare-Bones Particle Swarm Optimization with Function Independent Decomposition for Large-Scale Supply Chain Network Design with Uncertainties. *IEEE Trans. Cybern.* **2020**, *50*, 4454–4468. [[CrossRef](#)]
12. 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.* **2020**, *50*, 4053–4065. [[CrossRef](#)]
13. Ge, Q.; Guo, C.; Jiang, H.; Lu, Z.; Yao, G.; Zhang, J.; Hua, Q. Industrial Power Load Forecasting Method Based on Reinforcement Learning and PSO-LSSVM. *IEEE Trans. Cybern.* **2022**, *52*, 1112–1124. [[CrossRef](#)]
14. Cao, Y.; Zhang, H.; Li, W.; Zhou, M.; Zhang, Y.; Chaovallitwongse, W.A. Comprehensive Learning Particle Swarm Optimization Algorithm With Local Search for Multimodal Functions. *IEEE Trans. Evol. Comput.* **2019**, *23*, 718–731. [[CrossRef](#)]
15. Liang, J.J.; Suganthan, P.N. Dynamic Multi-swarm Particle Swarm Optimizer. In Proceedings of the IEEE Swarm Intelligence Symposium, Pasadena, CA, USA, 8–10 June 2005; pp. 124–129.
16. Mirjalili, S.; Lewis, A. Obstacles and Difficulties for Robust Benchmark Problems: A Novel Penalty-based Robust Optimisation Method. *Inf. Sci.* **2016**, *328*, 485–509. [[CrossRef](#)]
17. 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](#)]
18. 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](#)] [[PubMed](#)]
19. 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](#)]
20. Liu, Z.; Nishi, T. Strategy Dynamics Particle Swarm Optimizer. *Inf. Sci.* **2022**, *582*, 665–703. [[CrossRef](#)]
21. Kennedy, J.; Mendes, R. Population Structure and Particle Swarm Performance. In Proceedings of the Congress on Evolutionary Computation, Honolulu, HI, USA, 12–17 May 2002; pp. 1671–1676.
22. Clerc, M. Beyond Standard Particle Swarm Optimisation. *Int. J. Swarm Intell. Res.* **2010**, *1*. [[CrossRef](#)]
23. Xia, X.; Gui, L.; He, G.; Wei, B.; Zhang, Y.; Yu, F.; Wu, H.; Zhan, Z.-H. An Expanded Particle Swarm Optimization Based on Multi-exemplar and Forgetting Ability. *Inf. Sci.* **2020**, *508*, 105–120. [[CrossRef](#)]
24. Karim, A.A.; Isa, N.A.M.; Lim, W.H. Modified Particle Swarm Optimization with Effective Guides. *IEEE Access* **2020**, *8*, 188699–188725. [[CrossRef](#)]
25. Lynn, N.; Suganthan, P.N. Heterogeneous Comprehensive Learning Particle Swarm Optimization with Enhanced Exploration and Exploitation. *Swarm Evol. Comput.* **2015**, *24*, 11–24. [[CrossRef](#)]
26. Yue, C.; Qu, B.; Liang, J. A Multiobjective Particle Swarm Optimizer Using Ring Topology for Solving Multimodal Multiobjective Problems. *IEEE Trans. Evol. Comput.* **2018**, *22*, 805–817. [[CrossRef](#)]
27. Chakraborty, A.; Ray, K.S.; Dutta, S.; Bhattacharyya, S.; Kolya, A. Species Inspired PSO based Pyramid Match Kernel Model (PMK) for Moving Object Motion Tracking. In Proceedings of the Fourth International Conference on Research in Computational Intelligence and Communication Networks, Kolkata, India, 22–23 November 2018; pp. 152–157.
28. Nianyin, Z.; Zidong, W.; Weibo, L.; Hong, Z.; Kate, H.; Xiaohui, L. A Dynamic Neighborhood-based Switching Particle Swarm Optimization Algorithm. *IEEE Trans. Cybern.* **2020**, *8*, 701–717.
29. Vazquez, J.C.; Valdez, F. Fuzzy Logic for Dynamic Adaptation in PSO with Multiple Topologies. In Proceedings of the 2013 Joint IFSA World Congress and NAFIPS Annual Meeting, Edmonton, AB, Canada, 24–28 June 2013; pp. 1197–1202.

30. Liu, Q.; Wei, W.; Yuan, H.; Zhan, Z.-H.; Li, Y. Topology Selection for Particle Swarm Optimization. *Inf. Sci.* **2016**, *363*, 154–173. [[CrossRef](#)]
31. Lin, W.; Bo, Y.; Jeff, O. Particle Swarm Optimization Using Dynamic Tournament Topology. *Appl. Soft Comput.* **2016**, *48*, 584–596.
32. Xia, X.; Gui, L.; Zhan, Z.-H. A Multi-swarm Particle Swarm Optimization Algorithm based on Dynamical Topology and Purposeful Detecting. *Appl. Soft Comput.* **2018**, *67*, 126–140. [[CrossRef](#)]
33. 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](#)]
34. Parrott, D.; Xiaodong, L. Locating and Tracking Multiple Dynamic Optima by a Particle Swarm Model Using Speciation. *IEEE Trans. Evol. Comput.* **2006**, *10*, 440–458. [[CrossRef](#)]
35. Cervantes, A.; Galvan, I.M.; Isasi, P. AMPSO: A New Particle Swarm Method for Nearest Neighborhood Classification. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2009**, *39*, 1082–1091. [[CrossRef](#)]
36. Janson, S.; Middendorf, M. A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2005**, *35*, 1272–1282. [[CrossRef](#)]
37. Zhan, Z.; Zhang, J.; Li, Y.; Shi, Y. Orthogonal Learning Particle Swarm Optimization. *IEEE Trans. Evol. Comput.* **2011**, *15*, 832–847. [[CrossRef](#)]
38. Chen, S.; Hong, X.; Harris, C.J. Particle Swarm Optimization Aided Orthogonal Forward Regression for Unified Data Modeling. *IEEE Trans. Evol. Comput.* **2010**, *14*, 477–499. [[CrossRef](#)]
39. 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.* **2017**, *21*, 191–205. [[CrossRef](#)]
40. Yang, Q.; Chen, W.N.; Li, Y.; Chen, C.L.P.; Xu, X.M.; Zhang, J. Multimodal Estimation of Distribution Algorithms. *IEEE Trans. Cybern.* **2017**, *47*, 636–650. [[CrossRef](#)] [[PubMed](#)]
41. 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](#)]
42. Wei, F.F.; Chen, W.N.; Yang, Q.; Deng, J.; Zhang, J. A Classifier-Assisted Level-Based Learning Swarm Optimizer for Expensive Optimization. *IEEE Trans. Evol. Comput.* **2020**, *25*, 219–233. [[CrossRef](#)]
43. 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](#)]
44. Wu, G.; Mallipeddi, R.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for The CEC 2017 Competition on Constrained Real-Parameter Optimization*; Technical Report; National University of Defense Technology: Changsha, China; Kyungpook National University: Daegu, Korea; Nanyang Technological University: Singapore, 2017; pp. 1–16.
45. Shen, Y.; Wei, L.; Zeng, C.; Chen, J. Particle Swarm Optimization with Double Learning Patterns. *Comput. Intell. Neurosci.* **2016**, *2016*, 6510303. [[CrossRef](#)]
46. Xie, H.-Y.; Yang, Q.; Hu, X.-M.; Chen, W.N. Cross-generation Elites Guided Particle Swarm Optimization for large scale optimization. In Proceedings of the IEEE Symposium Series on Computational Intelligence, Athens, Greece, 6–9 December 2016; pp. 1–8.
47. 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.* **2017**, *47*, 2896–2910. [[CrossRef](#)]
48. Hesam, V.; Naser, S.H.; Mahsa, S. A Hybrid Generalized Reduced Gradient-based Particle Swarm Optimizer for Constrained Engineering Optimization Problems. *J. Comput. Civ. Eng.* **2021**, *5*, 86–119.
49. Riaan, B.; Engelbrecht, A.P.; van den Bergh, F. A Niche Particle Swarm Optimizer. In Proceedings of the Asia-Pacific Conference on Simulated Evolution and Learning, Orchid Country Club, Singapore, 18–22 November 2002; pp. 692–696.
50. Yousri, D.; Thanikanti, S.B.; Allam, D.; Ramachandaramurthy, V.K.; Eteiba, M.B. Fractional Chaotic Ensemble Particle Swarm Optimizer for Identifying the Single, Double, and Three Diode Photovoltaic Models' Parameters. *Energy* **2020**, *195*, 116979. [[CrossRef](#)]
51. Chen, X.; Tianfield, H.; Du, W. Bee-foraging Learning Particle Swarm Optimization. *Appl. Soft Comput.* **2021**, *102*, 107134. [[CrossRef](#)]
52. Zhan, Z.-H.; Shi, L.; Tan, K.C.; Zhang, J. A Survey on Evolutionary Computation for Complex Continuous optimization. *Artif. Intell. Rev.* **2021**, *55*, 59–110. [[CrossRef](#)]
53. Tao, X.; Li, X.; Chen, W.; Liang, T.; Qi, L. Self-Adaptive Two Roles Hybrid Learning Strategies-based Particle Swarm Optimization. *Inf. Sci.* **2021**, *578*. [[CrossRef](#)]
54. Xu, G.; Zhao, X.; Wu, T.; Li, R.; Li, X. An Elitist Learning Particle Swarm Optimization with Scaling Mutation and Ring Topology. *IEEE Access* **2018**, *6*, 78453–78470. [[CrossRef](#)]
55. Kennedy, J. Small Worlds and Mega-minds: Effects of Neighborhood Topology on Particle Swarm Performance. In Proceedings of the Congress on Evolutionary Computation, Washington, DC, USA, 6–9 July 1999; Volume 1933, pp. 1931–1938.
56. 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.* **2019**, *44*, 571–583. [[CrossRef](#)]
57. Turkey, M.; Poli, R. A Model for Analysing the Collective Dynamic Behaviour and Characterising the Exploitation of Population-Based Algorithms. *Evol. Comput.* **2014**, *22*, 159–188. [[CrossRef](#)]

58. Zhang, J.; Sanderson, A.C. JADE: Adaptive Differential Evolution with Optional External Archive. *IEEE Trans. Evol. Comput.* **2009**, *13*, 945–958. [[CrossRef](#)]
59. Djellali, H.; Ghoulmi, N. Improved Chaotic Initialization of Particle Swarm applied to Feature Selection. In Proceedings of the International Conference on Networking and Advanced Systems, Annaba, Algeria, 26–27 June 2019; pp. 1–5.
60. Watanabe, M.; Ihara, K.; Kato, S.; Sakuma, T. Initialization Effects for PSO Based Storage Assignment Optimization. In Proceedings of the Global Conference on Consumer Electronics, Kyoto, Japan, 12–15 October 2021; pp. 494–495.
61. Wang, C.J.; Fang, H.; Wang, C.; Daneshmand, M.; Wang, H. A Novel Initialization Method for Particle Swarm Optimization-based FCM in Big Biomedical Data. In Proceedings of the IEEE International Conference on Big Data, Santa Clara, CA, USA, 29 October–1 November 2015; pp. 2942–2944.
62. Farooq, M.U.; Ahmad, A.; Hameed, A. Opposition-based Initialization and A Modified Pattern for Inertia Weight (IW) in PSO. In Proceedings of the IEEE International Conference on INnovations in Intelligent SysTems and Applications, Gdynia, Poland, 3–5 July 2017; pp. 96–101.
63. Guo, J.; Tang, S. An Improved Particle Swarm Optimization with Re-initialization Mechanism. In Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 26–27 August 2009; pp. 437–441.