Journal of Engineering Science, Vol. 11, 27–48, 2015

Particle Swarm Optimisation with Improved Learning Strategy

Wei Hong Lim and Nor Ashidi Mat Isa*

School of Electrical and Electronic Engineering,
Universiti Sains Malaysia, Engineering Campus,
14300 Nibong Tebal, Pulau Pinang, Malaysia

*Corresponding author: ashidi@usm.my

Abstract: *In this paper, a new variant of particle swarm optimisation (PSO) called PSO with improved learning strategy (PSO-ILS) is developed. Specifically, an ILS module is proposed to generate a more effective and efficient exemplar, which could offer a more promising search direction to the PSO-ILS particle. Comparison is made on the PSO-ILS with 6 well-established PSO variants on 10 benchmark functions to investigate the optimisation capability of the proposed algorithm. The simulation results reveal that PSO-ILS outperforms its peers for the majority of the tested benchmarks by demonstrating superior search accuracy, reliability and efficiency.*

Keywords: Particle swarm optimisation, improved learning strategy, global optimisation, metaheuristic search, swarm intelligence

1. INTRODUCTION

Inspired by the collective and collaborative behaviours of bird flocking and fish schooling in searching for food sources,[1,2] Kennedy and Eberhart[1] proposed a new population-based metaheuristic search (MS) algorithm called particle swarm optimisation (PSO) in 1995. From the optimisation perspective, each individual member (i.e., particle) of the PSO swarm represents a potential solution to a given problem, whereas the location of the food source denotes the global optimum solution. Each particle moves stochastically to locate the food source during the search process. In addition, all the population members of the PSO swarm collaborate with each other through information sharing. This interaction enables all the particles to gradually move towards the food sources and eventually leads to the swarm convergence.[2] Since the inception of PSO, this algorithm has been applied to address various real-world problems due to its simplicity.[3,4]

Despite the popularity of PSO in computational intelligence research, previous study[5] revealed that this algorithm suffers from the premature convergence issue because the particle has a high tendency to be trapped in local optima regions of the search space. Another notable drawback of PSO is the intense conflict between the exploration and exploitation searches of the algorithm. Excessive exploration tends to inhibit the swarm convergence, whereas too much exploitation can lead to the rapid diversity loss of swarm.[6] Numerous works[2–4] have been conducted by researchers in the past decades to address the aforementioned drawbacks of PSO. While some improved PSO variants can preserve the swarm diversity to some extent, these improvements are usually attained at the expense of slow convergence or complicated algorithmic structures. Addressing the premature convergence issue of PSO without significantly jeopardising the simplicity of the algorithmic frameworks and the algorithm's convergence speed remains a challenge.

In this paper, a new PSO variant called the PSO with improved learning strategy (PSO-ILS) is proposed. The main innovation of this study is the development of a novel ILS module, which aims to generate promising exemplars to guide the search directions of PSO-ILS particles. Unlike most existing PSO variants, the exemplar of each PSO-ILS particle is unique and is generated by the ILS module by considering the useful information contributed by all population members of PSO-ILS.

The remainder of this paper is organised as follows. Section 2 briefly discusses some related works. Section 3 details the methodologies of the PSO-ILS. Section 4 provides the experimental settings and simulation results. Finally, Section 5 presents the conclusion drawn from the work performed.

2. RELATED WORKS

In this section, the mechanism of the basic PSO (BPSO) is briefly discussed and is followed by a literature review of several well-established PSO variants.

2.1 Basic PSO

In BPSO, each particle i represents a potential solution of a D-dimensional problem, and its current state is associated with 2 vectors, i.e., the position vector $X_i = [X_{i1}, X_{i2}, \ldots, X_{iD}]$ and the velocity vector $V_i = [V_{i1}, V_{i2}, \ldots, V_{iD}]$. Unlike most existing MS algorithms, each PSO particle i can memorise the best experience that it ever achieved, which is represented by the personal best position $P_i = [P_{i1}, P_{i2}, \ldots, P_{iD}]$. During the search process, the trajectory of each particle i in the search space is dynamically adjusted according to particle i's self-cognitive component P_i, as well as the group best experience observed by the population, $P_g = [P_{g1}, P_{g2}, \ldots, P_{gD}]$ [1, 6]. At the $(t + 1)$-th iteration of the search process, the d-th dimension of particle i's velocity, $V_{i,d}(t + 1)$, and position $X_{i,d}(t + 1)$, are updated as follows:

$$V_{i,d}(t+1) = \omega V_{i,d}(t) + c_1 r_1 (P_{i,d}(t) - X_{i,d}(t)) + c_2 r_2 (P_{g,d}(t) - X_{i,d}(t)) \quad (1)$$

$$X_{i,d}(t+1) = X_{i,d}(t) + V_{i,d}(t+1) \quad (2)$$

where $i = 1, 2, \ldots, S$ is the particle's index; S is the population size; c_1 and c_2 are the acceleration coefficients that control the effects of the self-cognitive (i.e., P_i) and social (i.e., P_g) components, respectively; r_1 and r_2 are two random numbers generated from a uniform distribution with the range of [0, 1]; and ω is the inertia weight used to balance the exploration/exploitation searches of particles.[6]

2.2 PSO Variants and Improvements

Numerous studies have been performed to alleviate the drawbacks of BPSO. One of the most commonly used strategies is known as parameter adaptation. Clerc and Kennedy[7] incorporated a constriction factor χ into the PSO to prevent swarm explosion. To achieve better regulation of exploration and exploitation searches, Ratnaweera et al.[8] developed a time-varying acceleration coefficient strategy to dynamically change c_1 and c_2 with time. Alternatively, Juang et al.[9] utilised the fuzzy set theory to adaptively adjust c_1 and c_2. Zhan et al.[10] developed an evolutionary state estimation (ESE) module to identify the swarm's evolutionary state of their proposed adaptive PSO (APSO). The outputs of the ESE module are used to adaptively adjust the ω, c_1 and c_2 of each APSO particle. Recently, Leu and Yeh[11] proposed a grey PSO by capitalising on the grey relational analysis to tune the particles' ω, c_1 and c_2.

Population topology emerges as another crucial factor to determine the PSO's performance because this factor controls the information flow rate of the best solution within the swarm.[12] Mendes et al.[13] advocated that each particle's movement is affected by all of its neighbourhood members and subsequently proposed the fully connected PSO (FIPS). A flexible PSO (FlexiPSO) was developed by Kathrada[14] by combining the global and local versions of PSO. Montes de Oca et al.[15] incorporated the concept of time-varying population topology into their Frankenstein PSO (FPSO). Initially, all FPSO particles are connected with fully connected topology. The topology connectivity of each FPSO particle is gradually decreased over time and eventually reduced into the ring topology. Marinakis and Marinaki[16] proposed a PSO with expanding neighbourhood topology (PSOENT), where the particle's neighbourhood expands based on the quality of the produced solutions.

Another promising line of research involves the exploration of the PSO learning strategy. Liang et al.[17] proposed a comprehensive learning PSO (CLPSO) by suggesting that each dimensional component of a particle can learn from its P_i or from the other particle's personal best position. An improved variant called feedback learning PSO with quadratic inertia weight (FLPSO-QIW) was proposed by Tang et al.,[18] where each particle generates the potential exemplars from the first 50% of the fitter particles. Alternatively, Nasir et al.[19] proposed the dynamic-neighbourhood-learning-based PSO (DNLPSO). The exemplar of each DNLPSO particle is selected from its neighbourhood made dynamic in nature. Huang et al.[20] employed multiple global best particles to update the particle's velocity in their example-based learning PSO (ELPSO). Zhan et al.[21] capitalised on the excellent prediction capability of the orthogonal experiment design (OED) technique to construct effective exemplars for their proposed orthogonal learning PSO (OLPSO). Conversely, Zhou et al.[22] proposed a random position PSO (RPPSO) by employing the random particle to guide the swarm.

3. METHODOLOGY

The motivation for developing the ILS module is first described in this section, followed by a presentation of detailed descriptions of the ILS module. The velocity updating mechanism and the complete framework of the proposed PSO-ILS are then presented.

3.1 Motivation

Premature convergence remains a challenging issue for PSO, despite many improved variants of this algorithm having been developed to address this drawback.[17] This issue occurs because the particles of most existing PSO variants tend to learn from the P_g particle and neglect the information contributed by the non-global best particle during the search process. The lack of interaction between the particles and other non-global best particles can lead to the rapid diversity loss of the swarm, especially when the algorithm is used to solve problems with complex search environments. This scenario tends to increase the likelihood of the PSO swarm being trapped in the inferior regions of the search space, which consequently leads to poor optimisation results.

Considering that there is no convincing evidence to indicate that the fittest particle in the neighbourhood can actually find a better region than the second or third fittest particles in the swarm,[13] Liang et al.[17] proposed a comprehensive learning strategy by advocating that all particles' personal best positions could be used to update the velocity of each particle. The excellent performances of CLPSO[17] and its descendants (e.g., FLPSO-QIW,[18] DNLPSO,[19] OLPSO,[21] etc.) in solving the complex multimodal problems demonstrate that the derivation of exemplars from the non-fittest candidate solutions is a viable approach to sustain swarm diversity and to discourage the premature convergence.

It is noteworthy that the modified learning strategy of CLPSO and most of its descendants achieve the preservation of swarm diversity by reducing the effect of the P_g particle during the search process. This strategy, however, could introduce different trade-offs. For example, although CLPSO exhibits excellent capability in avoiding the local optima in complex multimodal problems, the convergence speed of this algorithm in solving unimodal and simple multimodal problems is significantly compromised.[17] In addition, OLPSO has a more complicated algorithmic framework because this approach employs the OED technique, which is more mathematically intensive, to derive the exemplars.[21]

Motivated by these observations, we propose an ILS module that offers an innovative mechanism to generate a unique exemplar for each PSO-ILS particle. Specifically, these exemplars are derived to replace both the self-cognitive and social components to guide the particle's search. Unlike the previous approaches, the proposed ILS module is less computationally intensive and is able to generate the exemplar with more promising guidance capability. The working mechanism of the ILS module is described in the following subsection.

3.2 ILS Module

The proposed ILS module works as follows. Initially, two exemplars called the cognitive exemplar ($c_{exp,i}$) and the social exemplar ($s_{exp,i}$) are generated for each PSO-ILS particle i. The proposed ILS module begins the derivations of both the $c_{exp,i}$ and $s_{exp,i}$ exemplars by sorting the personal best positions of all the population members based on the personal best fitness criterion. Specifically, the fittest members with personal best fitness ranked in the first quartile range are stored in $upper_i$, whereas the members in the remaining 3 quartiles are stored in $lower_i$. The approaches used to generate the $s_{exp,i}$ and $c_{exp,i}$ exemplars are explained as follows.

The $s_{exp,i}$ exemplar of particle i is generated from $upper_i$ via the random selection technique. Specifically, for each d-th dimensional component of $s_{exp,i}$, i.e., $s_{exp,i}(d)$, one member of $upper_i$ is randomly selected, and the d-th dimensional component of this selected member is assigned to $s_{exp,i}(d)$. Considering that all the members of $upper_i$ have the same probability to be selected, each $upper_i$ member is regarded as having an equal opportunity to contribute itself in deriving each dimensional component of $s_{exp,i}$.

On the other hand, the idea of constructing the $c_{exp,i}$ exemplar is inspired from Mendes et al.[13] and is computed as follows:

$$c_{\exp,i} = \frac{\sum_{P_k \in lower_i} c_k r_k P_k}{\sum_{P_k \in lower_i} c_k r_k} \tag{3}$$

where P_k refers to the personal best positions of all the population members stored in $lower_i$; r_k is a random number in the range of [0, 1] and c_k is the acceleration coefficient that is equally distributed among the N_i members from $lower_i$, which is calculated as $c_k = c_{all}/N_i$, where $c_k = 4.1$[13] Equation 3 also allows all the members in the $lower_i$ to have equal chances to contribute themselves during the derivation of the $c_{exp,i}$ exemplar.

Considering that the guidance of two exemplars might cause the "oscillation" phenomenon, as described in literature,[21] the third exemplar called the overall exemplar ($o_{exp,i}$) is derived from both the $s_{exp,i}$ and $c_{exp,i}$ exemplars (via a simple crossover procedure) to guide the particle *i* during the optimisation process. Specifically, if a randomly generated number is smaller than 0.5, the *d*-th dimensional component of $o_{exp,i}$, i.e., $o_{exp,i}(d)$, is donated by the $s_{exp,i}(d)$. Otherwise, it is obtained from the *d*-th dimensional component of $c_{exp,i}$.

The overall implementation of the proposed ILS module is illustrated in Figure 1 and includes the procedures for deriving the $s_{exp,i}$, $c_{exp,i}$ and $o_{exp,i}$ exemplars. Notably, the P_g particle in the population could be replaced by the newly obtained $o_{exp,i}$ exemplar if the latter has a more promising fitness value than the former.

$o_{exp,i}$ = **ILS**(*particle i*, P, $f(P)$, P_g, $f(P_g)$, *fes*)

Sort all population members according to their personal best fitness;
Assign the members with better personal best fitness values (in the first quartile range) into $upper_i$;
Assign the remaining members with worse personal best fitness values into $lower_i$;
*/*Generate the $s_{exp,i}$ exemplar*/*
for each dimension d **do**
    Randomly select a member, i.e., *rand_member*, from $upper_i$;
    $s_{exp,i}(d)$ = d-th component of the selected *rand_member*;
end for
*/*Generate the $c_{exp,i}$ exemplar*/*
Calculate $c_{exp,i}$ from $lower_i$ using Equation (3);
*/*Generate the $o_{exp,i}$ exemplar*/*
for each dimension d **do**
    if rand < 0.5 **then**
        $o_{exp,i}(d) = s_{exp,i}(d)$;
    else
        $o_{exp,i}(d) = c_{exp,i}(d)$;
    end if
end for
Perform fitness evaluation on $o_{exp,i}$;
Update P_g and $f(P_g)$ if $o_{exp,i}$ has better fitness;
$fes = fes + 1$;

Figure 1: Implementation of ILS module.

3.3 Velocity Updating Mechanism of PSO-ILS

Unlike BPSO, the proposed PSO-ILS updates the velocity of particle i based on the $o_{exp,i}$ exemplar instead of P_i and P_g. Considering that the derivation of $o_{exp,i}$ involves the stochastic mechanism, two possible cases can be encountered: (1) the $o_{exp,i}$ exemplar has better fitness than the personal best fitness of particle i [i.e., $f(o_{exp,i}) < f(P_i)$]; or (2) the $o_{exp,i}$ exemplar has equal or worse fitness than the personal best fitness of particle i [i.e., $f(o_{exp,i}) \geq f(P_i)$].

For case 1, particle i is allowed to be attracted towards the fitter $o_{exp,i}$ exemplar because the latter has better fitness and hence is more likely to offer a prominent search direction to guide the former. For case 2, particle i is encouraged to be repelled away from the inferior $o_{exp,i}$ exemplar because it is unlikely for the latter to improve the former's fitness. The new velocity update mechanism that is used to update the velocity of each PSO-ILS particle (i.e., V_i) is mathematically described as:

$$V_i = \begin{cases} \omega V_i + cr_3(o_{\exp,i} - X_i), & f(o_{\exp,i}) < f(X_i) \\ \omega V_i - cr_4(o_{\exp,i} - X_i), & \text{otherwise} \end{cases} \tag{4}$$

where r_3 and r_4 are random numbers ranging between 0 and 1.

Once the new velocity of particle i is obtained from Equation 4, the new position of particle i (i.e., X_i) is computed using Equation 2. The updated fitness of particle i, i.e., $f(X_i)$, is then evaluated and compared with those of P_i and P_g. The updated position of particle i will replace both P_i and P_g if the former has better fitness than the latter two.

3.4 Complete Framework of PSO-ILS

The complete implementation of the proposed PSO-ILS is presented in Figure 2. To conserve computational resources, particle i will only reconstruct the $o_{exp,i}$ exemplar if this exemplar fails to update the P_g m successive times. The variable $flag_i$ is defined to monitor the successive iteration when particle i fails to improve the global best solution. Notably, too small or too large values of m are undesirable. The former tends to reconstruct the $o_{exp,i}$ exemplar frequently and thus jeopardises the particle's search direction, while the latter could waste many computational resources on the local optima with the $o_{exp,i}$ exemplar, which is no longer effective.

```
PSO-ILS
Generate initial swarm and set up parameters for each particle;
while fes<FE_max do
    for each particle i do
        if flag_i>m then  /Execute ILS module to generate new o_exp,i exemplar*/
            o_exp,i= ILS(particle i, P, f(P), P_g, f(P_g), fes);
            flag_i= 0;
    end if
    Calculate V_i and X_i using Equations (4) and (2), respectively;
    Perform the fitness evaluation on updated X_i;
    fes= fes + 1;
    Update P_i, f (P_i), P_g, and f (P_g);
        if f (X_i) < f (P_g) then
            flag_i = 0;
        else
            flag_i = flag_i + 1;
        end if
     end for
end while
```

Figure 2: Complete framework of PSO-ILS.

4. EXPERIMENTAL

4.1 Benchmark Functions

Ten benchmark functions used for the performance evaluations are presented in Tables 1 and 2, which provides brief descriptions of the benchmarks' formulae, their feasible search range RG, and their accuracy level ε. All the employed benchmarks have different fitness landscapes, and the fitness value of their respective global optimum is equal to zero, i.e., $F_{min} = 0$.

Table 1: Formulae of ten benchmark functions used in this study.

f	Function	Formulae
F1	Sphere	$F_1(X_i)=\sum_{d=1}^{D} X_{i,d}^2$
F2	Schewefel 2.22	$F_2(X_i)=\sum_{d=1}^{D}\lvert X_{i,d}\rvert+\prod_{d=1}^{D}\lvert X_{i,d}\rvert$
F3	Schewefel 1.2	$F_3(X_i)=\sum_{d=1}^{D}(\sum_{j=1}^{d} X_{i,j})^2$
F4	Schwefel 1.2	$F_4(X_i)=\sum_{d=1}^{D}(\sum_{j=1}^{d} X_{i,j})^2$
F5	Hyper Ellipsoid	$F_5(X_i)=\sum_{d=1}^{D} 2^d \cdot X_{i,d}^2$
F6	Rastrigin	$F_6(X_i)=\sum_{d=1}^{D}(X_{i,d}^2-10\cos(2\pi X_{i,d})+10)$
F7	Noncontinuous Rastrigin	$F_7(X_i)=\sum_{d=1}^{D}(Y_{i,d}^2-10\cos(2\pi Y_{i,d})+10)$ where $Y_{i,d}=\begin{cases} X_{i,d} & ,\lvert X_{i,d}\rvert<0.5 \\ round(2X_{i,d})/2 & ,\lvert X_{i,d}\rvert\geq 0.5 \end{cases}$
F8	Griewank	$F_8(X_i)=\sum_{d=1}^{D} X_{i,d}^2/4000-\prod_{d=1}^{D}\cos(X_{i,d}/\sqrt{d})+1$
F9	Ackley	$F_9(X_i)=-20\exp(-0.2\sqrt{\sum_{d=1}^{D} X_{i,d}^2/D})-\exp(\sum_{d=1}^{D}\cos(2\pi X_{i,d})/D)+20+e$
F10	Weierstrass	$F_{10}(X_i)=\sum_{d=1}^{D}(\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k(X_{i,d}+0.5))])-D\sum_{k=0}^{k\max}[a^k\cos(\pi b^k)]$ $a=0.5, b=3, k\max=20$

Table 2: The search range and accuracy level of ten employed benchmark functions.

f	Function	*RG*	ε
F1	Sphere	$[-100,100]^D$	1.0E–6
F2	Schewefel 2.22	$[-10,10]^D$	1.0E–6
F3	Schewefel 1.2	$[-100,100]^D$	1.0E–6
F4	Schwefel 1.2	$[-100,100]^D$	1.0E–6
F5	Hyper Ellipsoid	$[-100,100]^D$	1.0E–6
F6	Rastrigin	$[-5.12,5.12]^D$	1.0E–2
F7	Noncontinuous Rastrigin	$[-5.12,5.12]^D$	1.0E–2
F8	Griewank	$[-600,600]^D$	1.0E–2
F9	Ackley	$[-32,32]^D$	1.0E–2
F10	Weierstrass	$[-0.5,0.5]^D$	1.0E–2

4.2 Simulation Settings of All Involved PSO Variants

6 well-established PSO variants are employed for extensive comparison with PSO-ILS. CLPSO, FLPSO-QIW, FIPS and OLPSO were selected because their learning strategies share specific similarities with that of PSO-ILS, i.e., these variants derive the exemplars from non-fittest solutions to guide the search. APSO is used to investigate the effectiveness of our proposed strategy against the parameter adaptation approach. In addition, FlexiPSO is the representative PSO variant developed from the different swarm topology.

The parameter settings of all the tested algorithms were extracted from their respective literature and are summarised in Table 3. The parameter settings of the selected peer algorithms are optimal, considering that their respective authors tuned these parameters using similar benchmarks. In addition, our empirical study reveals that the proposed PSO-ISL with m = 8 delivers satisfactory search performance. In this paper, the employed benchmarks are solved in 10 dimensions. All the involved algorithms were tested using the same population size of S = 10, with the stopping criterion of FE_{max} = 5.00E+04.

Table 3: Parameter settings of the involved PSO algorithms.

Algorithm	Parameter settings
APSO[10]	$\omega: 0.9-0.4$, $c_1+c_2:[3.0,4.0]$, $\delta=[0.05,0.1]$, $\sigma_{\max}=1.0$, $\sigma_{\min}=0.1$
CLPSO[17]	$\omega: 0.9-0.4$, $c=2.0$, $m=7$
FLPSO-QIW[18]	$\omega: 0.9-0.2$, $c_1: 2-1.5$, $c_2: 1-1.5$, $m=1$, $P_i=[0.1,1]$, $K_1=0.1$, $K_2=0.001$, $\sigma_1=1$, $\sigma_2=0$
FlexiPSO[14]	$\omega: 0.5-0.0$, $c_1,c_2,c_3:[0.0,2.0]$, $\varepsilon=0.1$, $\alpha=0.01\%$
FIPS[13]	$\chi=0.729$, $\sum c_i=4.1$
OLPSO[21]	$\omega: 0.9-0.4$, $c=2.0$, $G=5$
PSO-ILS	$\omega: 0.9-0.4$, $c_1=c_2=2.0$, $m=5$

4.3 Performance Metrics

In this paper, the authors evaluate the algorithm's performances based on three criteria, namely accuracy, reliability and efficiency, using the mean fitness value (F_{mean}), success rate (*SR*) and success performances (*SP*), respectively.[18] F_{mean} is defined as the mean value of the differences between the best (i.e., lowest) fitness obtained by the algorithm and the fitness at the global optima (F_{min}). *SR* denotes the consistency of an algorithm to achieve a successful run, i.e., when the algorithm achieves the solution with predefined ε. Finally, *SP* computes the number of FEs required by the algorithm to solve the problems with predefined ε.

The authors also employ a set of non-parametric statistical procedures[23] to perform rigorous comparisons between PSO-ILS and its peers. Specifically, the Wilcoxon test[23] is used for a pairwise comparison between the PSO-ISL and its peers. This test is conducted at the 5% significance level (i.e., $\alpha = 0.05$), and the values of h, R^+, R^- and p are reported. The h value indicates whether the performance of PSO-ISL is better (i.e., h = '+'), insignificant (i.e., h = '='), or worse (i.e., h = '–') than the other six algorithms at the statistical level. R^+ and R^- denote the sum of ranks that PSO-ISL outperforms and underperforms compared with the other methods. In addition, the p-value represents the minimal level of significance for detecting differences. A p-value less than α provides strong evidence to indicate the better results achieved by the best algorithm are statistically significant and did not occur by chance.

To conduct the multiple comparisons of the algorithms in the set of test suite employed, the Friedman test[23] and a set of post-hoc procedures to characterise the concrete differences among the algorithms were employed. In this study, the adjusted p-values (APVs) obtained were reported using the Bonferroni-Dunn, Holm and Hochberg methods.[23]

4.4 Comparison of PSO-MSCL with Other PSO Variants

4.4.1 Comparison of the F_{mean} results

In Table 4, it is observed that the proposed PSO-ILS exhibits the best search accuracy because this method outperforms its peers by a large margin for the majority of tested problems. Specifically, PSO-ILS is the only algorithm that successfully locates the global optima of all the tested benchmarks by achieving F_{mean} = 0.00E+00. CLPSO, FLPSO-QIW and OLPSO also exhibit their competitive searching accuracies in solving the tested benchmarks because these algorithms solve the functions F1, F2, F5, F9 and F10 with satisfactory F_{mean}

values. However, FIPS is observed as the worst performing optimiser, exhibiting the largest (i.e., worst) F_{mean} values for almost all the tested problems.

Table 4: Mean fitness, standard deviation and Wilcoxon test results for 10-*D* problems.

		APSO	CLPSO	FLPSO-QIW	FlexiPSO	FIPS	OLPSO	PSO-ILS
F1	F_{mean}	3.54E–03	3.34E–77	2.76E–50	3.63E–05	1.03E–01	5.90E – 63	**0.00E+00**
	SD	5.83E–03	1.53E–76	1.51E–49	2.92E–05	2.30E–01	2.73E – 62	0.00E+00
	h	+	+	+	+	+	+	
F2	F_{mean}	3.88E–03	2.00E–45	3.42E–39	2.21E–03	4.38E–02	7.93E–35	**0.00E+00**
	SD	5.45E–03	5.70E–45	1.23E–38	8.31E–04	6.42E–02	1.89E–34	0.00E+00
	h	+	+	+	+	+	+	
F3	F_{mean}	2.84E+00	2.48E–07	4.61E–03	9.33E–03	3.86E–01	3.63E–05	**0.00E+00**
	SD	2.79E+00	4.84E–07	2.03E–02	6.31E–03	9.50E–01	1.30E–04	0.00E+00
	h	+	+	+	+	+	+	
F4	F_{mean}	3.28E–01	7.91E–06	3.90E–01	9.46E–03	3.05E–01	1.64E–10	**0.00E+00**
	SD	1.30E–01	3.87E–05	4.26E–01	2.18E–03	4.03E–01	3.32E–10	0.00E+00
	h	+	+	+	+	+	+	
F5	F_{mean}	1.57E+00	2.07E–75	2.38E–56	6.39E–05	1.05E+01	9.28E–62	**0.00E+00**
	SD	3.15E+00	5.72E–75	1.29E–55	4.01E–05	2.69E+01	3.12E–61	0.00E+00
	h	+	+	+	+	+	+	
F6	F_{mean}	2.61E–03	1.66E–01	5.77E–01	2.87E–05	9.05E–01	1.06E+00	**0.00E+00**
	SD	3.47E–03	4.59E–01	7.16E–01	2.18E–05	1.44E+00	1.14E+00	0.00E+00
	h	+	+	+	+	+	+	
F7	F_{mean}	1.41E–03	1.67E–01	1.38E+00	1.83E+00	7.40E–01	2.47E+00	**0.00E+00**
	SD	1.82E–03	4.61E–01	1.16E+00	7.01E+00	1.51E+00	1.22E+00	0.00E+00
	h	+	+	+	+	+	+	
F8	F_{mean}	7.66E–02	2.24E–02	1.49E–02	5.55E–02	1.73E–01	2.97E–02	**0.00E+00**
	SD	2.33E–02	1.65E–02	1.09E–02	2.63E–02	2.03E–01	2.19E–02	0.00E+00
	h	+	+	+	+	+	+	
F9	F_{mean}	2.47E–02	3.55E–15	1.94E–14	6.67E–01	5.40E–01	3.55E–15	**0.00E+00**
	SD	1.99E–02	0.00E+00	1.93E–14	3.65E+00	8.35E–01	0.00E+00	0.00E+00
	h	+	+	+	+	+	+	
F10	F_{mean}	3.97E–02	**0.00E+00**	1.75E–03	2.24E–02	3.78E–01	**0.00E+00**	**0.00E+00**
	SD	2.88E–02	0.00E+00	4.30E–03	6.74E–03	4.43E–01	0.00E+00	0.00E+00
	h	+	=	+	+	+	=	

Notably, the proposed PSO-ILS shares similarities with CLPSO, FLPSO-QIW, FIPS and OLPSO in terms of the algorithmic framework design because the particles of these PSO variants are also guided by exemplars generated from the non-fittest solutions. Based on these observations, it could be deduced that the search behaviours of these five PSO variants are governed by their respective exemplars. Intuitively, the qualities of the exemplars produced in the PSO-ILS, CLPSO, FLPSO-QIW, FIPS and OLPSO could be assessed by comparing the optimisation capabilities of these algorithms. Considering that the proposed PSO-ILS outperforms or performs similarly to CLPSO, FLPSO-QIW, FIPS and OLPSO in all the tested benchmarks, it is reasonable to conclude that the exemplars generated by the proposed ILS module are more effective than those of CLPSO, FLPSO-QIW, FIPS and OLPSO. In other words, the exemplars of PSO-ILS are more capable of guiding their particles towards the promising regions of the search space compared with the other four peers.

4.4.2 Comparison of the non-parametric statistical test

The pairwise comparison results between PSO-ILS and its peers using the Wilcoxon test are summarised in Tables 4 and 5. Specifically, Table 4 presents the pairwise comparison results in each employed benchmark using the *h* values, whereas Table 5 reports the R^+ and R^- values obtained for each comparison and the associated *p*-value.

Table 4: Wilcoxon test between PSO-ILS and 6 other variants.

PSO-ILS vs.	APSO	CLPSO	FLPSO-QIW	FlexiPSO	FIPS	OLPSO
R^+	55.0	45.0	55.0	55.0	55.0	45.0
R^-	0.0	0.0	0.0	0.0	0.0	0.0
p-value	**1.95E–03**	**3.91E–03**	**1.95E–03**	**1.95E–03**	**1.95E–03**	**3.91E–03**

Table 4 demonstrates that the *h* values obtained from the Wilcoxon test are consistent with the reported F_{mean} values. This finding implies that the number of problems for which PSO-ILS significantly outperforms its peers is much larger than the number of problems for which the former is statistically equivalent to the latter. Table 5 confirms the significant improvements of PSO-ILS over its six peers in the independent pairwise comparison because all the *p*-values attained from the Wilcoxon test in Table 4 are less than $\alpha = 0.05$.

Table 5: Average ranking and the associated *p*-value obtained through Friedman test.

Algorithm	PSO-ILS	CLPSO	OLPSO	FLPSO-QIW	FlexiPSO	APSO
Ranking	**1.55**	2.70	3.50	3.85	4.90	5.30
Statistic			32.80			
p-value			**1.10E-05**			

Multiple comparisons[23] are also employed to rigorously evaluate the effectiveness of PSO-ILS. The results of the Friedman test, which include the average rankings of the compared algorithms and the associated *p*-values, are summarised in Table 6. PSO-ILS emerges as the best performing algorithm with the smallest average rank value of 1.55. Another notable observation from Table 6 is that the *p*-value computed using the Friedman test (i.e., p = 1.10E–05) is smaller than the level of significance considered (i.e., $\alpha = 0.05$). This result implies that a significant global difference is detected among the compared algorithms.

Table 6: Average ranking and associated *p*-value obtained using Friedman test.

Algorithm	PSO-ILS	CLPSO	OLPSO	FLPSO-QIW	FlexiPSO	APSO
Ranking	**1.55**	2.70	3.50	3.85	4.90	5.30
Statistic			32.80			
p-value			**1.10E–05**			

Based on these results, a set of post-hoc statistical analyses[23] was performed to identify the concrete differences for the control algorithm (i.e., PSO-ILS). The associated z values, unadjusted *p*-values and adjusted *p*-values (APVs) obtained from the aforementioned post-hoc procedures are presented in Table 7. At the significant level of $\alpha = 0.05$, all the post-hoc procedures confirm the improvement of PSO-ILS over the FIPS, APSO and FlexiPSO algorithms. The Holm and Hochberg procedures reveal more powerful capabilities than the Bonferroni-Dunn procedure because the former tests are able to confirm the significant outperformance of PSO-ILS against FLPSO-QIW and OLPSO at $\alpha =$ 0.10.

Table 7: Adjusted *p*-values (APVs) obtained using Bonferroni-Dunn, Holm and Hochberg procedures.

PSO-ILS vs.	*z*	Unadjusted *p*	Bonferroni-Dunn *p*	Holm *p*	Hochberg *p*
FIPS	4.81E+00	**1.00E–06**	**9.00E–06**	**9.00E–06**	**9.00E–06**
APSO	3.88E+00	**1.04E–04**	**6.23E–04**	**5.19E–04**	**5.19E–04**
FlexiPSO	3.47E+00	**5.25E–04**	**3.15E–03**	**2.10E–03**	**2.10E–03**
FLPSO-QIW	2.38E+00	**1.73E–02**	1.04E–01	5.18E–02	5.18E–02
OLPSO	2.02E+00	**4.35E–02**	2.61E–01	8.71E–02	8.71E–02
CLPSO	1.19E+00	2.34E–01	1.00E+00	2.34E–01	2.34E–01

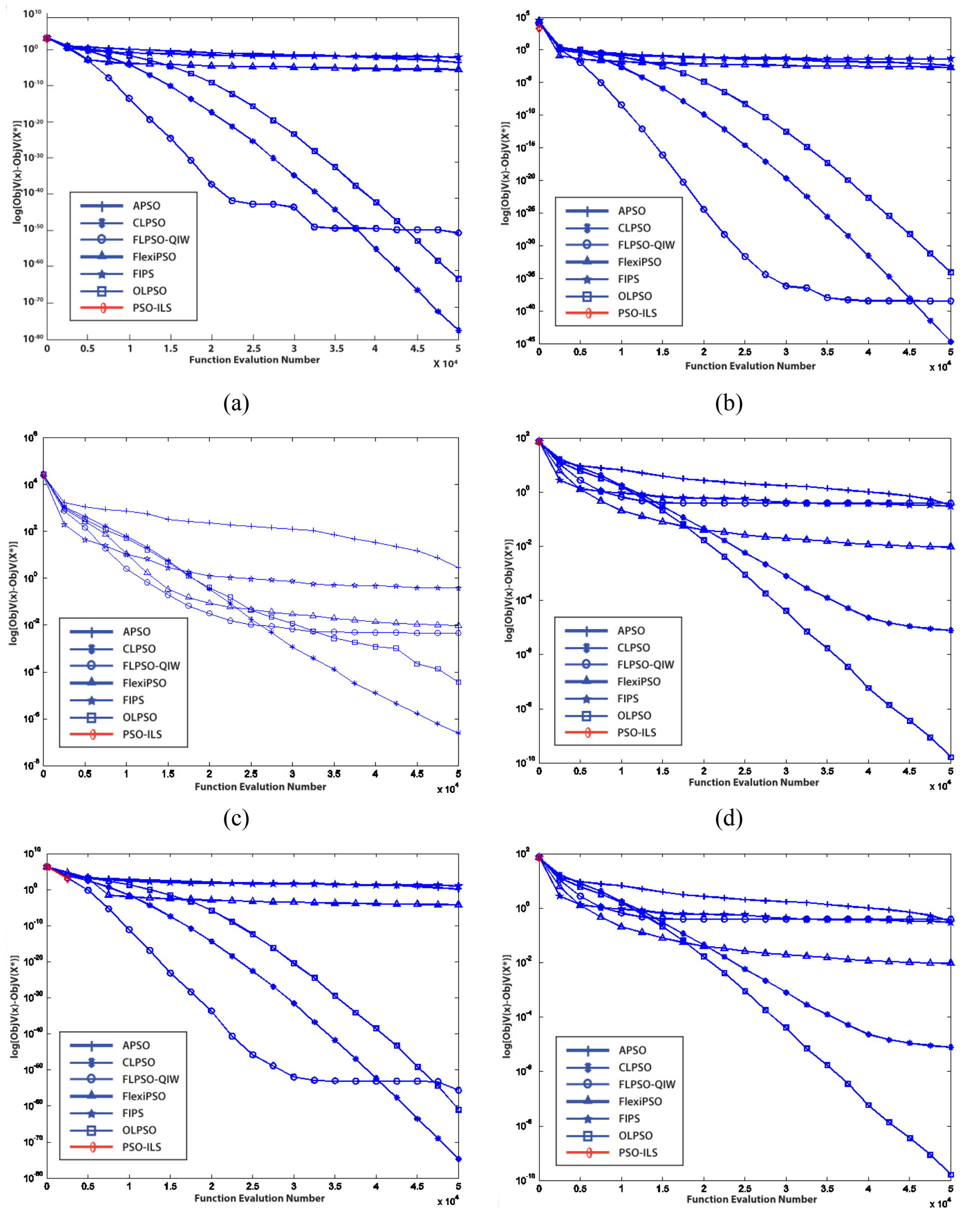
4.4.3 Comparison of the *SR* results

Table 8 demonstrates that PSO-ILS exhibits superior searching reliability, considering that this algorithm completely solves all the tested benchmarks with *SR* = 100%. Specifically, PSO-ILS is the only algorithm that successfully solves functions F3 and F8 within the predefined accuracy level in all the independent simulation runs. CLPSO, FLPSO-QIW and OLPSO also exhibit relatively robust search reliabilities because these PSO variants successfully solve some selected benchmarks (i.e., functions F1, F2, F4, F5, F9 and F10) with 100% success rate. In contrast, FlexiPSO has the worst search reliability because this algorithm produces *SR* = 0.00% in most of the benchmarks, i.e., five out of ten tested problems.

The competitive search reliabilities of the proposed PSO-ILS, CLPSO, FLPSO-QIW and OLPSO in solving the tested benchmarks imply that the strategy of deriving the exemplar from the non-fittest particles in the population is indeed viable to guide the PSO swarm towards the optimal regions of the search space. Among the four aforementioned PSO variants, the proposed PSO-ILS is considered to generate the most effective exemplars because this algorithm exhibits the most robust search reliability in solving all the tested benchmarks.

4.4.4 Comparison of the *SP* results

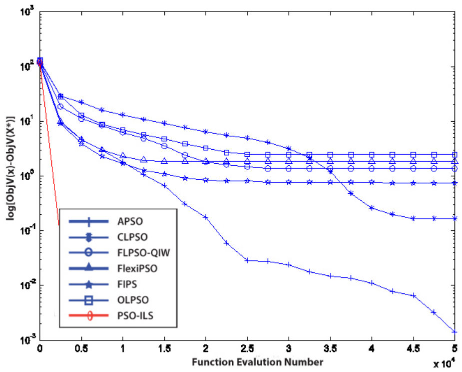
Obtaining the *SP* value is impossible if an algorithm never solves a particle problem (i.e., *SR* = 0%) because the *SP* value denotes the computational cost, i.e., the number of fitness evaluations (FEs), required by an algorithm to solve the problem with pre-specified ε. In this scenario, an infinity value "Inf" is assigned to the *SP* value, and only the convergence graphs are used to justify the algorithm's speed, as illustrated in Figure 3.



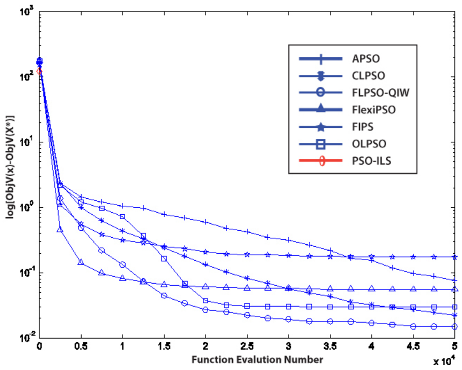
(f)

Figure 3: Convergence curves of the selected 10-*D* benchmark functions (a) F1, (b) F2, (c) F3, (d) F4, (e) F5, (f) F6, (g) F7, (h) F8, (i) F9 and (i) F10.

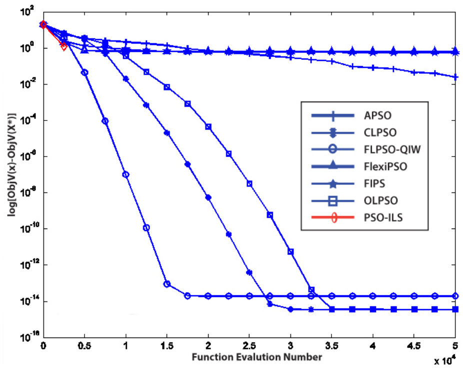
(*continue on next page*)



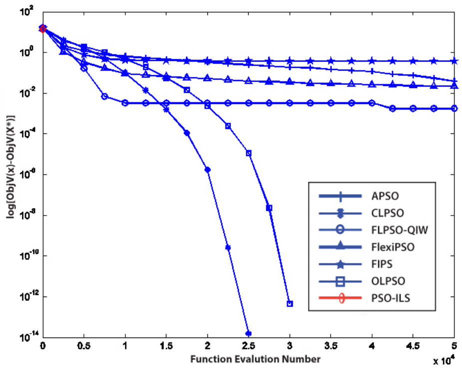
(g)



(h)



(i)



(j)

Figure 3: (*continued*)

In Table 8, we observe that PSO-ILS achieves the best *SP* values in all the tested benchmarks. This observation implies that our proposed algorithms require the least FEs to solve the given problems with acceptable ε. The excellent convergence characteristics exhibited by the PSO-ILS in solving all the tested problems are also illustrated by their respective convergence curves, as observed in Figure 3. Specifically, we observe a typical feature exhibited by the convergence curves of the PSO-ILS in all the tested problems, that is, a curve that sharply drops off at one point, usually during the early stage of the optimisation. This observation implies that the proposed PSO-ILS tends to exhibit faster convergence compared with the other algorithms, especially during the early stage of search process. However, the convergence graphs in Figure 3 reveal that most of the compared peers tend to stagnate at local optima during the early or middle stages of optimisation. This demerit prevents the compared peers from achieving promising solutions for the tested problems.

Although the proposed PSO-ILS, CLPSO, FLPSO-QIW, FIPS and OLPSO algorithms employ exemplars derived from the non-fittest particles to guide the search, the *SP* values produced by these algorithms are significantly different. Specifically, the *SP* values produces by PSO-ILS range from 10^2 to 10^3, whereas the other four compared peers have *SP* values ranging from 10^3 to 10^5. These observations indicate that the exemplars generated by the proposed ILS modules are more efficient in guiding the PSO swarm compared with CLPSO, FLPSO-QIW, FIPS and OLPSO. The rapid convergence characteristic of PSO-ILS enables the proposed algorithm to locate and exploit the optimal regions of the search space earlier than its peers. Thus, PSO-ILS has a greater opportunity to achieve higher quality solutions than the other algorithms in solving the tested benchmarks.

Table 8: Success rate and success performance results for 50-*D* problem.

		APSO	CLPSO	FLPSO-QIW	FlexiPSO	FIPS	OLPSO	PSO-ILS
F1	*SR*	13.33	**100.00**	**100.00**	0.00	60.00	**100.00**	**100.00**
	SP	2.38E+05	1.23E+04	6.88E+03	Inf	1.32E+04	1.75E+04	**6.49E+02**
F2	*SR*	6.67	**100.00**	**100.00**	0.00	50.00	**100.00**	**100.00**
	SP	3.36E+05	1.50E+04	8.17E+03	Inf	9.31E+03	2.15E+04	**6.75E+02**
F3	*SR*	0.00	93.33	10.00	0.00	60.00	73.33	**100.00**
	SP	Inf	4.57E+04	2.64E+05	Inf	2.05E+04	4.62E+04	**9.09E+02**
F4	*SR*	0.00	76.67	0.00	0.00	50.00	**100.00**	**100.00**
	SP	Inf	5.76E+04	Inf	Inf	1.59E+04	3.47E+04	**8.43E+02**
F5	*SR*	0.00	**100.00**	**100.00**	0.00	50.00	**100.00**	**100.00**
	SP	Inf	1.37E+04	7.65E+03	Inf	3.49E+04	1.96E+04	**1.42E+03**
F6	*SR*	93.33	86.67	43.33	**100.00**	50.00	33.33	**100.00**
	SP	4.10E+04	4.40E+04	5.00E+04	9.39E+03	1.83E+04	5.47E+04	**9.58E+02**
F7	*SR*	**100.00**	86.67	16.67	93.33	63.33	0.00	**100.00**
	SP	2.59E+04	4.40E+04	1.29E+05	1.12E+04	1.98E+04	Inf	**1.27E+03**
F8	*SR*	0.00	23.33	43.33	3.33	46.67	20.00	**100.00**
	SP	Inf	1.57E+05	6.05E+04	1.01E+06	1.40E+04	1.00E+05	**6.44E+02**
F9	*SR*	23.33	**100.00**	**100.00**	96.67	50.00	**100.00**	**100.00**
	SP	1.94E+05	1.04E+04	5.51E+03	1.29E+04	7.86E+03	1.44E+04	**9.11E+02**
F10	*SR*	20.00	**100.00**	96.67	3.33	30.00	**100.00**	**100.00**
	SP	2.11E 05	1.29E+04	7.21E+03	1.30E+06	1.99E+04	1.79E+04	**7.76E+02**

5. CONCLUSION

This paper presents an enhanced PSO algorithm called PSO-ILS. An innovative mechanism has been developed in the proposed ILS module to construct a more promising and efficient exemplar. This exemplar replaces the particle's self-cognitive and social components and is used to guide the particle's search direction. Based on the experimental results, it can be concluded that the proposed PSO-ILS significantly outperforms its peers in terms of search accuracy, reliability and efficiency. The results further suggest that the exemplar generated by the ILS module is more effective and efficient than that generated by the other PSO variants, which employ similar search mechanisms, i.e., CLPSO, FLPSO-QIW, FIPS and OLPSO.

6. ACKNOWLEDGEMENT

The authors express their sincere thanks to the anonymous reviewers for their significant contributions to the improvement of the final paper. This study was supported by a Universiti Sains Malaysia (USM) Research University Individual (RUI) grant entitled Development of an Intelligent Auto-Immune Disease Diagnostic System by Classification of Hep-2 Immunofluorescence Patterns (grant number 1001/PELECT/814205).

7. REFERENCES

1. Kennedy, J. & Eberhart, J. (1995). Particle swarm optimization. Proceedings of 1995 IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December.
2. Bank, A., Vincent, J. & Anyakoha, C. (2007). A review of particle swarm optimization. Part I: background and development. *Nat. Comput.*, 6, 467–484.
3. Banks, A., Vincent, J. & Anyakoha, C. (2008). A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Nat. Comput.*, 7, 109–124.
4. del Valle, Y. et al. (2008). Particle swarm optimization: Basic concepts, variants and applications in power systems. *IEEE Trans. Evol. Comput.*, 12, 171–195.
5. van den Bergh, F. & Engelbrecht, A. P. (2004). A cooperative approach to particle swarm optimization. *IEEE Trans. Evol. Comput.*, 8, 225–239.

6. Shi, Y. & Eberhart, R. (1998). A modified particle swarm optimizer. Proceedings of 1998 IEEE World Congress on Computational Intelligence, Anchorage, Alaska, 4–9 May.
7. Clerc, M. & Kennedy. J. (2002). The particle swarm: Explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.*, 6, 58–73.
8. Ratnaweera, A., Halgamuge, S. K. & Watson, H. C. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans. Evol. Comput.*, 8, 240–255.
9. Juang, Y. T., Tung, S. L. & Chiu, H. C. (2011). Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions. *Inform. Sci.*, 181, 4539–4549.
10. Zhan, Z. H. et al. (2009). Adaptive particle swarm optimization. *IEEE Trans. Syst., Man. Cybern. B: Cybernet.*, 39, 1362–1381.
11. Leu, M. S. & Yeh, M. F. (2012). Grey particle swarm optimization. *Appl. Soft. Comput.*, 12, 2985–2996.
12. Kennedy, J. (1999). Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. Proceedings of 1999 IEEE Congress on Evolutionary Computation (CEC '99), Washington, 6–9 July.
13. Mendes, R., Kennedy, J. & Neves, J. (2004). The fully informed particle swarm: Simpler, maybe better. *IEEE Trans. Evol. Comput.*, 8, 204–210.
14. Kathrada, M. (2009). The flexi-PSO: Towards a more flexible particle swarm optimizer. *OPSEARCH*, 46, 52–68.
15. de Oca, M. et al. (2009). Frankenstein's PSO: A composite particle swarm optimization algorithm. *IEEE Trans. Evol. Comput.*, 13, 1120–1132.
16. Marinakis, Y. & Marinaki, M. (2013). A hybridized particle swarm optimization with expanding neighborhood topology for the feature selection problem. In: Blesa, M. et al. (Eds.). *Hybrid metaheuristics*. Berlin: Springer Berlin Heidelberg, 37–51.
17. Liang, J. J. et al. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.*, 10, 281–295.
18. Tang, Y., Wang, Z. & Fang, J. A. (2011). Feedback learning particle swarm optimization. *Appl. Soft. Comput.*, 11, 4713–4725.
19. Nasir, M. et al. (2012). A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization. *Inform. Sci.*, 209, 16–36.
20. Huang, H. et al. (2012), Example-based learning particle swarm optimization for continuous optimization. *Inform. Sci.*, 182, 125–138.

21. Zhan, Z. H. et al. (2011). Orthogonal learning particle swarm optimization. *IEEE Trans. Evol. Comput.*, 15, 832–847.
22. Zhou, D. et al. (2011). Randomization in particle swarm optimization for global search ability. *Expert Syst. Appl.*, 38, 15356–15364.
23. Derrac, J. et al. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.*, 1, 3–18.