Intelligent Automation And Soft Computing, 2019

Vol.25,no.1,91–103

An accelerated convergent particle swarm optimizer (ACPSO) of multimodal functions

Yasir Mehmood and Waseem Shahzad

Department of Computer Science, National University of Computer and Emerging Science, Islamabad, Pakistan

ABSTRACT

Particle swarm optimization (PSO) algorithm is a global optimization technique that is used to find the optimal solution in multimodal problems. However, one of the limitation of PSO is its slow convergence rate along with a local trapping dilemma in complex multimodal problems. To address this issue, this paper provides an alternative technique known as ACPSO algorithm, which enables to adopt a new simplified velocity update rule to enhance the performance of PSO. As a result, the efficiency of convergence speed and solution accuracy can be maximized. The experimental results show that the ACPSO outperforms most of the compared PSO variants on a diverse set of problems.

KEY WORDS: Convergence speed, Global optimization, Multimodal, PSO, Premature convergence

1 INTRODUCTION

SWARM Intelligence (SI) techniques are population-based metaheuristics inspired by the social behavior of living and non-living organisms. SI algorithms are self-organized and the individuals of each algorithm share information without centralized control mechanism. Particle swarm optimization (James & Russell, 1995) and ant colony optimization (Dorigo & Di Caro, 1999) are widely used SI algorithms in various applications. Others recent SI algorithms are firefly algorithm (Yang, 2008), cuckoo search (Yang & Deb, 2009), earthworm optimization algorithm (Wang, Deb Coelho, 2015), krill herd (Wang et al., 2016), monarch butterfly optimization (Wang, Deb Cui, 2015), moth search (Wang, 2016), whale optimization algorithm (Mirjalili & Lewis, 2016), to name just a few.

The concept of PSO (Eberhart & Kennedy, 1995; James & Russell, 1995), we call it basic PSO (BPSO), inspired by social behavior of bird flocking and fish schooling, was put forth in 1995. It is a population-based optimization technique, which adopts leaders and shares information in order to guide the search for each particle in the swarm. Due to its simple concept and ease of implementation, BPSO has been applied to solve many optimization problems (Ali et al., 2012; Chu et al., 2016; Engelbrecht, 2006; Rauf & Aleisa, 2015; Sadhasivam & Thangaraj, 2017).

In spite of its simplicity, BPSO has several limitations which prevent it from achieving an efficient solutions (Bonyadi & Michalewicz, 2016). However, the two main limitations are its slow convergence rate and the local trapping dilemma. By slow convergence means the low rate of the fitness improvement of particles in the successive iterations. This is a very obliging feature of BPSO especially when the particles are near the optimal solution. However, on contrary to this, it happens that the most of the particles (except few particles) do not improve their fitness in the successive iterations. Therefore, these particles waste the computational resources and increase the computational cost (in terms of function evaluations or CPU time) (Hu et al., 2013; Zhan et al., 2011). Another drawback of

CONTACT Yasir Mehmood yasir.mehmood@nu.edu.pk

BPSO is the local trapping which means that all particles get stuck at some local optimum in the search space and no further improvement takes place. This is called premature convergence or stagnation (Van den Bergh & Engelbrecht, 2010). The convergence speed and premature convergence are the trade-off issues of BPSO. Because of the increase in convergence speed, the diversity rapidly lost which eventually leads to an undesirable premature convergence (Nakisa et al., 2014). In order to tackle this situation, researchers have tried to avoid the premature convergence by performing some extra computations (Adewumi & Arasomwan, 2016; Cui et al., 2009; Cui et al., 2012; Liang et al., 2006; Mendes et al., 2004; Tian, 2017) and have improved the convergence speed by introducing new parameters in BPSO (Eberhart & Shi, 2001; Eberhart, 1998; Park et al., 2010).

In order to enhance the performance of BPSO, for locating the global optima in complex multimodal problems, we have proposed an accelerated convergent PSO (ACPSO). In the projected model, the velocity-updating rule is modified by excluding the inertia part and altering the cognitive part of the equation. As a result, this reduces the need of memorizing the previous flying direction during evolution. Likewise, it does not incorporate any new parameter in BPSO and does not perform any extra computation while dealing with complex multimodal problems.

The advantages of ACPSO through comprehensive comparisons with eight state-of-the-art BPSO variants have been demonstrated. The empirical results reveal the fact that the ACPSO is faster, more accurate, consistent, reliable, and robust than the compared algorithms on 28 benchmark test functions (including unimodal, multimodal, and rotated) given in Table 1. It is also suggested that the proposed technique is the simplest among all the compared BPSO variants.

The rest of the paper is structured as follows: Section 2 describes the BPSO. Section 3 demonstrates the literature review. The ACPSO algorithm is introduced in Section 4. Section 5 discusses the experimental setup. The experimental results and discussion are summarized in Section 6. The paper is concluded in Section 7.

2 THE BASIC PSO (BPSO)

IN BPSO algorithm (James & Russell, 1995), the group of particles makes a swarm and each particle in a swarm represents a potential solution to a given optimization problem. Each particle i at iteration t is represented by a position vector X_t^i, the velocity vector V_t^i, and the personal best position vector $pbest_t^i$, . At the start, the position vectors are randomly initialized within the domain of a problem and the velocity vectors are initialized to zero (Engelbrecht, 2012; 2013). The velocity and position vectors are updated using Equation (1) and (2).

$$V_{(t+1)}^i = V_t^i + c1 \times rand1_t^i (pbest_t^i - X_t^i) + c2 \times rand2_t^i (gbest_t - X_t^i) \quad (1)$$

$$X_{(t+1)}^i = X_t^i + V_{(t+1)}^i \quad (2)$$

where $c1$ and $c2$ are cognitive and social acceleration coefficients and $rand1_t^i$, and $rand2_t^i$, are uniformly distributed random numbers between 0 and 1. The $gbest_t$ is the best position discovered by the whole swarm using Equation (4). The performance of each particle is evaluated using the fitness function f. During the evolution process, if the fitness of a particle is better than the fitness of its $pbest_t^i$ position, it will be updated using Equation (3).

$$pbest_t^i = \begin{cases} pbest_{(t-1)}^i & if\ f(X_t^i) \le f\left(pbest_{(t-1)}^i\right) \\ X_t^i & if\ f(X_t^i) > f\left(pbest_{t-1}^i\right) \end{cases} \quad (3)$$

$$gbest_t = \arg\min\ (pbest_t^i) \quad (4)$$

The velocity acts as a step size and is an aggregation of three components: the inertia, the cognitive component, and the social component as in Equation (5). The inertia performs the role of exploration while cognitive and social components perform the role of exploitation (Bonyadi et al., 2014).

$$V_{(t+1)}^i = \underbrace{\omega_i V_t^i}_{Inrtia} + \underbrace{c1\, rand_t^i \left(pbests_t^i - X_t^i\right)}_{Cognitive\ Component} + \underbrace{c2\, rand_t^i \left(gbest_t - X_t^i\right)}_{Social\ Component} \quad (5)$$

$$\omega_i = \frac{\omega_{fin} - \left(\omega_{fin} - \omega_{init}\right) \times iter}{\left(Max\_iter\right)} \tag{6}$$

where ω_i is introduced in (Eberhart, 1998) and its value is dynamically adjusted in every iteration using Equation (7). The $iter$ is the current iteration and $Max_{iter}$is the predefined maximum number of iterations. The ω_{fin} and ω_{init} are the final and the initial inertia weight values set to 0.9 and 0.4 respectively.

3 RELATED WORK

SINCE the design of BPSO, its many improved variants have been reported in the literature. The aim of these modifications was to improve the performance of BPSO either by introducing the additional parameters or by performing some extra computation.

Many researchers have introduced new parameters into BPSO to enhance the convergence speed. Shi and Eberhart (Eberhart, 1998) have proposed the linearly decreasing inertia weight ω in velocity update rule as in Equation (5). The proposed model achieved better performance than many other BPSO variants. In (Kennedy, 2002) suggested using the constriction factor χ into the velocity update Equation (7) and the value of χ is calculated using Equation (8). The proposed variant resolved he local trapping issue in unimodal problems.

$$V_{(t+1)}^{i} = \chi \left[\begin{array}{c} V_t^i + c1 \times rand1_t^i \left(pbest_t^i - X_t^i \right) + \\ c2 \times rand2_t^i \left(gbest_t - X_t^i \right) \end{array} \right] \tag{7}$$

$$\chi = \frac{2}{\left| 2 - \varnothing - \sqrt{\varnothing^2 - 4\varnothing} \right|}, \quad \varnothing = c1 + c2 \tag{8}$$

where the values of $c1$ and $c2$ are set to 2.05, which returns the value of χ = 0.719. The use of χ prevents each particle from exploring too far away from the search range and it eliminates the need of V_{max}. The (Eberhart & Shi, 2001) introduced a new variant of BPSO in which an inertia weight vector for each particle is randomly generated between 0.5 and 1.0. Researchers have also performed some extra computation to enhance the convergence rate (Qu et al., 2013; Zhan et al., 2009).

There is not only a slow convergence rate, but the local trapping dilemma is also a challenging task in BPSO. Many researchers have employed some extra computation to deal with this issue. In chaotic and crossover PSO (CCPSO) (Park *et al.*, 2010), the chaotic sequences and the crossover operation are combined to deal with the premature convergence issue. In order to improve the global search ability, a chaotic inertia weight has been suggested that dynamically combined the chaotic sequences with linearly decreasing inertia weight. Moreover, the crossover operation performed to enhance the diversity of the population.

It is quite difficult to accelerate the convergence speed and to avoid the local trapping problems at the same time. Mendes *et. al.* proposed the fully informed particle swarm (FPSO) (Mendes *et al.*, 2004) to accelerate the convergence speed. Each particle updates its position by taking the mean of $pbest$ from its surrounding neighborhood. Although it is a simple concept, yet it requires additional computation and the premature convergence problem is still there. Another approach namely a fitness-distance-ratio PSO (FDRPSO) (Peram et al., 2003) was proposed to resolve the premature convergence problem**.** In this method, the particles are influenced by its closest neighbor having better fitness rather than its personal best and global best positions. This strategy improved the premature convergence with extra computation and at the expense of slow convergence rate. In (Beheshti & Shamsuddin, 2015), the authors combined the global and local topologies to update the position of particles. Furthermore, they generated two new particles in the successive iteration to avoid the local trapping dilemma. However, this strategy needs an additional number of function evaluations in each iteration. In comprehensive learning PSO (CLPSO) (Liang *et al.*, 2006), the comprehensive learning (CL) strategy discourages premature convergence problem by using personal best information of all other particles instead of global best position. As a result, the CLPSO enables to generate better quality solutions in multimodal problems. However, the limitation of this model is that it performs extra computation and as a result causes a slow convergence in unimodal problems.

Simultaneously alleviating this trade-off goal is really a challenging task for the researchers. There are many attempts have been made to

achieve these goals. However, concurrently achieving these goals in an efficient manner still need to address and it requires more robust and efficient algorithms.

4 ACCELERATED CONVERGENT PSO (ACPSO)

IN literature, there are numerous global BPSO variants have been reported to accelerate the convergence speed and to reduce the premature convergence. Nevertheless, accelerating the convergence speed while avoiding the premature convergence problem has not been addressed well. In order to address this issue, we have proposed a new learning strategy that accelerates the convergence speed while avoiding the premature convergence using Equation (9).

$$V_{(t+1)}^{i} = \left(\omega_i \times c1 \times rand1_t^i \times pbest_t^i - X_t^i\right) + \left(c2 \times rand2_t^i \times \left(gbest_t - X_t^i\right)\right) \quad (9)$$

where ω_i is a linearly decreasing weight proposed in (Eberhart, 1998). The values of acceleration coefficients $c1$ and $c2$ are usually selected by trial and error methods (Engelbrecht, 2013; Li, 2004; Parrott & Li, 2006) or according to the fitness values (Wang & Yang, 2016). We have empirically set both values to 0.5. The other parameters are same as in Equation (1).

This new velocity equation is different from the BPSO variants, as it does not have the inertia part of Equation (5). However, the linearly decreasing weight decreases the impact of the personal best experience of particles. Furthermore, it modifies the cognitive component by multi-plying ω_i along with $c1$ and $rand1_t^i$ by the $pbest_t^i$ position of a particle and then the current position of that particle is subtracted from it. The social part adopts the global version of BPSO. The position update equation is same as Equation (2).

The basic inspiration of ACPSO has been taken from (R.C. Eberhart, 1995) which suggests to use the velocity update equation without previous velocity term. The projected variant is much simpler but inappropriate for locating a global optimum from multimodal problems. Hence, by utilizing this simpler variant, we have proposed a new velocity update equation that accelerates the convergence speed while reducing the local trapping issue in complex multimodal problems.

4.1 Managing the premature convergence problem

The risky property of BPSO is that the current position of a particle can be same as its $pbest$ and $gbest$ positions as in Equation (10). In this case, the cognitive component and the social component become zero as in Equation (11) and (12). Therefore, the acceleration coefficients and the random parameters also become useless and the movement of particles will solely depend on inertia part. If the inertia part is very close to zero, all the particles will stop moving. Meanwhile, if the $gbest$ particle caught by some local optimum, it will misguide the whole swarm and cause the premature convergence (van den Bergh & Engelbrecht, 2002).

$$x_t^i = pbest_t^i = gbest_t \quad (10)$$

$$\left(gbest_t - X_t^i\right) = 0 \quad (11)$$

$$\left(pbest_t^i - X_t^i\right) = 0 \quad (12)$$

In the proposed learning model, Equation (9), if the current position of a particle, its $pbest$ position, and the $gbest$ position become same, the cognitive component will produce non-zero result. Hence, this non-zero value will produce non-zero velocity with non-zero probability. Therefore, the modified model still can bring a change in the position of particles even if the current position, it's $pbest$ and the $gbest$ positions are same at any iteration. In ACPSO, we did not perform any additional operation neither to detect premature convergence nor to avoid from it.

Moreover, in the proposed learning model, particles will take long jumps at initial iterations, as particles are far from the targets and there will be larger position differences, and they gradually converge towards the optimum location in the latter iterations. In earlier iterations, the long jumps will explore the search space by maintaining the diversity, which avoids the algorithm to be trapped at some local optimum. However, in

later iterations, the particles will be exploiting around the promising solutions.

4.2 Accelerating the convergence speed

The fitness improvement rate of particles is usually very high at the initial stage of evolution and it gradually declines as evolution progress. This monotonic decreasing property is essential which prevents the particles to oscillate around the optimum location of a problem. However, during the convergence process, only a few number of particles may improve their fitness or the fitness improvement rate is very low in the successive iterations. This is because of poor exploration ability of particles that prevent them from moving towards the promising region in the search space. The convergence speed may improve if we let the particles move towards better location during iterations. This strategy reduces the unnecessary function evaluations, which ultimately improves the convergence speed.

In the proposed learning model, there is a larger momentum at initial iterations and it enthusiastically converges the swarm towards the global best location of a problem. The cognitive component plays the main role of exploration that brings larger changes at initial iterations while the social part exploits toward the optimum location. These combined explorative and exploitative behaviours of ACPSO make sure that the particles move toward the promising region during the convergence process.

The proposed learning strategy ensures that the current position never becomes same as its $pbest$ and $gbest$ positions and continuously moves the particles towards the better position and never freezes even for a single iteration. This fitness improvement procedure enhances the convergence speed and ensuring that the cognitive component never approaches to zero, which improves the explorations and avoids the trapping issue.

5 EXPERIMENTAL SETUP

IN this section, the benchmark test functions, contender algorithms and the performance metrics are presented.

5.1 Benchmark test functions

To validate the performance of the proposed approach, we have selected 28 benchmark test functions listed in Table 1. The test set includes nine unimodal functions, eleven multimodal functions, and eight rotated (rotated using Salomon's method (Salomon, 1996)) non-separable unimodal and multimodal functions with varying characteristics such as scalability, reparability, continuity, differentiability, and modality (Jamil & Yang, 2013; Suganthan et al., 2005). By definition, all the test functions are minimization functions and have an optimal value at the origin.

5.2 An experimental setting

In this study, the performance of proposed technique is compared with the following eight state-of-the-art BPSO variants on 10 and 30-*D* test functions. The population size of 20 is selected for all experiments and the other parameters of these algorithms are remained same as used in the original papers. For a fair comparison, all algorithms are executed 30 times and the mean results are reported.

- LWPSO: PSO with linear inertia weight
- RWPSO: PSO with random inertia weight
- CFPSO: PSO with constriction factor
- FDRPSO: Fitness-Distance-Ratio based PSO
- FPSO: Fully Informed PSO
- CLPSO: Comprehensive Learning PSO
- NPSO: Novel PSO
- CCPSO: PSO with Chaotic sequences and Crossover operation

Currently, two types of experiments have been conducted to evaluate the different measures mentioned in sub-section 4.3. In the first experiment, all algorithms are run for a fixed number of function evaluations (Max_{FEs}, i.e. 10000*D, where D is the dimension of a problem) and the best fitness values is recorded. In the second experiment, the execution of each algorithm is terminated when the error threshold $\boldsymbol{\varepsilon}$ (1.00e-5) had reached or in the worst case, Max_{FEs} had executed and the number of function evaluations FEs were recorded.

5.3 Performance metrics

The experimental results are verified in terms of (a) - The *convergence accuracy*, which shows the closeness of the best solution found by an algorithm with the actually known value of a problem using Equation (13). (b) - The *convergence speed* determines how long an

Table 1. Summary of Benchmark Test Functions

Name	Test Function	Properties	Domain	Optimum
Sphere	$F_1(X) = \sum_{d=1}^{D} x_d^2$	S,C,D,U	$[\pm 150]^D$	$f(\vec{x}) = 0$
Quartic	$F_2(X) = \sum_{d=1}^{D} d\, x_d^4$	S,C,D,U	$[\pm 50]^D$	$f(\vec{x}) = 0$
Quartic with noise	$F_3(X) = \sum_{d=1}^{D} d\, x_d^4 + rand[0,1]$	S,C,D,U	$[\pm 50]^D$	$f(\vec{x}) = 0$
Elliptic	$F_4(X) = \sum_{d=1}^{D} (10^6)^{\frac{d-1}{D-1}}\, {x_d}^2$	S,C,D,U	$[\pm 0.5]^D$	$f(\vec{x}) = 0$
Step	$F_5(X) = \sum_{d=1}^{D} (\lfloor x_d + 0.5 \rfloor)^2$	S,Cn,Dn,U	$[\pm 100]^D$	$f(\vec{x}) = 0$
Schwefel 1.2	$F_6(X) = \sum_{d=1}^{D} \left(\sum_{i=1}^{d} x_i\right)^2$	Sn,C,D,U	$[\pm 100]^D$	$f(\vec{x}) = 0$
Schwefel 2.22	$F_7(X) = \sum_{d=1}^{D} \lvert x_d \rvert + \prod_{d=1}^{D} \lvert x_d \rvert$	Sn,C,Dn,U	$[\pm 100]^D$	$f(\vec{x}) = 0$
Bent Cigar	$F_8(X) = {x_d}^2 + 10^6 \sum_{d=2}^{D} {x_d}^2$	Sn,C,D,U	$[\pm 100]^D$	$f(\vec{x}) = 0$
Schaffer's F7	$F_9(X) = \left(\sum_{d=1}^{D} x_d^2\right)^{1/4} \left[\sin^2\left(50\left(\sum_{d=1}^{D} x_d^2\right)^{1/10}\right) + 1.0\right]$.	Sn,C,D,U	$[\pm 100]^D$	$f(\vec{x}) = 0$
Csendes	$F_{10}(X) = \sum_{d=1}^{d} x_d^6 \left(2 + \sin\frac{1}{x_d}\right)$	S,C,D,M	$[\pm 1]^D$	$f(\vec{x}) = 0$
Rastrigin	$F_{11}(X) = \sum_{d=1}^{D} (x_d^2 - 10\cos(2\pi x_d) + 10)$	S,C,D,M	$[\pm 50]^D$	$f(\vec{x}) = 0$
Weierstrass	$F_{12}(X) = \sum_{d=1}^{D} \left(\sum_{k=0}^{20} \left[0.5^k \cos\left(2\pi 3^k (x_d + 0.5)\right)\right]\right) - D \sum_{k=0}^{20} [0.5^k \cos(2\pi 3^k * 0.5)]$	S,C,D,M	$[\pm 0.5]^D$	$f(\vec{x}) = 0$
Alpine	$F_{13}(X) = \sum_{d=1}^{D} \lvert x_d sin(x_d) + 0.1\, x_d \rvert$	S,C,Dn,M	$[\pm 10]^D$	$f(\vec{x}) = 0$
Deb 1	$F_{14}(X) = \frac{1}{D} \sum_{d=1}^{D} sin^6(5\pi x_d)$	S,C,D,M	$[\pm 1]^D$	$f(\vec{x}) = 0$
Non_continued Rastrigin	$F_{15}(X) = \sum_{d=1}^{D} (y_d - 10\cos(2\pi y_d) + 10)$, $y_d = f(x) = \begin{cases} x_d, & \lvert x_d \rvert < \frac{1}{2} \\ \frac{round\ (2\, x_d)}{2}, & \lvert x_d \rvert \geq \frac{1}{2} \end{cases}$	S,Cn,Dn,M	$[\pm 50]^D$	$f(\vec{x}) = 0$
Salomon	$F_{16}(X) = 1 - cos\left(2\pi\sqrt{\sum_{d=1}^{D} x_d^2}\right) + 0.1\sqrt{\sum_{d=1}^{D} x_d^2}$	Sn,C,D,M	$[\pm 100]^D$	$f(\vec{x}) = 0$
Ackley	$F_{17}(X) = -20 \exp\left(-0.2\sqrt{\frac{1}{D}\sum_{d=1}^{D} x_d^2}\right) - \exp\left(\frac{1}{D}\sum_{d=1}^{D} cos(2\pi x_d)\right) + 20 + e$	Sn,C,D,M	$[\pm 50]^D$	$f(\vec{x}) = 0$
Griewank	$F_{18}(X) = \sum_{d=1}^{D} \frac{x_d^2}{4000} - \prod_{d=1}^{D} cos\left(\frac{x_d}{\sqrt{d}}\right) + 1$	Sn,C,D,M	$[\pm 500]^D$	$f(\vec{x}) = 0$
Mishra 11	$F_{19}(X) = \left[\frac{1}{D}\sum_{d=1}^{D} \lvert x_d \rvert - \left(\prod_{d=1}^{D} \lvert x_d \rvert\right)^{\frac{1}{D}}\right]^2$	Sn,C,D,M	$[\pm 10]^D$	$f(\vec{x}) = 0$
Rot_Elliptic	$F_{22}(X) = F_4(y)\ ,\ y = x * M$	Sn,C,D,U	$[\pm 0.5]^D$	$f(\vec{x}) = 0$
Rot_Sphere	$F_{21}(X) = F_1(y)\ ,\ y = x * M$	Sn,C,D,U	$[\pm 150]^D$	$f(\vec{x}) = 0$
Rot_Elliptic	$F_{22}(X) = F_4(y)\ ,\ y = x * M$	Sn,C,D,U	$[\pm 0.5]^D$	$f(\vec{x}) = 0$
Rot_Schwefel 1.2	$F_{23}(X) = F_6(y)\ ,\ y = x * M$	Sn,C,Dn,U	$[\pm 100]^D$	$f(\vec{x}) = 0$
Rot_Rastrigin	$F_{24}(X) = F_{11}(y)\ ,\ y = x * M$	Sn,C,D,M	$[\pm 5.12]^D$	$f(\vec{x}) = 0$
Rot_Wierstras	$F_{25}(X) = F_{12}(y)\ ,\ y = x * M$	Sn,C,D,M	$[\pm 0.5]^D$	$f(\vec{x}) = 0$
Rot_Noncon Rast	$F_{26}(X) = F_{15}(z)\ ,\ z = y * M$	Sn,Cn,D,M	$[\pm 50]^D$	$f(\vec{x}) = 0$
Rot_Ackley	$F_{27}(X) = F_{17}(y)\ ,\ y = x * M$	Sn,C,D,M	$[\pm 50]^D$	$f(\vec{x}) = 0$
Rot_Griwank	$F_{28}(X) = F_{18}(y)\ ,\ y = x * M$	Sn,C,D,M	$[\pm 500]^D$	$f(\vec{x}) = 0$

*S=Separable, Sn=Non- Separable, C=Continue, Cn=Non-continue, D=differentiable, Dn=Non- differentiable, U =Unimodal, M=Multimodal

algorithm will take to reach an acceptable accuracy level ε and the mean number of function evaluations are recorded. (c) - The *success performance (SP)* determines the number of *FEs* of successful runs to solve a problem within the pre-specified accuracy level ε using Equation (14). (d) - The *success rate (SR)* is a ratio of numbers of successfully runs over the total number of runs using Equation (15).

In addition, the (e) - *t*-test compares the performance statistically. Two-tailed t-test with 58 degrees of freedom at a 0.05 level of significance was conducted between the ACPSO and the BPSO variants. Two algorithms are statistically different if the value of *t*-test is greater than 2.0017 (Hu *et al.*, 2013; Suganthan *et al.*, 2005).

$$Accuracy = \left| f(\hat{x}) - f(\vec{x}) \right| \quad (13)$$

$$SP = \frac{mean(FEs\ of\ successful\ runs) \times (Total\ number\ of\ runs)}{(Number\ of\ successful\ runs)} \quad (14)$$

$$SR = \frac{(Number\ of\ successful\ runs)}{(Total\ number\ of\ runs)} \times 100 \quad (15)$$

6 EXPERIMENTAL RESULTS AND DISCUSSIONS

EXPERIMENTS have been conducted to find the global optimum of 28 test functions listed in Table 1. The results demonstrate the closer picture of how much the ACPSO algorithm outperforms its contenders. The effectiveness of the results on the base of different metrics (described in sections 5.3) is presented below.

6.1 Results on convergence accuracy

The convergence accuracy of ACPSO and other BPSO variants is summarized in Table 2 and Table 3 for 10-*D* and 30-*D* test functions respectively. The results demonstrate that the ACPSO has outperformed its contenders in the majority of the test functions.

6.1.1 Convergence accuracy of 10-*D* test functions

The first column of each test function in Table 2 presents the mean accuracy of 10-*D* test functions. The results demonstrated that the ACPSO has higher accuracy than the majority of other BPSO variants. The FPSO has the same results as ACPSO for 13 functions that are F5, F10, F11, F12, F14, F15, F17, F18, F24, F25, F26, F27, and F28. In addition, the FPSO has competitive results with the ACPSO in 10 functions: F1, F2, F4, F6, F8, F16, F20, F21, F22, and F23. So overall, the ACPSO has better accuracy than the FPSO for five functions only. The CFPSO also achieved global optimum for F5 and F11 test functions. In noisy test function (F3), the ACPSO could not converge to the global optimum, but still has better accuracy than other BPSO variants. Except for ACPSO, other BPSO variants experience different levels of performance degradation on rotated test functions.

The presented results are obtained using Equation (13), where $f(\hat{x})$ is the mean fitness found by the algorithm and $f(\vec{x})$ is a known global optimum. The proposed learning model has larger momentum at initial iterations, which enables the particles to explore the search space thoroughly. This exploration capability ensures that the particles cannot trap at a local optimum.

6.1.2 Convergence accuracy of 30-*D* test functions

The first column of each test function in Table 3 presents the mean accuracy of 30-*D* test functions. The F5 function is a step function, as the dimensions of test function F5 increases from 10 to 30, four (ACPSO, FPSO, CFPSO, and FDRPSO) out of nine BPSO variants can converge to the global optimum. In this experiment, the FPSO has the same results of the proposed technique and it has successfully found the global optimum for 13 test functions (F5, F10, F11, F12, F14, F15, F17, F18, F24, F25, F26, F27, and F28), whereas FPSO achieved the near to optimum solution for 10 test functions (F1, F2, F4, F6, F8, F16, F20, F21, F22, and F23). The proposed technique outperforms the FPSO for five test functions only.

The results show that the performance of ACPSO on 30-*D* test functions is better than the results of the 10-*D* test functions. This is because, in the proposed technique, each dimension is updated independently. At initial iterations, there is a larger momentum in each dimension of a particle, as the dimensions increase, the particles move more enthusiastically. This exploration strategy, as the dimensions increase, avoids the local trapping issue and improves the convergence accuracy.

The comprehensive results are presented in Table 4. In the first row, the w/t/l presents the number of problems for which ACPSO wins, equal, or loss in terms of solution accuracy on 10 and 30-*D* test functions. The results reveal that the ACPSO significantly performs better than the other BPSO variants. However, LWPSO, RWPSO, FDRPSO, CLPSO, and NPSO can achieve equal results for only one function and CFPSO can achieve better results for two functions in case of 10-*D* test functions. The FPSO is the only competitor, which has equal results of 13 test functions for both 10 and 30-*D* test functions.

Table 2. Results of 10-dimensional Test Functions

	Mean	FEs	SP	SR	Mean	FEs	SP	SR	Mean	FEs	SP	SR	Mean	FEs	SP	SR
		F1				F2				F3				F4		
LWPSO	1.7E-20	1.1E+05	1.1E+05	**100**	9.5E-33	1.0E+05	1.0E+05	**100**	6.5E-03	2.0E+05	Inf	0	1.3E-23	9.6E+04	9.6E+04	**100**
RWPSO	2.5E-01	1.9E+05	1.6E+06	10	1.1E-02	1.7E+05	1.9E+05	80	1.9E-01	2.0E+05	Inf	0	3.1E-04	1.4E+05	1.4E+05	**100**
CFPSO	4.6E-48	1.7E+04	1.7E+04	**100**	2.4E-86	1.2E+04	1.2E+04	**100**	8.9E-04	2.0E+05	Inf	0	1.1E-51	1.2E+04	1.2E+04	**100**
FDRPSO	4.2E-54	1.7E+04	1.7E+04	**100**	9.3E-94	1.1E+04	1.1E+04	**100**	2.8E-03	2.0E+05	Inf	0	3.9E-01	1.4E+06	3.6E+05	70
FPSO	4.9E-134	6.1E+03	6.1E+03	**100**	4.5E-269	3.7E+03	3.7E+03	**100**	7.9E-05	**1.5E+05**	**1.8E+05**	**40**	1.2E-134	5.3E+03	5.3E+03	**100**
CLPSO	7.6E-08	7.1E+04	7.1E+04	**100**	3.9E-12	5.7E+04	5.7E+04	**100**	2.2E-02	2.0E+05	Inf	0	2.4E-10	5.5E+04	5.5E+04	**100**
NPSO	1.4E-33	2.1E+04	2.1E+04	**100**	3.5E-59	1.4E+04	1.4E+04	**100**	3.3E-03	2.0E+05	Inf	0	7.7E-15	2.8E+04	2.8E+04	**100**
CCPSO	2.3E+04	2.0E+05	Inf	0	9.1E+06	2.0E+05	Inf	0	6.3E+06	2.0E+05	Inf	0	6.2E+01	2.0E+05	Inf	0
ACPSO	**0**	**6.0E+02**	**6.0E+02**	**100**	**0**	**4.0E+02**	**4.0E+02**	**100**	**1.1E-05**	1.8E+05	4.1E+05	27	**0**	**4.6E+02**	**4.6E+02**	**100**
		F5				F6				F7				F8		
LWPSO	**0**	7.6E+04	7.6E+04	**100**	2.7E-19	1.2E+05	1.2E+05	**100**	1.7E-11	1.3E+05	1.3E+05	**100**	9.9E-18	1.2E+05	1.2E+05	**100**
RWPSO	**0**	1.1E+04	1.1E+04	**100**	2.6E+00	2.0E+05	Inf	0	7.0E-01	2.0E+05	Inf	0	8.3E+02	2.0E+05	Inf	0
CFPSO	**0**	2.7E+03	2.7E+03	**100**	4.6E-48	1.9E+04	1.9E+04	**100**	1.0E-25	2.7E+04	2.7E+04	**100**	4.9E-45	2.8E+04	2.8E+04	**100**
FDRPSO	**0**	2.3E+03	2.3E+03	**100**	4.4E-55	4.0E+05	4.0E+05	**100**	2.2E+00	1.0E+05	9.8E+04	80	5.7E-53	8.1E+02	8.1E+02	**100**
FPSO	**0**	7.4E+02	7.4E+02	**100**	5.5E-133	6.9E+03	6.9E+03	**100**	3.3E-67	1.0E+04	1.0E+04	**100**	1.9E-130	8.7E+03	8.7E+03	**100**
CLPSO	**0**	1.9E+04	1.9E+04	**100**	6.6E-07	7.7E+04	7.7E+04	**100**	9.2E-05	9.7E+04	9.7E+04	**100**	2.5E-04	8.9E+04	8.9E+04	**100**
NPSO	**0**	2.5E+03	2.5E+03	**100**	6.3E-31	2.5E+04	2.5E+04	**100**	9.2E-05	3.9E+04	3.9E+04	**100**	9.2E+03	1.9E+05	2.2E+06	5
CCPSO	1.1E+04	2.0E+05	Inf	0	2.7E+05	2.0E+05	Inf	0	1.2E+03	2.0E+05	Inf	0	7.9E+07	2.0E+05	Inf	0
ACPSO	**0**	**2.9E+02**	**2.9E+02**	**100**	**0**	**6.7E+02**	**6.8E+02**	**100**	**0**	**9.1E+02**	**9.1E+02**	**100**	**0**	**8.1E+02**	**8.1E+02**	**100**
		F9				**F10**				F11				F12		
LWPSO	3.6E-01	2.0E+05	Inf	0	1.1E-58	3.3E+04	3.3E+04	**100**	5.9E+00	2.0E+05	Inf	0	4.1E-04	1.4E+05	1.5E+05	95
RWPSO	2.2E+00	2.0E+05	Inf	0	2.7E-14	5.7E+03	5.7E+03	**100**	2.5E+01	2.0E+05	Inf	0	2.7E-01	2.0E+05	Inf	0
CFPSO	3.7E-03	1.7E+05	1.9E+05	80	1.7E-140	2.4E+03	2.4E+03	**100**	**0**	3.5E+04	3.5E+04	**100**	5.5E-01	1.2E+05	8.4E+04	53
FDRPSO	3.8E-02	2.0E+05	Inf	0	1.4E-159	1.8E+03	1.8E+03	**100**	7.6E+00	1.9E+05	5.9E+05	10	1.0E-01	2.4E+06	1.5E+06	50
FPSO	3.7E-34	1.6E+04	1.6E+04	**100**	**0**	9.9E+02	9.9E+02	**100**	**0**	7.1E+03	7.1E+03	**100**	**0**	1.4E+04	1.4E+04	**100**
CLPSO	1.7E+00	2.0E+05	Inf	0	8.9E-22	1.7E+04	1.7E+04	**100**	7.3E+00	2.0E+05	Inf	0	1.2E-02	1.4E+05	1.7E+05	65
NPSO	1.7E+00	2.0E+05	Inf	0	1.3E-12	1.9E+03	1.9E+03	**100**	6.5E+00	2.0E+05	Inf	0	3.3E-01	1.7E+05	2.4E+05	25
CCPSO	1.2E+01	2.0E+05	Inf	0	1.4E-01	2.0E+05	Inf	0	2.8E+03	2.0E+05	Inf	0	1.0E+01	2.0E+05	Inf	0
ACPSO	**0**	**1.4E+03**	**1.4E+03**	**100**	**0**	**1.0E+02**	**1.0E+02**	**100**	**0**	**7.0E+02**	**7.0E+02**	**100**	**0**	**1.2E+03**	**1.2E+03**	**100**
		F13				F14				F15				F16		
LWPSO	4.4E-06	1.3E+05	1.3E+05	**100**	2.8E-05	1.2E+05	1.2E+05	100	6.4E+00	2.0E+05	Inf	0	9.9E-02	4.6E+03	4.6E+03	**100**
RWPSO	5.0E-02	2.0E+05	Inf	0	2.9E-04	1.7E+05	3.2E+05	40	3.2E+01	2.0E+05	Inf	0	2.7E-01	2.0E+05	Inf	0
CFPSO	8.9E-20	2.9E+04	2.9E+04	**100**	2.3E-03	2.0E+05	Inf	0	1.9E+00	1.1E+05	1.1E+05	80	1.2E-03	2.0E+05	Inf	0
FDRPSO	7.4E-16	6.0E+05	6.0E+05	**100**	8.9E-19	8.9E+05	6.0E+05	90	3.4E+00	1.8E+05	5.8E+05	20	9.9E-02	2.0E+05	Inf	0
FPSO	3.6E-69	7.1E+03	7.1E+03	**100**	**0**	2.4E+03	2.4E+03	**100**	**0**	7.1E+03	7.1E+03	**100**	2.9E-135	4.9E+03	4.9E+03	**100**
CLPSO	4.1E-05	9.0E+04	9.0E+04	**100**	5.7E-09	7.9E+04	7.9E+04	**100**	9.4E+00	2.0E+05	Inf	0	9.9E-02	2.0E+05	Inf	0
NPSO	3.4E-01	2.0E+05	Inf	0	3.7E-04	1.1E+05	1.1E+05	65	4.6E+00	1.9E+05	1.0E+06	15	9.9E-02	2.0E+05	Inf	0
CCPSO	1.0E+01	2.0E+05	Inf	0	1.5E-02	2.0E+05	Inf	0	2.5E+03	2.0E+05	Inf	0	1.0E+03	2.0E+05	Inf	0
ACPSO	**0**	**7.0E+02**	**7.0E+02**	**100**	**0**	**1.3E+03**	**1.3E+03**	**100**	**0**	**6.9E+02**	**6.9E+02**	**100**	**0**	**5.2 E+02**	**5.2 E+02**	**100**
		F17				F18				F19				F20		
LWPSO	2.1E-11	1.3E+05	1.3E+05	**100**	1.1E-01	2.0E+05	Inf	0	2.1E-12	4.2E+04	4.2E+04	**100**	1.3E-23	9.8E+04	9.8E+04	**100**
RWPSO	8.4E-01	2.0E+05	Inf	0	4.8E-01	2.0E+05	Inf	0	1.8E-10	2.9E+04	2.9E+04	**100**	7.9E-04	1.8E+05	2.2E+05	80
CFPSO	3.0E-15	6.3E+04	3.6E+04	80	4.4E-11	5.2E+04	5.2E+04	**100**	6.4E-07	6.5E+03	6.5E+03	**100**	5.4E-52	1.2E+04	1.2E+04	**100**
FDRPSO	2.0E+00	2.6E+04	2.6E+04	**100**	9.4E-02	2.0E+05	Inf	0	3.2E-06	6.0E+03	6.0E+03	**100**	1.7E-56	1.2E+04	1.2E+04	**100**
FPSO	8.9E-16	9.1E+03	9.1E+03	100	**0**	6.1E+03	6.1E+03	**100**	3.2E-07	3.5E+03	3.5E+03	**100**	1.1E-136	4.5E+03	4.5E+03	**100**
CLPSO	1.8E+00	1.2E+05	1.3E+05	80	4.8E-02	2.0E+05	Inf	0	5.8E-08	2.7E+04	2.7E+04	**100**	1.1E-10	5.7E+04	5.7E+04	**100**
NPSO	5.8E-02	4.2E+04	4.2E+04	**100**	7.8E-02	2.0E+05	Inf	0	1.7E-07	1.5E+04	1.5E+04	**100**	3.0E-36	1.6E+04	1.6E+04	**100**
CCPSO	2.0E+01	2.0E+05	Inf	0	8.9E+01	2.0E+05	Inf	0	1.9E-08	1.8E+04	1.8E+04	**100**	6.5E+09	2.0E+05	Inf	0
ACPSO	**8.9E-16**	**8.7 E+02**	**8.7 E+02**	**100**	**0**	**6.1 E+02**	**6.1 E+02**	**100**	**0**	**4.0 E+02**	**4.0 E+02**	**100**	**0**	**4.8 E+02**	**4.8 E+02**	**100**
		F21				F22				F23				F24		
LWPSO	1.3E-20	1.1E+05	1.1E+05	**100**	4.8E-01	2.0E+05	Inf	0	6.1E-01	1.9E+05	9.7E+05	20	2.5E+01	2.0E+05	Inf	0
RWPSO	1.0E-01	2.0E+05	Inf	0	2.1E+00	2.0E+05	Inf	0	8.5E+02	2.0E+05	Inf	0	5.6E+01	2.0E+05	Inf	0
CFPSO	2.5E-50	1.7E+04	1.7E+04	**100**	2.4E-01	1.3E+05	1.5E+05	70	1.6E-17	3.2E+04	3.2E+04	**100**	1.7E+01	1.8E+05	3.9E+05	23
FDRPSO	4.7E-55	1.9E+04	1.9E+04	**100**	5.2E+00	2.0E+05	Inf	0	2.4E-08	8.8E+04	8.3E+04	90	4.0E+01	2.0E+05	Inf	0
FPSO	4.2E-134	6.1E+03	6.1E+03	**100**	3.6E-135	5.2E+03	5.2E+03	**100**	3.9E-133	6.9E+03	6.9E+03	**100**	**0**	7.1E+03	7.1E+03	**100**
CLPSO	6.4E-09	7.3E+04	7.3E+04	**100**	3.8E+00	2.0E+05	Inf	0	8.1E+02	2.0E+05	Inf	0	2.3E+01	2.0E+05	Inf	0
NPSO	7.7E-33	2.3E+04	2.3E+04	**100**	1.5E+00	2.0E+05	Inf	0	2.0E+00	1.8E+05	5.9E+05	20	1.5E+01	2.0E+05	Inf	0
CCPSO	2.3E+04	2.0E+05	Inf	0	4.2E+01	2.0E+05	Inf	0	2.9E+05	2.0E+05	Inf	0	2.6E+03	2.0E+05	Inf	0
ACPSO	**0**	**9.6 E+02**	**9.6 E+02**	**100**	**0**	**5.0 E+02**	**5.0 E+02**	**100**	**0**	**9.7 E+02**	**9.7 E+02**	**100**	**0**	**7.1 E+02**	**7.1 E+02**	**100**
		F25				F26				F27				F28		
WPSO	1.6E+00	2.0E+05	Inf	0	2.3E+01	2.0E+05	Inf	0	4.0E-01	1.6E+05	2.0E+05	65	1.9E-01	2.0E+05	Inf	0
RWPSO	2.4E+00	2.0E+05	Inf	0	4.3E+01	2.0E+05	Inf	0	1.3E+00	2.0E+05	Inf	0	7.1E-01	2.0E+05	Inf	0
CFPSO	7.7E-01	1.2E+05	9.7E+04	50	1.3E+01	1.9E+05	1.8E+06	5	4.4E-15	3.1E+04	2.6E+04	97	3.0E-01	1.4E+05	1.1E+05	35
FDRPSO	3.2E+00	2.0E+05	Inf	0	1.7E+01	2.0E+05	Inf	0	2.3E-01	1.2E+06	6.9E+05	80	3.7E-01	2.0E+05	Inf	0
FPSO	**0**	1.4E+04	1.4E+04	**100**	**0**	7.1E+03	7.1E+03	**100**	**8.9E-16**	9.1E+03	9.1E+03	**100**	**0**	6.1E+03	6.1E+03	**100**
CLPSO	4.3E+00	2.0E+05	Inf	0	**2.1E+01**	2.0E+05	Inf	0	2.5E+00	1.3E+05	1.3E+05	80	1.0E-01	1.9E+05	1.9E+06	5
NPSO	2.5E+00	2.0E+05	Inf	0	1.8E+01	2.0E+05	Inf	0	2.7E-01	7.6E+04	4.6E+04	75	1.1E-01	2.0E+05	Inf	0
CCPSO	1.0E+01	2.0E+05	Inf	0	2.6E+03	2.0E+05	Inf	0	2.0E+01	2.0E+05	Inf	0	9.5E+01	2.0E+05	Inf	0
ACPSO	**0**	**1.2E+03**	**1.2E+03**	**100**	**0**	**7.2 E+02**	**7.2 E+02**	**100**	**8.9E-16**	**8.5 E+02**	**8.5 E+02**	**100**	**0**	**6.1 E+02**	**6.1 E+02**	**100**

Table 3. Results of 30-dimensional Test Functions

	Mean	FEs	SP	SR	Mean	FEs	SP	SR	Mean	FEs	SP	SR	Mean	FEs	SP	SR
		F1				F2				F3				F4		
LWPSO	2.8E-02	5.5E+05	5.5E+05	**100**	8.7E-01	5.5E+05	5.5E+05	**100**	6.8E-01	6.0E+05	Inf	0	2.7E-05	4.9E+05	4.9E+05	**100**
RWPSO	3.6E+03	6.0E+05	Inf	0	5.9E+05	6.0E+05	Inf	0	6.6E+05	6.0E+05	Inf	0	3.7E+00	6.0E+05	Inf	0
CFPSO	8.2E-16	1.4E+05	1.4E+05	**100**	2.4E-20	1.3E+05	1.3E+05	**100**	6.7E-03	6.0E+05	Inf	0	5.3E-18	1.1E+05	1.1E+05	**100**
FDRPSO	1.1E-09	2.4E+05	2.4E+05	**100**	5.1E-13	2.5E+05	2.3E+05	90	6.2E+05	6.0E+05	Inf	0	3.5E+01	6.0E+05	Inf	0
FPSO	2.8E-133	2.0E+04	2.0E+04	**100**	1.1E-267	1.1E+04	1.1E+04	**100**	6.9E-05	**4.8E+05**	**6.9E+05**	**55**	9.0E-135	1.8E+04	1.8E+04	**100**
CLPSO	1.9E+01	6.0E+05	Inf	0	9.1E+02	6.0E+05	Inf	0	3.7E+02	6.0E+05	Inf	0	5.6E-02	5.8E+05f	3.9E+06	10
NPSO	2.5E-08	2.1E+05	2.1E+05	**100**	4.1E-09	1.9E+05	1.9E+05	100	6.4E-02	6.0E+05	Inf	0	1.5E+00	5.6E+05	2.1E+06	20
CCPSO	1.1E+05	6.0E+05	Inf	0	1.8E+08	6.0E+05	Inf	0	1.7E+08	6.0E+05	Inf	0	6.1E+02	6.0E+05	Inf	0
ACPSO	**0**	**2.1 E+03**	**2.1 E+03**	**100**	**0**	**1.5 E+03**	**1.5 E+03**	**100**	**1.9E-05**	5.3E+05	1.2E+06	27	**0**	**1.6 E+03**	**1.6 E+03**	**100**
		F5				F6				F7				F8		
LWPSO	2.3E+00	4.4E+05	4.4E+05	**100**	2.2E+00	5.9E+05	7.8E+05	75	1.3E+02	6.0E+05	1.2E+07	5	1.3E+02	5.9E+05	3.9E+06	0
RWPSO	1.1E+03	6.0E+05	Inf	0	1.8E+05	6.0E+05	6.0E+05	**100**	2.8E+18	6.0E+05	Inf	0	1.3E+07	6.0E+05	Inf	0
CFPSO	**0**	3.2E+04	3.2E+04	**100**	2.4E-14	1.7E+05	1.7E+05	**100**	2.4E-10	1.9E+05	1.9E+05	**100**	1.8E-11	2.8E+05	2.8E+05	**100**
FDRPSO	**0**	5.5E+04	5.5E+04	**100**	2.8E+04	1.0E+07	1.8E+07	30	3.3E+02	6.0E+05	Inf	0	5.0E+03	2.7E+03	**2.7E+03**	**100**
FPSO	**0**	2.4E+03	2.4E+03	**100**	3.2E-131	2.4E+04	2.4E+04	100	1.5E-66	3.1E+04	3.1E+04	**100**	1.2E-129	2.7E+04	2.7E+04	**100**
CLPSO	4.4E+01	1.3E+05	1.3E+05	**100**	1.1E+03	6.0E+05	Inf	0	3.0E+01	6.0E+05	Inf	0	8.4E+04	6.0E+05	Inf	0
NPSO	1.0E-01	1.6E+05	1.6E+05	**100**	4.3E+03	6.0E+05	Inf	0	3.0E+01	4.8E+05	6.1E+05	70	4.5E+05	6.0E+05	Inf	0
CCPSO	5.0E+04	6.0E+05	Inf	0	1.3E+07	6.0E+05	Inf	0	2.2E+28	6.0E+05	Inf	0	4.9E+08	6.0E+05	Inf	0
ACPSO	**0**	**9.0 E+02**	**9.0 E+02**	**100**	**0**	**2.4 E+03**	**2.4 E+03**	**100**	**0**	**3.2E+03**	**3.2E+03**	**100**	**0**	**2.7E+03**	**2.7E+03**	**100**
		F9				F10				F11				F12		
LWPSO	6.2E+00	6.0E+05	Inf	0	8.7E-10	3.6E+05	3.6E+05	**100**	8.9E+01	6.0E+05	Inf	0	8.2E-01	6.0E+05	Inf	0
RWPSO	9.8E+00	6.0E+05	Inf	0	6.7E-04	6.0E+05	Inf	0	7.9E+02	6.0E+05	Inf	0	1.5E+01	6.0E+05	Inf	0
CFPSO	2.1E-01	6.0E+05	Inf	0	2.0E-36	3.9E+04	3.9E+04	**100**	1.8E-01	2.7E+05	2.7E+05	**100**	5.2E+00	6.0E+05	Inf	0
FDRPSO	3.4E+00	6.0E+05	Inf	0	7.4E-09	7.1E+04	7.1E+04	**100**	6.2E+01	6.0E+05	Inf	0	1.3E+00	6.0E+05	Inf	0
FPSO	5.7E-34	4.8E+04	4.8E+04	**100**	0	4.1E+03	4.1E+03	**100**	0	2.3E+04	2.3E+04	**100**	0	4.3E+04	4.3E+04	**100**
CLPSO	8.7E+00	6.0E+05	Inf	0	6.9E-07	1.6E+05	1.6E+05	**100**	1.1E+02	6.0E+05	Inf	0	3.3E+00	6.0E+05	Inf	0
NPSO	8.7E+00	6.0E+05	Inf	0	8.8E-07	8.9E+04	8.9E+04	**100**	1.0E+02	6.0E+05	Inf	0	6.7E+00	6.0E+05	Inf	0
CCPSO	1.7E+01	6.0E+05	Inf	0	1.9E+00	6.0E+05	Inf	0	1.2E+04	6.0E+05	Inf	0	4.2E+01	6.0E+05	Inf	0
ACPSO	**0**	**4.7E+03**	**4.7E+03**	**100**	**0**	**5.7 E+02**	**5.7 E+02**	**100**	**0**	**2.4 E+03**	**2.4 E+03**	**100**	**0**	**3.9 E+03**	**3.9 E+03**	**100**
		F13				F14				F15				F16		
LWPSO	6.2E-02	5.9E+05	1.3E+06	45	3.8E-03	5.9E+05	1.4 E+06	40	1.2E+02	6.0E+05	Inf	0	6.2E-01	2.6E+03	2.6E+03	**100**
RWPSO	1.4E+01	6.0E+05	6.0E+05	**100**	4.3E-02	6.0E+05	6.0E+05	**100**	6.6E+02	6.0E+05	Inf	0	1.2E+02	6.0E+05	Inf	0
CFPSO	5.4E-09	1.9E+05	1.9E+05	**100**	6.9E-02	6.0E+05	6.0E+05	**100**	3.9E+01	4.0E+05	4.4E+05	85	9.9E-02	6.0E+05	Inf	0
FDRPSO	5.1E-06	6.1E+06	6.1E+06	**100**	8.0E-04	4.0E+06	2.5E+06	80	8.4E+01	6.0E+05	Inf	0	2.0E-01	6.0E+05	Inf	0
FPSO	1.5E-68	2.4E+04	2.4E+04	**100**	0	7.9E+03	7.9E+03	**100**	0	2.3E+04	2.3E+04	**100**	1.2E-134	1.5E+04	1.5E+04	**100**
CLPSO	4.4E-01	6.0E+05	Inf	0	6.1E-04	4.8E+05	6.5E+05	60	1.2E+02	6.0E+05	Inf	0	4.2E+00	6.0E+05	Inf	0
NPSO	5.1E+01	6.0E+05	Inf	0	1.8E-02	6.0E+05	Inf	0	1.4E+02	6.0E+05	Inf	0	2.2E-01	6.0E+05	Inf	0
CCPSO	4.6E+01	6.0E+05	Inf	0	5.6E-02	6.0E+05	Inf	0	1.2E+04	6.0E+05	Inf	0	4.9E+03	6.0E+05	Inf	0
ACPSO	**0**	**2.4 E+03**	**2,40 E+03**	**100**	**0**	**1.1 E+03**	**1.1 E+03**	**100**	**0**	**2.4 E+03**	**2.4 E+03**	**100**	**0**	**1.9E+03**	**1.9E+03**	**100**
		F17				F18				F19				F20		
LWPSO	6.9E+00	6.0E+05	Inf	0	4.6E-02	5.9E+05	2.2E+06	25	7.7E-14	2.4E+05	2.4E+05	**100**	4.8E-05	4.9E+05	4.9E+05	**100**
RWPSO	1.4E+01	6.0E+05	Inf	0	1.1E+01	6.0E+05	Inf	0	2.9E-09	1.7E+05	1.7E+05	**100**	5.4E+06	6.0E+05	Inf	0
CFPSO	3.5E-09	6.0E+05	Inf	0	6.2E-14	1.6E+05	1.6E+05	**100**	7.0E-06	3.9E+04	3.9E+04	**100**	5.2E-18	1.1E+05	1.1E+05	**100**
FDRPSO	5.6E+00	4.2E+05	4.8E+05	70	5.4E-03	5.6E+05	2.3E+06	10	7.5E-05	1.2E+03	1.2E+03	**100**	1.2E+06	4.2E+05	3.8E+05	40
FPSO	**8.9E-16**	2.7E+04	2.7E+04	**100**	0	1.9E+04	1.9E+04	**100**	2.8E-137	1.1E+04	1.1E+04	**100**	6.3E-136	1.4E+04	1.4E+04	**100**
CLPSO	1.3E+01	6.0E+05	Inf	0	8.7E-01	6.0E+05	Inf	0	2.4E-06	1.9E+05	1.9E+05	**100**	3.4E+00	5.5E+05	1.9E+06	20
NPSO	5.0E-01	5.1E+05	8.8E+05	45	2.8E-02	4.7E+05	6.2E+05	35	8.9E-07	1.1E+05	1.1E+05	**100**	3.8E+01	1.7E+05	1.7E+05	**100**
CCPSO	2.1E+01	6.0E+05	Inf	0	3.9E+02	6.0E+05	Inf	0	1.2E-04	1.8E+05	1.8E+05	**100**	1.7E+14	6.0E+05	Inf	0
ACPSO	**8.9E-16**	**2.7 E+03**	**2.7 E+03**	**100**	**0**	**2.1 E+03**	**2.1 E+03**	**100**	**0**	**1.2E+03**	**1.2E+03**	**100**	**0**	**1.7 E+03**	**1.7 E+03**	**100**
		F21				F22				F23				F24		
LWPSO	5.7E-02	5.5E+05	5.5E+05	**100**	1.9E+01	6.0E+05	Inf	0	1.2E+04	6.0E+05	Inf	0	2.1E+02	6.0E+05	Inf	0
RWPSO	3.6E+03	6.0E+05	Inf	0	7.3E+01	6.0E+05	Inf	0	6.9E+05	6.0E+05	Inf	0	7.9E+02	6.0E+05	Inf	0
CFPSO	4.7E-15	3.0E+05	3.0E+05	100	2.1E+01	6.0E+05	Inf	0	1.1E-04	1.4E+05	1.4E+05	**100**	1.8E+02	6.0E+05	Inf	0
FDRPSO	3.9E-10	6.0E+05	Inf	0	2.2E+01	6.0E+05	Inf	0	6.8E+03	2.5E+05	2.5E+05	**100**	1.8E+02	6.0E+05	Inf	0
FPSO	2.8E-133	2.4E+04	2.4E+04	**100**	1.3E-134	1.8E+04	1.8E+04	**100**	3.1E-131	2.0E+04	2.0E+04	**100**	0	2.3E+04	2.3E+04	**100**
CLPSO	1.9E+01	6.0E+05	Inf	0	2.8E+01	6.0E+05	Inf	0	1.4E+05	6.0E+05	Inf	0	2.4E+02	6.0E+05	Inf	0
NPSO	5.8E+04	6.0E+05	Inf	0	1.7E+01	6.0E+05	Inf	0	2.2E-08	2.1E+05	2.1E+05	**100**	1.9E+02	6.0E+05	Inf	0
CCPSO	1.2E+05	6.0E+05	Inf	0	4.6E+02	6.0E+05	Inf	0	1.1E+07	6.0E+05	Inf	0	1.2E+04	6.0E+05	Inf	0
ACPSO	**0**	**3.3E+03**	**3.3E+03**	**100**	**0**	**1.7E+03**	**1.7E+03**	**100**	**0**	**3.5E+03**	**3.5E+03**	**100**	**0**	**2.4E+03**	**2.4E+03**	**100**
		F25				F26				F27				F28		
WPSO	1.7E+01	6.0E+05	Inf	0	2.5E+02	6.0E+05	Inf	0	1.0E+01	6.0E+05	Inf	0	4.9E-02	6.0E+05	1.2E+07	0
RWPSO	2.6E+01	6.0E+05	Inf	0	7.4E+02	6.0E+05	Inf	0	1.6E+01	6.0E+05	Inf	0	1.0E+01	6.0E+05	Inf	0
CFPSO	1.0E+01	4.3E+05	4.6E+05	90	1.7E+02	6.0E+05	Inf	0	1.8E-08	2.1E+05	2.1E+05	**100**	2.1E-11	1.9E+05	1.8E+05	95
FDRPSO	1.2E+01	6.0E+05	Inf	0	2.4E+02	6.0E+05	Inf	0	1.3E+00	1.0E+07	1.8E+07	40	6.9E-03	4.6E+05	6.5E+05	50
FPSO	**0**	4.3E+04	4.3E+04	**100**	**0**	2.3E+04	2.3E+04	**100**	**8.9E-16**	2.7E+04	2.7E+04	**100**	**0**	1.8E+04	1.8E+04	**100**
CLPSO	2.2E+01	6.0E+05	Inf	0	2.6E+02	6.0E+05	Inf	0	1.5E+01	6.0E+05	Inf	0	1.0E+00	6.0E+05	Inf	0
NPSO	2.1E+01	6.0E+05	Inf	0	2.2E+02	6.0E+05	Inf	0	2.8E+00	6.0E+05	Inf	0	2.4E-02	4.9E+05	8.5E+05	35
CCPSO	3.9E+01	6.0E+05	Inf	0	1.3E+04	6.0E+05	Inf	0	2.1E+01	6.0E+05	Inf	0	3.9E+02	6.0E+05	Inf	0
ACPSO	**0**	**3.9E+03**	**3.9E+03**	**100**	**0**	**2.4 E+03**	**2.4 E+03**	**100**	**8.9E-16**	**2.7 E+03**	**2.7 E+03**	**100**	**0**	**2.1 E+03**	**2.1 E+03**	**100**

Table 4. Statistical results on five different aspects to evaluate the performance

	D	LWPSO	RWPSO	CFPSO	FDRPSO	FPSO	CLPSO	NPSO	CCPSO	ACPSO
w/t/l	10	27/1/0	27/1/0	26/2/0	27/1/0	15/13/0	27/1/0	27/1/0	28/0/0	
	30	28/0/0	28/0/0	27/1/0	27/1/0	15/13/0	28/0/0	28/0/0	28/0/0	
#bfes	10	0	0	0	0	1	0	0	0	27
	30	0	0	0	0	1	0	0	0	27
#s/#ps/#ns	10	15/3/10	4/4/20	15/10/3	11/8/9	27/1/0	13/4/11	11/6/11	1/0/27	27/1/0
	30	9/5/14	4/0/24	17/3/8	7/10/11	27/1/0	3/3/22	7/5/16	1/0/21	27/1/0
+/=/-	10	21/9/0	26/2/0	16/12/0	25/3/0	13/15/0	21/7/0	22/6/0	28/0/0	
	30	12/16/0	26/2/0	23/5/0	26/2/0	15/13/0	26/2/0	24/4/0	28/0/0	

The ACPSO algorithm is more robust as it finds the global optima of unimodal, multimodal, and rotated function with significant accuracy. The new learning strategy of ACPSO has a better exploration ability, which enables the swarm to explore the search space. This efficient exploration ability of ACPSO helps to avoid local trapping and improves its convergence speed.

6.2 Results on convergence speed

The mean results on the convergence speed of all algorithms are presented in column two of Table 2 and Table 3 for 10 and 30-*D* test functions respectively. In this experiment, the value of Max_{FEs} is set to 2.0E+05 and 6.0E+05 for 10 and 30-*D* test functions respectively. Theresults show that the ACPSO outperforms its contenders in the majority of test functions.

6.2.1 Convergence speed of 10-dimensional test functions

The second column of each function in Table 2 shows the mean number of function evaluations *FEs* for 10-*D* test functions. It is observed that only the ACPSO and FPSO are able to reach the pre specified accuracy threshold ε for F3 noisy function, while other BPSO variants are unable to converge until they Max_{FEs} reached. Also, the FPSO is faster than the ACPSO for F3 function on 10-*D* test functions. Similarly, the FDRPSO has the same number of *FEs* for F8 function. The presented results show that the ACPSO has the best number of *FEs* for almost all functions except F3 and F8 functions.

Although the FPSO has the same mean accuracy for 13 test functions and competitive results for 10 test functions with the ACPSO, its convergence speed toward the optimum location is very low, as it requires the larger number of FES to achieve the desired goal. To present the results qualitatively, the convergence graph of 10-*D* test functions is shown Figure 1. Due to space limitation, the mean results of eight test functions of nine BPSO variants are plotted. The plots show that the ACPSO performs better than its contenders with better convergence rate. It also shows that the convergence rate of ACPSO is improving during evolution. However, the convergence speed of other BPSO variants is decreasing. The function F3 is a noisy function so, the ACPSO has a slow convergence rate, but still better than other BPSO variants. The Figure 1 also shows that the ACPSO needs a small number of to reach the pre-specified accuracy threshold ε. This is due to the intelligent learning mechanism of ACPSO, which consistently improves the fitness of particles during each iteration. Moreover, locating the global optimum while improving the convergence speed has been achieved through the proposed technique.

6.2.2 Convergence speed of 30-dimensional test functions

The mean number of function evaluations FEs for 30-*D* test functions are presented in the second column of Table 3. The reason for increasing the number of dimensions for test functions is to investigate the impact of increasing the size of dimensions of test functions on the performance of the proposed algorithm. It is noted that the FPSO has less number of FEs for the F3 function. In addition, the FDRPSO has an equal number of FEs as ACPSO has for F8 and F19 functions. For the rest of all functions, ACPSO achieved the best number of FEs . For each test function, all algorithms have increased the number of FEs as compared with 10-*D* test function's results.

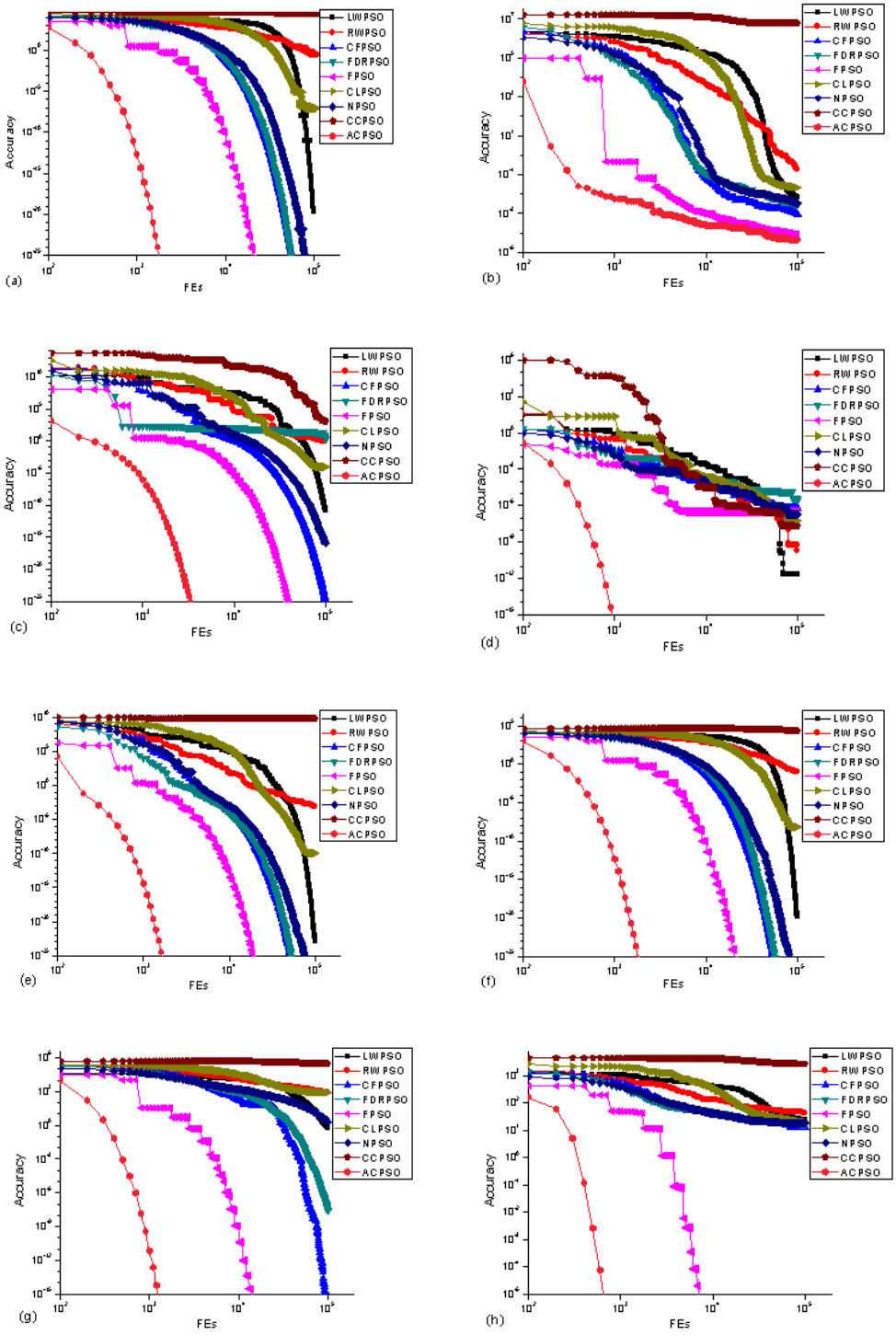

Figure 1. Convergence graphs (a-h) are representing the convergence speed on 10-*dimensional* functions F1, F3, F7, F19, F20, F21, F23, and F26 respectively.

Table 5. ***t*-test results comparing ACPSO with other algorithms**

Algorithm	Dim	Test Functions																										
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
LWPSO	10	=	=	+	+	=	=	+	+	+	=	+	=	=	=	+	+	+	+	=	+	+	+	+	+	+	+	+	+
	30	=	=	+	=	=	=	=	=	+	=	+	=	=	=	+	+	=	=	=	=	=	+	+	+	+	+	+	+
RWPSO	10	+	+	+	+	=	+	+	+	+	+	+	+	+	+	+	+	+	+	=	+	+	+	+	+	+	+	+	+
	30	+	+	+	+	+	+	=	+	+	+	+	+	+	+	+	+	+	+	=	+	+	+	+	+	+	+	+	+
CFPSO	10	+	+	+	+	=	+	+	=	+	=	=	+	=	+	=	=	+	=	+	+	+	=	=	+	=	+	=	+
	30	+	+	+	+	=	+	+	=	+	+	=	+	+	+	+	+	+	=	+	+	=	+	+	+	+	+	+	+
FDRPSO	10	+	+	+	+	=	+	+	+	+	=	+	+	+	+	+	+	=	+	+	+	+	+	+	+	+	+	+	+
	30	+	+	=	+	=	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
FPSO	10	+	=	+	+	=	+	+	+	+	=	=	=	+	=	=	+	=	=	=	+	+	+	+	=	=	=	=	=
	30	+	=	+	+	=	+	+	+	+	=	=	=	+	=	=	+	=	=	+	+	+	+	+	=	=	=	=	=
CLPSO	10	=	=	+	=	=	+	+	+	+	=	+	+	=	=	+	+	+	+	+	+	+	+	+	+	+	+	+	+
	30	+	=	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	=	+	+	+	+	+	+	+	+
NPSO	10	+	=	+	=	=	=	+	+	+	+	+	+	+	=	+	+	=	+	+	+	+	+	+	+	+	+	+	+
	30	+	=	+	+	=	+	+	+	+	+	+	+	+	+	+	+	+	+	+	=	+	+	=	+	+	+	+	+
CCPSO	10	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
	30	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Although the FPSO has the best mean fitness value for 13 test functions, it has only one function with an equal number of function evaluations to ACPSO. For the rest of 27 problems, the ACPSO has better speed with less number of FEs on 30-*D* test functions. The row number two of Table 4 represents the statistical results as the best mean number of function evaluations "#bfes" for the 10 and 30-*D* test functions. The comprehensive results show that the ACPSO has the tendency to converge faster than the compared BPSO variants with a smaller number of function evaluations. The results also reveal that the FPSO is the only algorithm that has the least number of function evaluations for F3 test function for both 10 and 30-*D* test functions.

The ACPSO is a robust algorithm that consistently reduces the fitness value (during all runs) below the specified accuracy threshold ε. The efficient learning mechanism of ACPSO ensures that the particles do not waste the computational resources and improve their fitness during evolution.

6.3 Results on success performance (SP)

The third column of each test function in Table 2 and Table 3 shows the SP for 10-*D* and 30-*D* test functions respectively. If any algorithm does not converge to predefined accuracy level ε within the Max_{FEs} for any run, the number of successful runs will be zero and the result will become infinity which is represented by 'Inf'. The results show that the FPSO achieves better performance than the ACPSO for F3 function in both 10 and 30-*D* test functions. No variants of BPSO could converge to the accuracy threshold within the maximum number of function evaluations for F3 function. The FDRPSO has the same value as ACPSO has for F8 in case of 10-*D* test functions and it also has the same results for F8 and F19 in the case of 30-*D* test functions.

It is also observed that the value of SP for 30-*D* test functions are higher than the 10-*D* test functions. An intelligent learning mechanism enables the ACPSO to reach the accuracy threshold within the Max_{FEs},which ensures the smaller number of function evaluations as success performance.

6.4 Results on success rate (SR)

The fourth column of each test function in Table 2 and Table 3 shows the success rate for 10-*D* and 30- *D* test functions respectively. It is observed that the ACPSO offers the highest reliability over all the test functions. During all runs, the ACPSO reliably achieves the acceptable solution with success rate of 100% on all test functions except the F3 function. The FPSO has higher SR for F3 function. For F9 function, the ACPAO and FPSO find the acceptable solution during all runs for 10 and 30-*D* test functions, whereas, the CFPSO successfully locates the optimum for 80% of runs for 10-*D* test functions. It is observed that the success rate of BPSO variants on un-rotated and most of the separable functions are significantly better than rotated and un-separable functions due to higher complexity.

The SR is summarized in the third row of Table 4 as *"#s/#ps/#ns"*. It indicates the number

of test functions that solved the problem completely (i.e., SR=100%), partially solved (i.e., 0%<SR<100%) or has never solved (i.e., SR=0%). The results show that the ACPSO and FPSO have 100% success rate for 27 test functions, whereas one test function was partially solved. However, the FPSO has a better success rate than ACPSO for the F3 test function. The F19 test function has the 100% success rate for all algorithms on 10 and 30-*D* test functions.

On the other hand, the FPSO successfully locates the optimal solution for all test functions, except test function F3, with a 100 % success rate for the 10 and 30-*D* test functions. The FPSO has competitive results and even better success rates than ACPSO for partially solved F3 test function. It is also observed that the success rate of other variants of BPSO is decreased as the dimensions of a test function are increased. Those test functions whose success rates are 100% have the same *FEs* . Similarly, the SP is directly proportional to the success rate. As the success rate is decreased, the success performance is also decreased. The ACPSO and the FPSO are the only algorithms, which successfully locate the optimum during all runs for 27 test functions on 10 and 30- *D* test functions.

6.5 Results on t-test

The *t*-test results are represented as "+/=/-" in Table 5 for 10 and 30-*D* test functions. The '+' indicates that the ACPSO is statistically better, '=' shows that they both are statistically equal, and '-' represents that the proposed algorithm is statistically worse than the compared BPSO. The results also show that the ACPSO statistically performs better than its competitor. For instance, the ACPSO performs better than BPSO variants on 77% and 92% in 10-*D* and 30-*D* test functions.

An important observation is that despite the large difference between the mean fitness values produced by the ACPSO and other BPSO variants, the *t*-test results revealed the fact that the performance difference is insignificant at the statistical level. Like in Table 5, for F16 in CFPSO on 10-dimensions test function, the statistical result is insignificant in the presence of zero percent SR as shown in Table 2. The similar ambiguous scenarios are observed for test function F2 of NPSO, test function F3 of RWPSO, the F7 test function of RWPSO, test functions F8, F12, and F17 of LWPSO. This ambiguity is because when certain algorithm runs for a predefined number of function evaluations, it has a probability to stagnate at some local optimum and to produce relatively larger fitness value.

The summarized results of *t*-test are represented in the fourth row of Table 4. We have also observed that as the dimensions of test functions are increased, the performance of ACPSO at statistical level is also increased. The only LWPSO whose statistical results, for 30-*D* test functions as compared with 10-*D* test functions, are improved. The statistical results of ACPSO are significantly better than LWPSO for 21 test functions in 10-*D* and for 12 test functions in 30-*D* test functions. Whereas, the statistical results for RWPSO are remained same. The statistical results of the rest of BPSO variants have been decreased.

7 CONCLUSION

FROM empirical results and analysis, it has been observed that the ACPSO algorithm shows very promising results on a diverse set of test functions. The achieved results show that the ACPSO has higher accuracy, improved convergence speed, more consistent, reliable, and robust than the other BPSO variants. These promising results have been achieved without introducing any additional parameter or any extra computation into the basic PSO framework. Moreover, the ACPSO have better performance on 30-*D* test functions as compared to 10-*D* test functions. Our results also reveal the fact that the ACPSO offers better results for those problems whose optimal solution lies at the origin.

Therefore, this simple variant can be applied to many real-world optimization problems where the quick response with higher accuracy is required. In future, we will test the performance of ACPSO on a large scale and multimodal test functions and will also apply it to some real-world application.

8 ACKNOWLEDGMENT

THANKS to Ms. Sehrish Shafi (University of York, UK) for her expertise in proofreading and reviewing this paper.

9 REFERENCES

Adewumi A. O., & Arasomwan M. A. (2016). On the performance of particle swarm

optimisation with (out) some control parameters for global optimisation. *International Journal of Bio-Inspired Computation, 8*(1), 14-32.

Ali H., Shahzad W., & Khan F. A. (2012). Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization. *Applied Soft Computing, 12*(7), 1913-1928.

Beheshti Z., & Shamsuddin S. M. (2015). Non-parametric particle swarm optimization for global optimization. *Applied Soft Computing, 28*, 345-359.

Bonyadi M. R., Michalewicz Z., & Li X. (2014). An analysis of the velocity updating rule of the particle swarm optimization algorithm. *Journal of Heuristics, 20*(4), 417-452.

Bonyadi M. R., & Michalewicz Z. (2016). Particle swarm optimization for single objective continuous space problems: a review. *Evolutionary Computation*.

Chu X., Niu B., Liang J., & Lu Q. (2016). An orthogonal-design hybrid particle swarm optimiser with application to capacitated facility location problem. *International Journal of Bio-Inspired Computation, 8*(5), 268-285.

Cui Z., Cai X., & Zeng J. (2009). Chaotic performance-dependent particle swarm optimization. *International Journal of Innovative Computing, Information and Control, 5*(4), 951-960.

Cui Z., Cai X., & Shi Z. (2012). Using fitness landscape to improve the performance of particle swarm optimization. *Journal of Computational and Theoretical Nanoscience, 9*(2), 258-265.

Dorigo M., & Di Caro G. (1999). *Ant colony optimization: a new meta-heuristic*. Paper presented at the Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on.

Eberhart R., & Kennedy J. (1995). *A new optimizer using particle swarm theory*. Paper presented at the 6ht Int. Symp. Micromachine Human Sci., Nagoya Japan.

Eberhart R. C., & Shi Y. (2001). *Tracking and optimizing dynamic systems with particle swarms.* Paper presented at the Evolutionary Computation, 2001. Proceedings of the 2001 Congress on.

Eberhart Y. S. a. R. C. (1998). *A Modified particle swarm optimizer*. Paper presented at the IEEE International Conf. on Evolutionary Computation.

Engelbrecht A. (2012). *Particle swarm optimization: Velocity initialization.* Paper presented at the Evolutionary Computation (CEC), 2012 IEEE Congress on.

Engelbrecht A. P. (2006). *Fundamentals of computational swarm intelligence*: John Wiley & Sons.

Engelbrecht A. P. (2013). *Particle Swarm Optimization: Global Best or Local Best?* Paper presented at the Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC), 2013 BRICS Congress on.

Fang H., Chen L., & Wang W. (2008). *A novel PSO algorithm for global optimization of multi-dimensional function.* Paper presented at the 2008 Chinese Control and Decision Conference.

Hu M., Wu T., & Weir J. D. (2013). An adaptive particle swarm optimization with multiple adaptive methods. *Evolutionary Computation, IEEE Transactions on, 17*(5), 705-720.

James K., & Russell E. (1995). *Particle swarm optimization.* Paper presented at the Proceedings of 1995 IEEE International Conference on Neural Networks.

Jamil M., & Yang X. S. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation, 4*(2), 150-194.

Kennedy M. C. a. J. (2002). The Particle Srwarm -- Explosion, Stability, and Convergence in a Multidimensional Complex Space. *IEEE Transactions on Evolutionary Computation, 6*(1), 58-73.

Li X. (2004). *Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization.* Paper presented at the Genetic and Evolutionary Computation–GECCO 2004.

Liang J. J., Qin A. K., Suganthan P. N., & Baskar S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *Evolutionary*

Computation, IEEE Transactions on, 10(3), 281-295.

Mendes R., Kennedy J., & Neves J. (2004). The fully informed particle swarm: simpler, maybe better. *Evolutionary Computation, IEEE Transactions on, 8*(3), 204-210.

Mirjalili S., & Lewis A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51-67.

Nakisa B., Nazri M. Z., Rastgoo M. N., & Abdullah S. (2014). A survey: Particle swarm optimization based algorithms to solve premature convergence problem. *Journal of Computer Science, 10*(9), 1758.

Park J.-B., Jeong Y.-W., Shin J.-R., & Lee K. Y. (2010). An improved particle swarm optimization for nonconvex economic dispatch problems. *Power Systems, IEEE Transactions on, 25*(1), 156-166.

Parrott D., & Li X. (2006). Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *Evolutionary Computation, IEEE Transactions on, 10*(4), 440-458.

Peram T., Veeramachaneni K., & Mohan C. K. (2003). *Fitness-distance-ratio based particle swarm optimization.* Paper presented at the Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE.

Qu B.-Y., Suganthan P. N., & Das S. (2013). A distance-based locally informed particle swarm model for multimodal optimization. *Evolutionary Computation, IEEE Transactions on, 17*(3), 387-402.

R.C. Eberhart J. K. a. (1995). *Particle Swarm Optimization*. Paper presented at the IEEE Intenational Conf. on Neural Networks.

Rauf A., & A. Aleisa E. (2015). PSO based Automated Test Coverage Analysis of Event Driven Systems. *Intelligent Automation & Soft Computing, 21*(4), 491-502.

Sadhasivam N., & Thangaraj P. (2017). Design of an improved PSO algorithm for workflow scheduling in cloud computing environment. *Intelligent Automation & Soft Computing, 23*(3), 493-500.

Salomon R. (1996). Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms. *BioSystems, 39*(3), 263-278.

Suganthan P. N., Hansen N., Liang J. J., Deb K., Chen Y.-P., Auger A., & Tiwari S. (2005). Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *KanGAL Report, 2005005*.

Tian D. (2017). Particle Swarm Optimization with Chaos-based Initialization for Numerical Optimization. *Intelligent Automation & Soft Computing*, 1-12.

van den Bergh F., & Engelbrecht A. (2002). *A new locally convergent particle swarm optimizer.* Paper presented at the Proceedings of the IEEE international conference on systems, man, and cybernetics.

Van den Bergh F., & Engelbrecht A. P. (2010). A convergence proof for the particle swarm optimiser. *Fundamenta Informaticae, 105*(4), 341-374.

Wang G.-G., Deb S., & Coelho L. (2015). Earthworm optimization algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *International Journal of Bio-Inspired Computation*.

Wang G.-G., Deb S., & Cui Z. (2015). Monarch butterfly optimization. *Neural computing and applications*, 1-20.

Wang G.-G. (2016). Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Comput*, 1-14.

Wang G.-G., Gandomi A. H., Alavi A. H., & Deb S. (2016). A multi-stage krill herd algorithm for global numerical optimization. *International Journal on Artificial Intelligence Tools, 25*(02).

Wang H.-C., & Yang C.-T. (2016). Enhanced Particle Swarm Optimization With Self-Adaptation Based On Fitness-Weighted Acceleration Coefficients. *Intelligent Automation & Soft Computing, 22*(1), 97-110.

Yang X.-S., & Deb S. (2009). *Cuckoo search via Lévy flights.* Paper presented at the Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on.

Yang X. (2008). Firefly Algorithm (chapter 8). *Nature-inspired Metaheuristic Algorithms, Luniver Press*.

Zhan Z.-H., Zhang J., Li Y., & Chung H.-H. (2009). Adaptive particle swarm optimization. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 39*(6), 1362-1381.

Zhan Z.-H., Zhang J., Li Y., & Shi Y.-H. (2011). Orthogonal learning particle swarm

optimization. *Evolutionary Computation, IEEE Transactions on*, *15*(6), 832-847.

10 NOTES ON CONTRIBUTORS

Yasir Mehmood is currently a doctoral student at the Dept. of Computer Sciences, National University of Computer and Emerging Sciences, Islamabad.
His research interest includes multimodal function optimization using swarm intelligence and evolutionary computation techniques.

Waseem Shahzad is a Professor and Head of Computer Sciences Department.
He is interested in Machine Learning, Data Mining, Data Warehousing, and Evolutionary Computing. Currently, he is working on associative classification and feature subset selection using Swarm Intelligence techniques.