



ELSEVIER

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Particle swarm optimization with increasing topology connectivity



Wei Hong Lim, Nor Ashidi Mat Isa*

Imaging and Intelligent System Research Team (ISRT), School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia

ARTICLE INFO

Article history:

Received 15 May 2013

Received in revised form

18 September 2013

Accepted 20 September 2013

Available online 17 October 2013

Keywords:

Particle swarm optimization (PSO)

Increasing topology connectivity (ITC)

Metaheuristic search (MS)

Neighborhood search (NS) operator

ABSTRACT

In this paper, we propose a new variant of particle swarm optimization (PSO), namely PSO with increasing topology connectivity (PSO-ITC), to solve unconstrained single-objective optimization problems with continuous search space. Specifically, an ITC module is developed to achieve better control of exploration/exploitation searches by linearly increasing the particle's topology connectivity with time as well as performing the shuffling mechanism. Furthermore, we introduce a new learning framework that consists of a new velocity update mechanism and a new neighborhood search operator that aims to enhance the algorithm's searching performance. The proposed PSO-ITC is extensively evaluated across 20 benchmark functions with various features as well as two engineering design problems. Simulation results reveal that the performance of the PSO-ITC is superior to nine other PSO variants and six cutting-edge metaheuristic search algorithms.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Particle swarm optimization (PSO) is a metaheuristic search (MS) algorithm proposed by Kennedy and Eberhart (1995). It is inspired by the collaborative behavior of a school of fish or a flock of birds in search for food (Banks et al., 2007; del Valle et al., 2008; Eberhart and Shi, 2001; Kennedy and Eberhart, 1995). In PSO, each individual (i.e., particle) represents the potential solution to the optimization problem, whereas the location of the food source represents the global optimum solution. During the searching process, all these particles share information and collaborate with each other. This collaboration enables the population to move toward the food source from different directions, thereby leading to swarm convergence (Banks et al., 2007; Eberhart and Shi, 2001). PSO is characterized by conceptual simplicity and high efficiency. Thus, it has received increasing attention and has been widely applied to solve a large class of engineering design problems such as power system design (AlRashidi and El-Hawary, 2009; Chen et al., 2007; del Valle et al., 2008; dos Santos Coelho and Mariani, 2008; Neyestani et al., 2010; Wang et al., 2013), trajectory planning (Alonso Zotes and Santos Peñas, 2012; Fu et al., 2012; Marinakis et al., 2010), artificial neural network (ANN) training (Gudise and Venayagamoorthy, 2003; Mirjalili et al., 2012; Yaghini et al., 2013), data mining (Holden and Freitas, 2008; Özbakır and Delice, 2011; Sarath and Ravi; Wang et al., 2007), data clustering (Kiranyaz et al.,

2010; Shih, 2006; Sun et al., 2012; Van Der Merwe and Engelbrecht, 2003; Yang et al., 2009), parameter estimation and system identification (Liu et al., 2008; Modares et al., 2010; Sakthivel et al., 2010), and many other engineering problems (Banks et al., 2008; Huang et al., 2009; Lin et al., 2009; Paoli et al., 2009; Sharma et al., 2009; Wachowiak et al., 2004).

However, despite its competitive performance, PSO possesses certain undesirable dynamic characteristics that restrict its searching capability. Previous studies (van den Bergh and Engelbrecht, 2004) revealed that PSO is beset by a premature convergence issue, as the particles tend to be trapped in the local optima solution, which is attributed to the rapid convergence of PSO and the loss of swarm diversity. Another major concern in the application of PSO is the algorithm's capability in balancing exploration/exploitation searches. Excessive exploration or exploitation searches are undesirable, as the former prevents swarm convergence, whereas the latter has a high tendency to cause premature swarm convergence (Shi and Eberhart, 1998).

Although various approaches (Banks et al., 2007, 2008; del Valle et al., 2008) have been reported to alleviate the drawbacks of PSO, little effort has been made to address the issue of balancing the PSO's exploration/exploitation searches by varying the particle's topology connectivity with time. Furthermore, the most existing PSO variants do not provide an alternative strategy to the particles when they fail to improve their fitness during the searching process. This lack of an alternative strategy inevitably limits the algorithm's performance. Motivated by these facts, we propose a new PSO variant, namely the PSO with increasing topology connectivity (PSO-ITC). The key benefit of the PSO-ITC is the synergy of a novel ITC module into the PSO, in offering better

* Corresponding author. Tel.: +60 45996093; fax: +60 45941023.

E-mail addresses: limweihong87@yahoo.com (W.H. Lim), ashidi@eng.usm.my (N.A. Mat Isa).

control of the PSO's exploration/exploitation searches. Specifically, the ITC module linearly increases the topology connectivity of each particle with time as the lower topology connectivity encourages the particle to perform exploration, whereas the higher topology connectivity favors exploitation (Kennedy, 1999; Kennedy and Mendes, 2002). A shuffling mechanism is also introduced into the ITC module to prevent swarm stagnation. We also develop a learning framework that consists of a new velocity update mechanism and a new neighborhood search (NS) operator to further enhance the searching performance of PSO-ITC. The NS operator is triggered only if the PSO-ITC particle fails to improve its personal best fitness when it is evolved through the new velocity update mechanism.

The remainder of this paper is organized as follows: Section 2 briefly discusses some related works. Section 3 provides novel insights into the workings of the proposed PSO-ITC. Section 4 presents the experimental setting and results. Section 5 concludes the study.

2. Related works

For a more or less comprehensive survey, we briefly discuss the mechanism of the basic PSO (BPSO), after which we review several well-established PSO variants.

2.1. Basic PSO

For a D -dimensional problem hyperspace, each BPSO particle i represents a potential solution to a problem, and its current state is associated with two vectors, namely the position vector $X_i = [X_{i1}, X_{i2}, \dots, X_{iD}]$ and the velocity vector $V_i = [V_{i1}, V_{i2}, \dots, V_{iD}]$. One unique feature of BPSO that differentiates it from other metaheuristic search (MS) algorithms is the ability of particle i to memorize the best position that it has ever achieved, that is, its personal best experience $P_i = [P_{i1}, P_{i2}, \dots, P_{iD}]$. During the searching process, each particle in the population stochastically adjusts its trajectory according to its personal best experience P_i and to the group best experience found by all the particles so far, $P_g = [P_{g1}, P_{g2}, \dots, P_{gD}]$ (Eberhart and Shi, 2001; Kennedy and Eberhart, 1995). Specifically, for each particular d th dimension of particle i , its velocity $V_{i,d}(t+1)$ and position $X_{i,d}(t+1)$ at $(t+1)$ th iteration of the searching process 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 $i = 1, 2, \dots, S$ is the particle's index; S is the population size; c_1 and c_2 are the acceleration coefficients that control the influences of 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 (Shi and Eberhart, 1998).

2.2. PSO variants and improvements

Various approaches have been proposed to improve the performance of PSO, among which parameter adaptation strategy has become one of the more promising. Clerc and Kennedy (2002) incorporated a constriction factor χ into the PSO to address the swarm explosion issue. Ratnaweera et al. (2004) developed a time-varying acceleration coefficient (TVAC) strategy, where the c_1 and c_2 are dynamically changed with time to better regulate the exploration/exploitation searches. To this end, two variants of PSO-TVAC, namely the PSO-TVAC with mutation (MPSO-TVAC)

and the self-organizing hierarchical PSO-TVAC (HPSO-TVAC), have been proposed. Zhan et al. (2009) developed an adaptive PSO (APSO) capable of identifying the swarm's evolutionary states through the proposed evolutionary state estimation (ESE) module. The outputs of the ESE module are used to adaptively adjust the particles' ω , c_1 , and c_2 . Leu and Yeh (2012) employed the gray relational analysis in their gray PSO to tune the particles' ω , c_1 , and c_2 . Based on the searching feedback status obtained from the population manager, Hsieh et al. (2009) developed an efficient population utilization strategy for PSO (EPUS-PSO) to adaptively adjust the population size.

Population topology also plays a major role in PSO's performance as it decides the information flow rate of the best solution within the swarm (Kennedy, 1999; Kennedy and Mendes, 2002). Mendes et al. (2004) proposed a fully connected PSO (FIPSO) by advocating that each particle's movement is influenced by all its topological neighbors. Kathrada (2009) combined the global and the local version of PSO, and proposed a flexible PSO (FlexiPSO). A simple heuristic is developed in the FlexiPSO to increase the flexibility of the swarm to search across various types of fitness landscape. Inspired by the social behavior of the clan, Carvalho and Bastos-Filho (2008) proposed a clan PSO, where the population is divided into several clans. Each clan first performs the search, and the particle with the best fitness is selected as the clan leader. A conference is then performed among the leaders to adjust their position. Bastos-Filho et al. (2009) and Pontes et al. (2011) further improved the clan PSO by hybridizing it with the migration mechanism and APSO, respectively. To alleviate the deficiencies of fixed neighborhoods, Liang and Suganthan (2005) proposed a dynamic multi-swarm PSO (DMS-PSO) with a dynamically changing neighborhood structure. Montes de Oca et al. (2009) adopted the concept of time-varying population topology into their Frankenstein PSO (FPSO). Initially, the particles in FPSO are connected with a fully connected topology. The topology is then decreased over time and eventually reduced into the ring topology. Marinakis and Marinaki (2013) proposed a PSO with an expanding neighborhood topology (PSOENT) by hybridizing the PSO with the variable neighborhood search strategy. In their approaches, the particle's neighborhood expands based on the quality of the produced solutions. The PSOENT was applied to solve eight feature selection problems and successfully achieved an overall average classification accuracy of 92.48%.

Another area of research is the exploration of PSO's learning strategies. Liang et al. (2006) developed a comprehensive learning PSO (CLPSO). Accordingly, each particle is allowed to learn either from its P_i or from other particle's historical best position in each dimension. Tang et al. (2011) further improved the CLPSO by introducing the feedback learning PSO with quadratic inertia weight (FLPSO-QIW). Unlike the CLPSO, their FLPSO-QIW generates potential exemplars from the first 50% fittest particles. Moreover, the learning probability of FLPSO-QIW particles is assigned according to the particle's fitness instead of the particle's index. Inspired by the DMS-PSO and the CLPSO, Nasir et al. (2012) proposed a dynamic neighborhood learning-based PSO (DNLPSO). In their approach, the particle's exemplar is selected from a neighborhood, which is made dynamic in nature. Huang et al. (2012) proposed an example-based learning PSO (ELPSO). Instead of a single P_g particle, an example set of multiple global best particles is employed to update the particle's velocity in ELPSO. Zhou et al. (2011) introduced the random position PSO (RPPSO), where a random particle is used to guide the swarm if a randomly generated number is smaller than the proposed probability $P(\Delta f)$. Zhan et al. (2011) proposed the orthogonal learning PSO (OLPSO) that employs the orthogonal experimental design (OED) (Hicks, 1993) to construct an effective exemplar to guide the search. OLPSO with the local topology (OLPSO-L) reportedly outperforms

its counterpart with the global topology (OLPSO-G). Another OED-based PSO variant is the orthogonal PSO (OPSO) introduced by Ho et al. (2008). Accordingly, an intelligent move mechanism is designed to predict the particle's next position. Numerical results show that OPSO can find a better solution to the 12 selected benchmark functions and one task assignment problem.

Mariani et al. (2012) combined the chaotic Zaslavskii map (Zaslavsky, 1978) with the quantum PSO (QPSO) (Sun et al., 2004) to solve heat exchanger optimization problems. Sun et al. (2012) developed a new clustering scheme based on a QPSO variant, namely the multi-elitist QPSO (MEQPSO). The MEQPSO-based clustering algorithm was applied in the gene expression data analysis for discovering the function of a gene. Yaghini et al. (2013) proposed a hybrid improved opposition-based PSO and a backpropagation algorithm with a momentum term to produce an efficient ANN training algorithm. Accordingly, opposition-based learning and random perturbations help the algorithm to maintain the population diversity. The simulation results revealed that the training time and the accuracy of the proposed algorithm are superior to those of the three other well-known ANN training algorithms. Sun et al. (2011) explored the applicability of QPSO to combinatory optimization problems. Combining the QPSO with the loop deletion operation produced a modified QPSO that was used to solve the quality-of-service (QoS) multicast routing problem. To address the vehicle routing problem, Marinakis et al. (2010) introduced hybrid PSO (HybPSO) by incorporating the multiple phase neighborhood search-greedy randomized search procedure (MPNS-GRASP) algorithm, the expanding neighborhood search strategy, and a path relinking strategy into the PSO. Extensive experimental analyses show that the HybPSO is very promising in solving very large-scale vehicle routing problems within a short computational time.

To efficiently solve multi-objective (MO) problems, Mousa et al. (2012) combined the PSO and the genetic algorithm to form a hybrid MO evolutionary algorithm. More specifically, the proposed method employs a local search scheme to explore less crowded areas in the current archive to obtain more non-dominated solutions. Wang et al. (2013) developed a novel multi-objective PSO (MOPSO) variant, namely dynamic neighborhood small population PSO (DNSPPSO), based on the regeneration and dynamic neighborhood strategies. The former strategy improves the algorithm's convergence speed, whereas the latter converts MO problems into single-objective (SO) ones by sorting and evaluating the objectives one by one. DNSPPSO is successfully applied as an intelligent dynamic reconfiguration strategy to prevent power failure in an electric ship's power system. Motivated by the fact that the algorithm's convergence characteristics towards the Pareto-optima set is significantly dependent on the proper selection of local guides, Sahoo et al. (2011) proposed a heuristics-based selection of guides in MOPSO (HSG-MOPSO). Accordingly, the HSG-MOPSO consists of two types of local guides, namely non-dominated and dominated guides. During optimization, a certain number of PSO members follow their nearest non-dominated guides, while the remaining ones are guided by the nearest dominated solutions. Sahoo et al. (2012) employed the principles of fuzzy Pareto-dominance into the MOPSO (FPD-MOPSO) to efficiently discover and rank non-dominated solutions on the Pareto-approximation front. The proposed strategy is capable of simultaneously maintaining the quality and the diversity of the solutions retained in the elite archive. Both HSG-MOPSO and FPD-MOPSO are used as multi-objective planning algorithms for electrical distribution systems. Wang and Singh (2009) proposed an improved MOPSO to solve multi-area economic dispatch problems by combining MOPSO with the synchronous particle local search strategy (Liu et al., 2007). This hybridization strategy can preserve distribution diversity and uniformity, and speed up the search process. Omkar et al. (2012) developed a novel parallel

approach to vector evaluated PSO (VEPSO) (Parsopoulos and Vrahatis, 2002) in solving the multi-objective design of composite structures. Extensive simulations revealed that the parallel implementation of VEPSO outperforms the one with the serial implementation.

3. PSO with increasing topology connectivity (PSO-ITC)

In this section, we describe in detail the ITC module employed in the proposed PSO-ITC. Next, we present the methodologies employed in our proposed learning framework. Finally, we provide the complete framework of PSO-ITC.

3.1. ITC module

The ITC module is one of the key factors that determine PSO-ITC's performance by dynamically changing the particle's topology connectivity during the searching process. Specifically, the ITC module aims to better control exploration/exploitation searches by linearly increasing the particle's topology connectivity with time as well as performing the shuffling mechanism. The mechanism of the ITC module is inspired by early studies performed as follows. PSO with larger topology connectivity favors the simple problem, whereas the smaller connectivity counterpart performs better in complex problems (Kennedy, 1999; Kennedy and Mendes, 2002). This finding implies that the former PSO variant is more exploitative, whereas the latter is more explorative. Shi and Eberhart (1998) advocated that PSO particles in the early stage of optimization require higher diversity to wander around the full range of the search space, thus emphasizing the exploration search. At the latter stage of the search, fine-tuning of the solution becomes the priority, and thus, the exploitation search is required.

The ITC module works as follows: initially, each particle in the PSO-ITC is connected with one neighbor that is randomly selected from the population. As the optimization process evolves, the ITC module gradually increases the particle's topology connectivity until all particles are fully connected. Mathematically, each particle's topology connectivity is linearly increased as follows:

$$TC_i = \lfloor TC_{\min} + TC_{\max}[(k-1)/(FE_{\max} - 1)] \rfloor \quad (3)$$

where TC_i is the current connectivity of particle i ; TC_{\min} and TC_{\max} represent the maximum and minimum connectivity, respectively, where $TC_{\min} = 1$ and $TC_{\max} = S - 1$; k is the current fitness evaluations (FEs) consumed by the algorithm; FE_{\max} is the maximum FEs allocated; and $\lfloor \rfloor$ represents the floor operator.

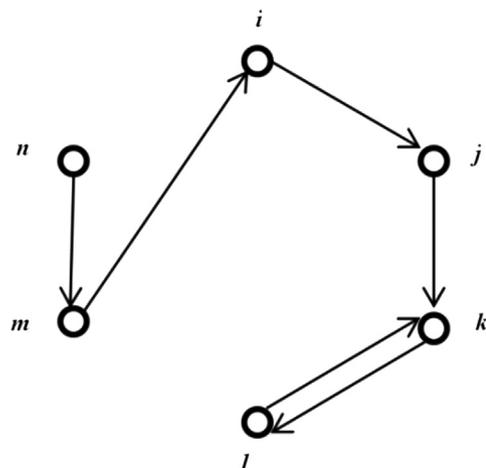


Fig. 1. Possible topology connectivity of each particle when TC for all particles is equal to 1.

Each time the TC_i of particle i is increased by ΔTC_i , particle i randomly selects ΔTC_i new neighbors from the population. Unlike the neighborhood structure shown in the previously proposed DMS-PSO (Liang and Suganthan, 2005) and DNLPSO (Nasir et al., 2012), the particles in our PSO-ITC are not connected in a bidirectional manner. For example, in the case of $TC_i=1$, if particle i has selected particle j as its neighbor, then particle j does not necessarily select particle i as its neighbor as well. Instead, particle j may select another particle, for example, particle k , as its neighbor. Fig. 1 illustrates the aforementioned scenario.

Aside from linearly increasing the particle's topology connectivity with time, the ITC module also incorporates a shuffling mechanism to alleviate swarm stagnation. As shown in Fig. 2, if particle i fails to improve the fitness of the group best experience found by all the particles so far (P_g), that is, $f(P_g)$ for z successive FEs, the ITC module reassigns new neighbors to particle i by randomly selecting TC_i members from the population. This mechanism provides the new topological information for particle i , thereby allowing it to perform the search in a new direction provided by the new neighborhood members. We also perform perturbation on the P_g particle to help it escape from the local optimum. Specifically, one d th dimension of the P_g particle, that is, $P_{g,d}$, is randomly selected and perturbed as follows:

$$P_{g,d}^{per} = r_3 P_{g,d} + (1 - r_3)(P_{x,d} - P_{y,d}) \quad (4)$$

where $P_{g,d}^{per}$ is the perturbed $P_{g,d}$; r_3 is a random number with the range of $[0, 1]$; $P_{x,d}$ and $P_{y,d}$ are the personal best positions of the two particles randomly selected from the population. The perturbed P_g particle ($P_{g,d}^{per}$) replaces the current P_g if the former has better (i.e., lower) fitness than the latter. Eq. (4) is in fact inspired by the social learning strategy proposed by Montes de Oca et al. (2011). As shown in Eq. (4), the $P_{g,d}^{per}$ particle learns socially from a subset of more experienced particles (i.e., P_g , P_x , and P_y particles), without incurring the cost of acquiring that knowledge individually from scratch. Consequently, the $P_{g,d}^{per}$ particle produced is biased toward the best particle. This approach lets the $P_{g,d}^{per}$ particle jump to promising regions of the search space instead of to inferior ones.

Fig. 3 illustrates the mechanism of the proposed ITC module. In the initial stage, each particle in the population randomly selects one population member as its neighbor. For example, particles i and n select particles j and m as their neighbors, respectively. After a certain number of FEs, the connectivity of all particles is increased from one to two, and each particle of the population randomly selects another population member as its neighbor. For example, particle i selects particle k as its new neighbor when its topology connectivity is increased to two. Thus, particle i 's neighborhood, that is, a set of particles connected with particle i , now consists of two members, namely particles j and k . Next, particle i performs the shuffling mechanism as it fails to improve

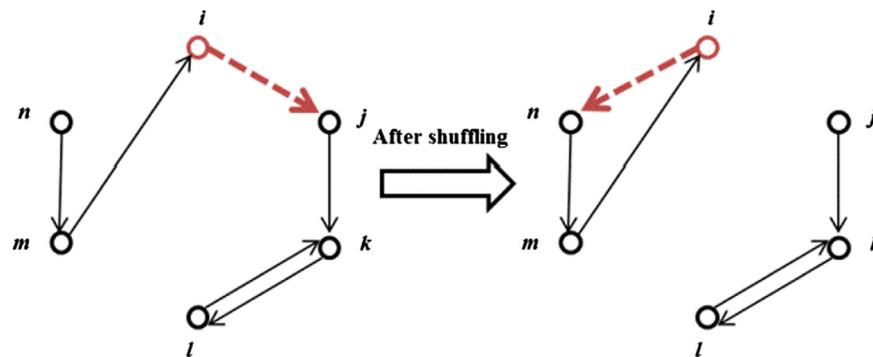


Fig. 2. Shuffling mechanism is performed on particle i when it fails to improve P_g fitness for m successive FEs. Note that the topology connectivity for other particles remains the same.

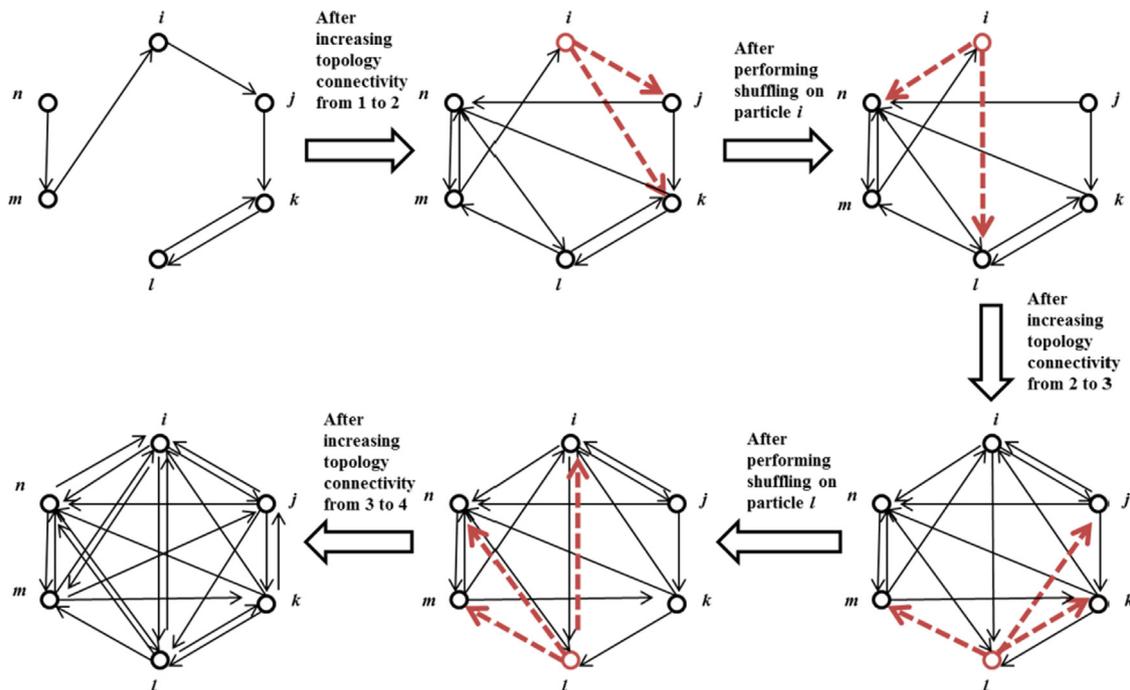


Fig. 3. Graphical illustration of the mechanism in ITC module.

the global best fitness for z successive FEs. Specifically, particle i gives up its original neighbors, that is, particles j and k , and randomly selects particles l and n as its new neighbors. The new neighbors (i.e., particles l and n) provide particle i a new searching direction, thereby preventing the latter from stagnating in the local optima. Both steps—(1) linearly increasing topology connectivity and (2) shuffling mechanisms—are repeated until all particles in the population are fully connected.

Fig. 4 illustrates the implementation of the ITC module. As shown in the figure, a variable fc_i is defined to record the number of successive FEs where particle i fails to improve the $f(P_g)$. As shown in the main algorithm block (i.e., Fig. 9), the fc_i variable is reset to zero if the global best particle's fitness is successfully improved. Otherwise, the value of fc_i is incrementally increased by one. Also, the two exemplars that play major roles in evolving particle i through our proposed learning framework (i.e., the cognitive exemplar $c_{exp,i}$ and the social exemplar $s_{exp,i}$) are updated. Both $c_{exp,i}$ and $s_{exp,i}$ exemplars are generated by the Generate_Exemplars procedure described in Section 3.2.1.

3.2. Proposed learning framework

In this section, we introduce the new learning framework adopted by the proposed PSO-ITC. We first explain the methodology employed to generate the $c_{exp,i}$ and $s_{exp,i}$ exemplars that are essential in guiding particle i in the proposed learning framework. Next, we provide a detailed description of the new velocity update mechanism and the new NS operator employed in the PSO-ITC.

3.2.1. Derivation of the cognitive exemplar ($c_{exp,i}$) and the social exemplar ($s_{exp,i}$)

From Eq. (1), we observe that the second component of $(P_{i,d} - X_{i,d})$ acts as the cognitive component, whereas the third component of $(P_{g,d} - X_{i,d})$ represents the social component. It is worth mentioning that the fitness of the global best particle P_g is always

not worse (i.e. not higher) than the personal best position of particle i , P_i .

Inspired by this observation, we propose to generate two exemplars, the $c_{exp,i}$ and the $s_{exp,i}$ exemplars, from the neighborhood of particle i (i.e., $neighbor_i$) to guide particle i in the searching process. Specifically, we first sort the neighborhood members of particle i , including particle i itself, according to their personal best fitness. The fitter neighborhood members with their personal best fitness ranked at the first quartile range (i.e., $neighbor_upper_i$) are used to generate the $s_{exp,i}$ exemplar, while the members in the remaining three quartiles (i.e., $neighbor_lower_i$) produce the $c_{exp,i}$ exemplar. As the $c_{exp,i}$ exemplar is derived from three-fourth of the neighbors with the worst fitness, it may have an inferior fitness to particle i 's personal best fitness. Nevertheless, as explained in the following subsection, using the individuals with worse fitness than that of the particle itself makes sense, because they are used to repel the particle. In addition, unlike the CLPSO (Liang et al., 2006) and the DNLPSO (Nasir et al., 2012) that employed tournament selection, we utilize roulette wheel selection to generate the $c_{exp,i}$ and the $s_{exp,i}$ exemplars from $neighbor_lower_i$ and $neighbor_upper_i$, respectively. Specifically, each member k in the $neighbor_lower_i$ and $neighbor_upper_i$ is assigned a weightage value, W_k

$$W_k = \frac{f_{max} - f(P_k)}{f_{max} - f_{min}}, \quad \forall k \in [1, K] \quad (5)$$

where f_{max} and f_{min} represent the maximum (i.e., worst) and the minimum (i.e., best) personal best fitness values of the members in $neighbor_lower_i$ or $neighbor_upper_i$, respectively, and K represents the number of members in $neighbor_lower_i$ or $neighbor_upper_i$. From Eq. (5), the neighborhood member k with a lower fitness is assigned a larger W_k value, which implies that it has a greater probability of being selected as the $c_{exp,i}$ or $s_{exp,i}$ exemplar. To prevent the derivation of the $c_{exp,i}$ and $s_{exp,i}$ exemplars solely from the fittest members of the $neighbor_lower_i$ and $neighbor_upper_i$, respectively, we randomly select one dimension d_r of the $c_{exp,i}$ and $s_{exp,i}$ exemplars, $c_{exp,i}(d_r)$ and $s_{exp,i}(d_r)$. The

Algorithm 1: ITC_module($fc_i, P_g, f(P_g), k, TC_i, neighbor_i$)	
Input: Failure counter (fc_i), global best particle's position (P_g), global best particle's fitness ($f(P_g)$), number of fitness evaluation consumed (k), particle i 's topology connectivity (TC_i), neighborhood member of particle i ($neighbor_i$)	
1:	$TC_{old} := TC_i$;
2:	Calculate current TC_i by using Eq. (3);
3:	if current $TC_i \neq TC_{old}$ then
4:	/*increase the particle 's topology connectivity*/
5:	$\Delta TC_i := TC_i - TC_{old}$;
6:	Randomly select ΔTC_i members from the population and update the members of particle i 's neighborhood, i.e., $neighbor_i$;
7:	$[c_{exp,i} \ s_{exp,i}] := \text{Generate_Exemplars}(neighbor_i)$;
8:	Perform fitness evaluation on $c_{exp,i}$ and $s_{exp,i}$;
9:	Update the P_g and $f(P_g)$ if $c_{exp,i}$ or $s_{exp,i}$ has better fitness;
10:	$k := k + 2$;
11:	else
12:	if $fc_i > z$ then
13:	/*activate the shuffling mechanism*/
14:	Randomly select TC_i particles as new neighbors and update the $neighbor_i$;
15:	Perform perturbation on P_g by using Eq. (4);
16:	Update P_g and $f(P_g)$ if P_g^{per} has better fitness;
17:	$k := k + 1$;
18:	$[c_{exp,i} \ s_{exp,i}] := \text{Generate_Exemplars}(neighbor_i)$;
19:	Perform fitness evaluation on $c_{exp,i}$ and $s_{exp,i}$;
20:	Update the P_g and $f(P_g)$ if $c_{exp,i}$ or $s_{exp,i}$ has better fitness;
21:	$k := k + 2$;
22:	end if
23:	end if
Output: Cognitive and social exemplars ($c_{exp,i}$ and $s_{exp,i}$), updated P_g , $f(P_g)$, TC_i , $neighbor_i$, and k	

Fig. 4. ITC module of the PSO-ITC algorithm.

former is replaced with the d_r -th component of the P_i particle itself, $P_i(d_r)$, whereas the latter is derived from the d_r -th component of a randomly selected particle from $neighbor_upper_i$. The procedure that generates the $c_{exp,i}$ and $s_{exp,i}$ exemplars for particle i is illustrated in Fig. 5.

3.2.2. Proposed velocity update mechanism

In our proposed learning framework, the velocity of particle i is updated according to its $c_{exp,i}$ exemplar and the P_g particle. As each component of the $c_{exp,i}$ exemplar is selected through a probabilistic mechanism, two possible scenarios may be encountered: (1) the $c_{exp,i}$ exemplar has better (i.e., lower) fitness than particle i , that is, $f(c_{exp,i}) < f(P_i)$, and (2) the $c_{exp,i}$ exemplar has worse (i.e., higher) fitness than particle i , that is, $f(c_{exp,i}) \geq f(P_i)$. Thus, two velocity update mechanisms are employed in response to these two scenarios. For scenario (1), particle i is encouraged to attract towards the $c_{exp,i}$ exemplar as the latter has better fitness, thereby offering particle i a more promising search direction. For scenario (2),

as the $c_{exp,i}$ exemplar is unlikely to contribute to particle i 's fitness improvement, the latter is thus repelled from the former to let particle i search the unexplored regions of the search space. Mathematically, particle i 's velocity V_i is updated as follows:

$$V_i = \begin{cases} \omega V_i + c_1 r_4 (c_{exp,i} - X_i) + c_2 r_5 (P_g - X_i), & f(c_{exp,i}) < f(P_i) \\ \omega V_i - c_1 r_6 (c_{exp,i} - X_i) + c_2 r_7 (P_g - X_i) & \text{otherwise} \end{cases} \quad (6)$$

where r_4, r_5, r_6 , and r_7 are the random numbers in the range of $[0, 1]$.

The new position of particle i is updated by using Eq. (2). It is then evaluated and compared with the fitness of P_i and P_g . If the former has better fitness than the latter, the latter's fitness and position are replaced by the former. When particle i successfully improves its personal best fitness, it may have some useful information about certain components of the newly improved P_i position. Thus, when particle i successfully improves its P_i position and if the improved P_i is different from P_g , an elitist-based learning strategy (EBLS) is employed to extract the useful information from

Algorithm 2: Generate Exemplars($neighbor_i$)	
Input: Neighborhood members of particle i ($neighbor_i$)	
1:	Sort the members in $neighbor_i$ according to their personal best fitness;
2:	Assign the members with their personal best fitness in the first quartile range into the $neighbor_upper_i$;
3:	Assign the remaining members with higher personal best fitness into the $neighbor_lower_i$;
4:	<i>/*generate the $s_{exp,i}$ exemplar*/</i>
5:	for each member k in $neighbor_upper_i$ do
6:	Calculate W_k for each $neighbor_upper_i$ member k by using Eq. (5);
7:	end for
8:	Randomly select a dimension, d ;
9:	for each dimension d do
10:	if $d \neq d_r$ then
11:	Perform the roulette wheel selection based on the W_k of each member of $neighbor_upper_i$;
12:	$s_{exp,i}(d) := d$ -th component of the selected member from $neighbor_upper_i$;
13:	else <i>/*$d = d_r$*/</i>
14:	Randomly select one member from the $neighbor_upper_i$;
15:	$s_{exp,i}(d_r) := d_r$ -th component of the randomly selected member from $neighbor_upper_i$;
16:	end if
17:	end for
18:	<i>/*generate the $c_{exp,i}$ exemplar*/</i>
19:	for each member k in $neighbor_lower_i$ do
20:	Calculate W_k for each $neighbor_lower_i$ member k using Eq. (5);
21:	end for
22:	for each dimension d do
23:	if $d \neq d_r$ then
24:	Perform the roulette wheel selection based on the W_k of each member of $neighbor_lower_i$;
25:	$c_{exp,i}(d) := d$ -th component of the selected member from $neighbor_lower_i$;
26:	else <i>/*$d = d_r$*/</i>
27:	Randomly select one member from the $neighbor_lower_i$;
28:	$c_{exp,i}(d_r) := d_r$ -th component of the randomly selected member from $neighbor_lower_i$;
29:	end if
30:	end for
Output: Cognitive and social exemplars ($c_{exp,i}$ and $s_{exp,i}$)	

Fig. 5. Derivation of the $c_{exp,i}$ and $s_{exp,i}$ exemplars in the PSO-ITC algorithm.

Algorithm 3: EBLS($P_i, P_g, f(P_g), k$)	
Input: Particle i 's personal best position (P_i), global best particle's position (P_g), global best particle's fitness ($f(P_g)$), number of fitness evaluation consumed (k)	
1:	for each dimension d do
2:	$P_{g,temp} := P_g$ and $f(P_{g,temp}) := f(P_g)$; <i>//where $P_{g,temp}$ is a temporary particle</i>
3:	$P_{g,temp}(d) := P_i(d)$;
4:	Perform fitness evaluation on the updated $P_{g,temp}$;
5:	if $f(P_{g,temp}) < f(P_g)$ then
6:	$P_g(d) := P_{g,temp}(d)$;
7:	end if
8:	$k := k + 1$;
9:	end for
Output: Updated $P_g, f(P_g)$, and k	

Fig. 6. EBLS of the PSO-ITC.

the newly improved P_i to further improve the P_g particle. Specifically, when particle i successfully finds a better P_i , the EBLs will iteratively check each dimension of P_g by replacing the dimension with the corresponding dimensional value of P_i if P_g is improved by doing so. This mechanism enables the P_g to learn useful information from dimensions of a P_i that have been improved, thereby improving the algorithm's convergence speed. The implementation of EBLs is presented in Fig. 6.

3.2.3. Proposed neighborhood search operator

The ability of particle i to improve its P_i 's fitness each time when it is evolved through the proposed velocity update mechanism is not guaranteed. To address this issue, we developed a NS operator as an alternative strategy to further evolve particle i when it fails to improve its P_i 's fitness during the first learning stage.

To perform the proposed NS operator on particle i , we first exclude the $c_{exp,i}$ and the $s_{exp,i}$ exemplars produced by particle i itself, and store the cognitive and social exemplars produced by other particles into the arrays of $c_{candidate,i} = [c_{exp,1}, c_{exp,2}, \dots, c_{exp,i-1}, c_{exp,i+1}, \dots, c_{exp,S}]$ and $s_{candidate,i} = [s_{exp,1}, s_{exp,2}, \dots, s_{exp,i-1}, s_{exp,i+1}, \dots, s_{exp,S}]$, respectively. Based on the fitness criterion, we select two guidance particles, namely $s_{guide,i}$ and $c_{guide,i}$ from the arrays of

$c_{candidate,i}$ and $s_{candidate,i}$, respectively, through the roulette wheel selection. An $o_{exp,i}$ exemplar, which is used to guide particle i in the NS operator, is then derived from the selected $c_{guide,i}$ and $s_{guide,i}$. Specifically, if a randomly generated number is smaller than 0.5, the d th dimension of the $o_{exp,i}$ exemplar, $o_{exp,i}(d)$, is donated by the $s_{guide,i}(d)$. Otherwise, it is contributed by the d th component of $c_{guide,i}$. The procedure NS_Generate_Exemplars used in generating the $o_{exp,i}$ exemplar is illustrated in Fig. 7.

Similar to the $c_{exp,i}$ exemplar, the fitness of $o_{exp,i}$ exemplar could either be better or worse than the personal best fitness of particle i . Thus, a similar strategy is employed to handle these scenarios: (1) if $f(o_{exp,i}) < f(P_i)$, particle i is attracted towards the $o_{exp,i}$ exemplar and (2) if $f(o_{exp,i}) \geq f(P_i)$, particle i is repelled from the $o_{exp,i}$ exemplar. Specifically, each particle i adjusts its P_i as follows:

$$P_{i,temp} = \begin{cases} P_i + cr_8(o_{exp,i} - P_i), & f(o_{exp,i}) < f(P_i) \\ P_i - cr_9(o_{exp,i} - P_i) & \text{otherwise} \end{cases} \quad (7)$$

where $P_{i,temp}$ is the adjusted cognitive experience of particle i and r_8 and r_9 are the random numbers in the range of $[0, 1]$. The fitness of $P_{i,temp}$ is then evaluated and compared with the P_i and P_g . If the former's fitness is better than the latter, the improved $P_{i,temp}$ replaces both P_i and P_g . As in the previous subsection, if the newly improved $P_{i,temp}$ has better fitness than the old P_i , the EBLs is

Algorithm 4: NS_Generate_Exemplars ($c_{exp_all}, s_{exp_all}, \text{particle } i, P_g, f(P_g), k$)
Input: All cognitive and social exemplars generated by the population (c_{exp_all} and s_{exp_all}), particle i 's index, global best particle's position (P_g), global best particle's fitness ($f(P_g)$), number of fitness evaluation consumed (k)
1: Exclude $c_{exp,i}$ and $s_{exp,i}$ exemplars from c_{exp_all} and s_{exp_all} , respectively; 2: Store the non-excluded cognitive and social exemplars into the $c_{candidate,i}$ and $s_{candidate,i}$, respectively; 3: /*Select $s_{guide,i}$ */ 4: for each member k in $s_{candidate,i}$ do 5: Calculate W_k for each $s_{candidate,i}$ member k using Eq. (5); 6: end for 7: Perform the roulette wheel selection based on W_k to select the $s_{guide,i}$; 8: /*Select $c_{guide,i}$ */ 9: for each member k in $c_{candidate,i}$ do 10: Calculate W_k for each $c_{candidate,i}$ member k using Eq. (5); 11: end for 12: Perform the roulette wheel selection based on W_k to select the $c_{guide,i}$; 13: /*Derive $o_{exp,i}$ */ 14: for each dimension d do 15: if $rand < 0.5$ then 16: $o_{exp,i}(d) := s_{guide,i}(d)$; 17: else 18: $o_{exp,i}(d) := c_{guide,i}(d)$; 19: end if 20: end for 21: Perform fitness evaluation on $o_{exp,i}$; 22: Update P_g and $f(P_g)$ if $o_{exp,i}$ has better fitness; 23: $k := k + 1$; Output: $o_{exp,i}$ exemplar, updated P_g , $f(P_g)$, and k

Fig. 7. Derivation of the $o_{exp,i}$ exemplar in the PSO-ITC algorithm.

Algorithm 5: NS_Operator ($c_{exp_all}, s_{exp_all}, \text{particle } i, P_i, P_g, f(P_g), k$)
Input: All cognitive and social exemplars generated by the population (c_{exp_all} and s_{exp_all}), particle i 's index, personal best position of particle i (P_i), global best particle's position (P_g), global best particle's fitness ($f(P_g)$), number of fitness evaluation consumed (k)
1: $o_{exp,i} := \text{NS_Generate_Exemplars}(c_{exp_all}, s_{exp_all}, \text{particle } i, P_g, f(P_g), k)$; 2: Calculate the $P_{i,temp}$ of particle i using (7); 3: Perform fitness evaluation on $P_{i,temp}$; 4: $k := k + 1$; 5: $previous_position_i := P_i$, $previous_fitness_i := f(P_i)$; 6: Update P_i , P_g , $f(P_i)$, and $f(P_g)$; 7: if $f(P_i) < previous_fitness_i$ and $P_g \neq P_i$ then 8: Perform the EBLs($P_i, P_g, f(P_g), k$); 9: end if Output: Updated P_g , $f(P_g)$, and k

Fig. 8. NS operator of the PSO-ITC algorithm.

Main Algorithm Block: PSO-ITC Algorithm	
Input: Population size (S), dimensionality of problem space (D), objective function (F), the initialization domain (RG), problem's accuracy level (ε), maximum number of fitness evaluation (FE_{max})	
1:	Generate initial swarm and set up parameter for each particle;
2:	Reset $k := 0$;
3:	for each particle i do
4:	Set $TC_i := 1$ and $fc_i := 0$;
5:	Initialize $neighbor_i$ by randomly selecting one particle from the population as neighborhood member;
6:	end for
7:	while $k < FE_{max}$ do
8:	for each particle i do
9:	Perform the ITC_module ($fc_i, P_g, f(P_g), k, TC_i, neighbor_i$)
10:	Update velocity \bar{V}_i and position X_i of particle i using Eqs. (6) and (2), respectively;
11:	Perform fitness evaluation on the updated X_i ;
12:	$k := k + 1$;
13:	$previous_position_i := P_i, previous_fitness_i := f(P_i)$;
14:	Update $P_i, P_g, f(P_i)$, and $f(P_g)$;
15:	if P_g is improved then
16:	$fc_i := 0$;
17:	else
18:	$fc_i := fc_i + 1$;
19:	end if
20:	if $f(P_i) < previous_fitness_i$ and $P_g \neq P_i$ then
21:	Perform the EBLS ($P_i, P_g, f(P_g), k$);
22:	else /*Particle i fails to improve its $f(P_i)$ fitness and needs to perform the NS operator*/
23:	Perform the NS_Operator ($c_{exp_all}, s_{exp_all}, particle\ i, P_i, P_g, f(P_g), k$);
24:	end if
25:	end for
25:	end while
Output: The best found solution, i.e., the global best particle's position (P_g)	

Fig. 9. Complete framework of the PSO-ITC algorithm.

triggered to extract the useful information from $P_{i,temp}$ to refine the P_g . The NS operator's implementation is presented in Fig. 8.

3.3. Complete framework of the PSO-ITC

Together with the aforementioned components, the implementation of the proposed PSO-ITC is summarized in Fig. 9. Although some working mechanisms in PSO with expanding neighborhood topology (PSOENT) proposed by Marinakis and Marinaki (2013) are similar to PSO-ITC, several differences exist between these two PSO variants. First, the particle's topology connectivity in our PSO-TVTC is updated according to the schedule defined in Eq. (3), whereas PSOENT updates its neighborhood based on the quality of the produced solution. Second, our PSO-ITC is equipped with the shuffling mechanism in the ITC module to reassign a particle's neighborhood members when the algorithm fails to improve the global best fitness for certain FEs. In contrast, PSOENT expands its neighborhood when a similar circumstance is encountered. Finally, a learning framework is designed for the PSO-ITC to improve the algorithm's performance, whereas no such learning framework is adopted by the PSOENT.

4. Simulation results

In this section, we describe the experimental settings used to evaluate the PSO-ITC and present the experimental results.

4.1. Benchmark functions

In this paper, we employ a total of 20 scalable benchmark functions (Suganthan et al., 2005; Tang et al., 2011; Yao et al., 1999) to extensively evaluate the performance of PSO-ITC and its contenders. We perform the evaluation with 50 variables, $D=50$. Table 1 lists these benchmarks and describes their formulae, their feasible search range RG , their global minimum fitness F_{min} , and their accuracy level ε . The accuracy level is used to decide whether a particular algorithm run is

successful, that is, the problem is considered solved when the approximate solution is not farther than ε from the actual one.

Table 1 shows that the employed benchmarks are categorized into four classes: (1) conventional problems, (2) rotated problems, (3) shifted problems, and (4) complex problems. Each function in the conventional problems (F1–F8) has a different characteristic, which enables us to assess the algorithm's performance with the use of various criteria. For example, function F1 is used to test the algorithm's convergence speed, as it is relatively easy to be solved. Functions F4, F5, F7, and F8 are multimodal functions used to evaluate the algorithm's capability to escape from the local optima, as these functions consist of a large number of local optima in a high-dimensional case. The rotated problems (F9–F13) are developed to prohibit a one-dimensional search, which is permissible in certain conventional problems (e.g., functions F1, F4, F5, and F8). The prevention of one-dimensional search can be done by multiplying the original X_i variable with an orthogonal matrix M (Salomon, 1996) to produce a rotated variable Z_i , that is, $Z_i = M * X_i$. As a result, any change that occurs in X_i will affect all dimensions in Z_i , which then leads to non-separability of the rotated problems. For shifted problems (F14–F17), a vector $o = [o_1, o_2, \dots, o_D]$ is defined to adjust the global optima of the conventional problems to a new location, $Z_i = X_i - o$. The complex problems (F18–F20) consist of the shifted and rotated problems (F18 and F19) and the expanded problem (F20). The former integrates both the rotating and the shifting characteristics into the conventional problems, whereas the latter is derived by taking the two-dimensional Rosenbrock function (F3) as the input argument of the Griewank function (F8) (Suganthan et al., 2005).

4.2. Simulation settings for the involved PSO algorithms

In this paper, we employ nine well-established PSO variants for a thorough comparison of the PSO-ITC with other PSO variants. The parameter settings for all PSO variants are extracted from their corresponding literatures and summarized in Table 2. The parameter settings of these nine PSO variants are set by their

Table 1
Twenty benchmark functions used in this study (note: $f_{biasj, \forall j \in [1,7]}$ denotes the shifted fitness value applied to the corresponding functions).

No.	Function name	Formulae	RG	F_{min}	ϵ
Category I: conventional problems					
F1	Sphere	$F_1(X_i) = \sum_{d=1}^D X_{i,d}^2$	$[-100, 100]^D$	0	$1.0e-6$
F2	Schwefel 1.2	$F_2(X_i) = \sum_{d=1}^D \left(\sum_{j=1}^d X_{i,j} \right)^2$	$[-100, 100]^D$	0	$1.0e-6$
F3	Rosenbrock	$F_3(X_i) = \sum_{d=1}^{D-1} (100(X_{i,d}^2 - X_{i,d+1})^2 + (X_{i,d} - 1)^2)$	$[-2.048, 2.048]^D$	0	$1.0e-2$
F4	Rastrigin	$F_4(X_i) = \sum_{d=1}^D (X_{i,d}^2 - 10 \cos(2\pi X_{i,d}) + 10)$	$[-5.12, 5.12]^D$	0	$1.0e-2$
F5	Noncontinuous Rastrigin	$F_5(X_i) = \sum_{d=1}^D (Y_{i,d}^2 - 10 \cos(2\pi Y_{i,d}) + 10)$ where $Y_{i,d} = \begin{cases} X_{i,d}, & X_{i,d} < 0.5 \\ \text{round}(2X_{i,d})/2, & X_{i,d} \geq 0.5 \end{cases}$	$[-5.12, 5.12]^D$	0	$1.0e-2$
F6	Griewank	$F_6(X_i) = \sum_{d=1}^D X_{i,d}^2 / 4000 - \prod_{d=1}^D \cos(X_{i,d} / \sqrt{d}) + 1$	$[-600, 600]^D$	0	$1.0e-2$
F7	Ackley	$F_7(X_i) = -20 \exp\left(-0.2 \sqrt{\sum_{d=1}^D X_{i,d}^2 / D}\right) - \exp\left(\sum_{d=1}^D \cos(2\pi X_{i,d}) / D\right) + 20 + e$	$[-32, 32]^D$	0	$1.0e-2$
F8	Weierstrass	$F_8(X_i) = \sum_{d=1}^D \left(\sum_{k=0}^k [a^k \cos(2\pi b^k (X_{i,d} + 0.5))] \right) - D \sum_{k=0}^k [a^k \cos(\pi b^k)]$ $a = 0.5, b = 3, k \max = 20$	$[-0.5, 0.5]^D$	0	$1.0e-2$
Category II: rotated problems					
F9	Rotated sphere	$F_9(X_i) = F_1(Z_i), Z_i = M * X_i$	$[-100, 100]^D$	0	$1.0e-6$
F10	Rotated Schwefel 1.2	$F_{10}(X_i) = F_2(Z_i), Z_i = M * X_i$	$[-100, 100]^D$	0	$1.0e-2$
F11	Rotated Rosenbrock	$F_{11}(X_i) = F_3(Z_i), Z_i = M * X_i$	$[-2.048, 2.048]^D$	0	$1.0e-2$
F12	Rotated Rastrigin	$F_{12}(X_i) = F_4(Z_i), Z_i = M * X_i$	$[-5.12, 5.12]^D$	0	$1.0e-2$
F13	Rotated Griewanks	$F_{13}(X_i) = F_6(Z_i), Z_i = M * X_i$	$[-600, 600]^D$	0	$1.0e-2$
Category III: shifted problems					
F14	Shifted sphere	$F_{14}(X_i) = F_1(Z_i) + f_{bias1}, Z_i = X_i - o, f_{bias1} = -450$	$[-100, 100]^D$	-450	$1.0e-6$
F15	Shifted Rastrigin	$F_{15}(X_i) = F_4(Z_i) + f_{bias2}, Z_i = X_i - o, f_{bias2} = -330$	$[-5.12, 5.12]^D$	-330	$1.0e-2$
F16	Shifted Noncontinuous Rastrigin	$F_{16}(X_i) = F_5(Z_i) + f_{bias3}, Z_i = X_i - o, f_{bias3} = -330$	$[-5.12, 5.12]^D$	-330	$1.0e-2$
F17	Shifted Griewank	$F_{17}(X_i) = F_6(Z_i) + f_{bias4}, Z_i = X_i - o, f_{bias4} = -180$	$[-600, 600]^D$	-180	$1.0e-2$
Category IV: complex problems					
F18	Shifted Rotated Griewank	$F_{18}(X_i) = F_6(Z_i) + f_{bias5}, Z_i = (X_i - o) * M, f_{bias5} = -180$	$[-600, 600]^D$	-180	$1.0e-2$
F19	Shifted Rotated High Conditioned Elliptic	$F_{19}(X_i) = \sum_{d=1}^D (10^6)^{d-1/D-1} Z_{i,d}^2 + f_{bias6}, Z_i = (X_i - o) * M, f_{bias6} = -450,$	$[-100, 100]^D$	-450	$1.0e-6$
F20	Shifted Expanded Griewank's plus Rosenbrock	$F_{20} = F_6(F_3(Z_{i,1}, Z_{i,2})) + F_6(F_3(Z_{i,2}, Z_{i,3})) + \dots + F_6(F_3(Z_{i,D-1}, Z_{i,D})) + F_6(F_3(Z_{i,D}, Z_{i,1})) + f_{bias7}$	$[-5, 5]^D$	-130	$1.0e-2$
		$Z_i = X_i - o, f_{bias7} = -130$			

Table 2
Parameter settings of the involved PSO algorithms.

Algorithm (references)	Population topology	Parameter settings
APSO (Zhan et al., 2009)	Fully connected	$\omega : 0.9-0.4, c_1 + c_2 : [3.0, 4.0], \delta = [0.05, 0.1], \sigma_{max} = 1.0, \sigma_{min} = 0.1$
FLPSO-QIW (Tang et al., 2011)	Comprehensive learning	$\omega : 0.9-0.2, c_1 : 2-1.5, c_2 : 1-1.5, m = 1, P_1 = [0.1, 1], K_1 = 0.1, K_2 = 0.001, \sigma_1 = 1, \sigma_2 = 0$
FlexiPSO (Kathrada, 2009)	Fully connected and local ring	$\omega : 0.5-0.0, c_1, c_2, c_3 : [0.0, 2.0], \epsilon = 0.1, \alpha = 0.01\%$
FPSO (Montes de Oca et al., 2009)	Decreasing	$\chi = 0.729, \sum c_i = 4.1$
FIPSO (Mendes et al., 2004)	Local URing	$\chi = 0.729, \sum c_i = 4.1$
OLPSO-L (Zhan et al., 2011)	Orthogonal learning	$\omega : 0.9-0.4, c = 2.0, G = 5$
HPSO-TVAC (Ratnaweera et al., 2004)	Fully connected	$\omega : 0.9-0.4, c_1 : 2.5-0.5, c_2 : 0.5-2.5$
RPPSO (Zhou et al., 2011)	Random	$\omega : 0.9-0.4, c_{large} = 6, c_{small} = 3$
BPSO (Shi and Eberhart, 1998)	Fully connected	$\omega : 0.9-0.4, c_1 = c_2 = 2.0$
PSO-ITC	Increasing	$\omega : 0.9-0.4, c_1 = c_2 = c = 2.0, z = 5, TC_{min} = 1, TC_{max} = S-1$

corresponding authors, and these settings are the optimized ones. For our PSO-ITC, a parameter sensitivity analysis described in the following subsection is performed to investigate the effect of parameter z on the searching performance of the PSO-ITC.

To ensure fair performance assessment between the PSO-ITC and its contenders, all PSO variants are run independently 30 times on the employed 20 benchmarks. We use the maximum number of fitness evaluation FE_{max} as the termination criterion for all involved algorithms. In addition, the calculations are stopped if the exact solution X^*

is found, $F(X) = F(X^*)$. The population size and FE_{max} used in $D=50$ cases are 30 and 300,000, respectively (Suganthan et al., 2005).

4.3. Performance metrics

We assess the PSO's performance based on three criteria, namely accuracy, reliability, and efficiency, through the mean fitness value (F_{mean}), success rate (SR), and success performance (SP), respectively (Suganthan et al., 2005).

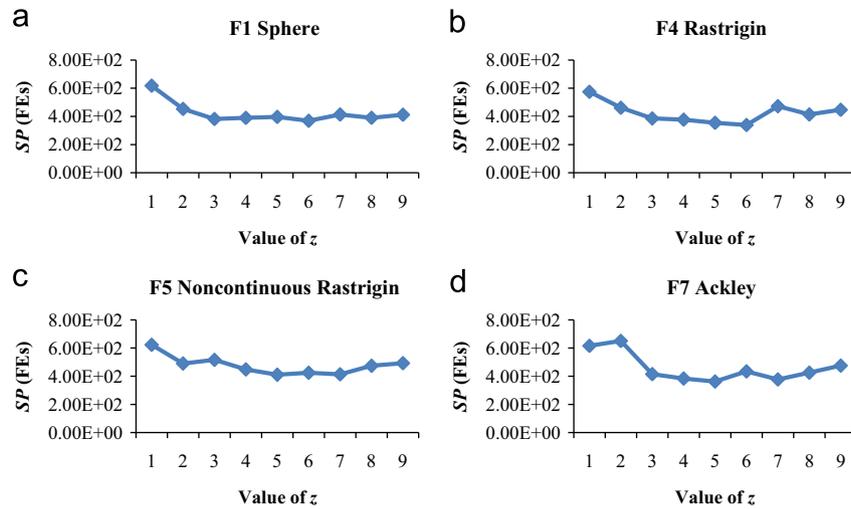


Fig. 10. Mean fitness (F_{mean}) and success performance (SP) values obtained by the PSO-ITC on (a) F1 Sphere, (b) F4 Rastrigin, (c) F5 Noncontinuous Rastrigin, and (d) F7 Ackley functions with different z values.

F_{mean} value is the mean value of the differences between the best (i.e., lowest) fitness value found by the algorithm and the actual global optimum's fitness (F_{min}) (Suganthan et al., 2005). A smaller F_{mean} is favorable, as it implies that the algorithm has better searching accuracy. The SR value is used to evaluate the consistency of an algorithm to achieve a successful run, that is, the ability to solve the given problem with the predefined accuracy level ε (Suganthan et al., 2005). The algorithm with the larger SR value is more reliable, as it can consistently solve the problem with predefined ε .

The computational cost required by an algorithm to solve the problem with the predefined ε can be measured by the SP value (Suganthan et al., 2005) or the mean computational time (t_{mean}). However, previous papers state that the former is more suitable than the latter for evaluating the performance of algorithms that solve real-world problems (Zhan et al., 2009). Thus, we employ the SP value to justify the algorithm's speed. A smaller SP value is preferable as it implies that the algorithm requires less computation cost to solve the problems with acceptable accuracy levels. Finally, to thoroughly investigate the significance of the performance deviation between the PSO-ITC and its peers, we perform a two-tailed t -test (Tang et al., 2011) with 58 degrees of freedom at a 0.05 level of significance (or 95% confidence level). More precisely, the h sign produced by the t -test is used to evaluate if the performance of the PSO-ITC is better (i.e., $h = "+"$), insignificant (i.e., $h = "="$), or worse (i.e., $h = "-"$) than the other nine algorithms at the statistical level.

4.4. Parameter sensitivity analysis

As explained in the previous section, the ITC module performs the shuffling mechanism when the PSO-ITC fails to improve the fitness of the global best particle, that is, $f(P_g)$ for z successive FEs. The dependency of the ITC module on z implies that different values of z may affect the performance of the PSO-ITC. Thus, we perform parameter sensitivity analysis to investigate the performance of the PSO-ITC under variant z values.

The parameter settings of the PSO-ITC in the parameter sensitivity analysis are as follows: we evaluate four 10- D benchmarks with different characteristics, namely the Sphere (F1), Rastrigin (F4), Noncontinuous Rastrigin (F5), and Ackley (F7) functions. These problems are solved by PSO-ITC with the use of z with an integer value from 1 to 9. Each different z value is run 30 times, with the population size (S) and the maximum fitness

evaluation numbers (FE_{max}) of 10 and $5.00E+04$, respectively. The experimental findings obtained by the PSO-ITC with different values of z are summarized in Fig. 10.

The simulation results reveal that the searching accuracy of the PSO-ITC, represented by the F_{mean} value, is not sensitive to the parameter z . More specifically, the PSO-ITC successfully locates the global optima of all employed benchmarks (i.e., functions F1, F4, F5, and F7) regardless of which z value is chosen. We omit the results of F_{mean} from the graphs in Fig. 10 because the F_{mean} values produced by the PSO-ITC in all employed benchmarks are zero for $z = 1, \dots, 9$ and presenting all-zero data in the graphs is not useful. We also observe that the parameter z influences the algorithm's efficiency, which is represented by SP values. Specifically, the PSO-ITC's efficiency deteriorates when the values of z are set too high (i.e., $z = 7, 8, 9$) or too low (i.e., $z = 1, 2, 3$). We conjecture that when the value of z is set too low, the shuffling mechanism is not triggered frequently enough. Consequently, the diversity provided by the ITC modules to the population is inadequate, which in turn entraps the PSO-ITC swarm at local optima for too long a time. In contrast, the shuffling mechanism in the ITC module can be overemphasized when the z value is set too high. In this extreme scenario, the ITC module provides excessive diversity to the swarm, which in turn potentially jeopardizes the convergence rate of the PSO-ITC toward the problem's global optimum. Finally, the results of parameter sensitivity analysis reveal that the PSO-ITC efficiently solves the four employed benchmarks with z values between 4 and 6. More precisely, the PSO-ITC achieves the best SP values in functions F1 and F4 when z is set as 6, whereas the z value of 5 solves the functions F5 and F7 with the least computational cost. Based on these findings, we set z to be 5 for PSO-IDL in the following performance evaluations.

4.5. Comparisons of the PSO-ITC with other well-established PSO variants

The results of the F_{mean} , standard deviation (SD), t -test (h), SR , and SP attained by all nine PSO variants for all problems are presented in Table 3. The best result for each benchmark is indicated in boldface. We also include the convergence graphs of all benchmark problems in Appendix A1 for us to qualitatively (i.e., visually) compare the F_{mean} and convergence speed of all involved algorithms.

At the bottom of Table 3, we summarize the F_{mean} comparison results between PSO-ITC and other peers as " $w/t/l$ " and "#BMF." " $w/t/l$ " means that PSO-ITC wins over a particular peer in w functions, ties for

Table 3
Overall experimental results for 50-D problems.

f		APSO	FLPSO-QJW	FlexiPSO	FPSO	FIPSO	OLPSO-L	HPSO-TVAC	RPPSO	BPSO	PSO-ITC
F1	F_{mean}	2.50E-01	2.90E-81	1.78E-04	7.02E+01	2.96E-01	4.86E-33	1.09E-05	1.28E-02	4.67E+03	0.00E+00
	SD	1.81E-01	5.97E-81	5.23E-05	6.98E+01	8.06E-01	5.15E-33	3.69E-06	2.98E-02	7.30E+03	0.00E+00
	SR	0.00	100.00	0.00	13.33	80.00	100.00	0.00	73.33	63.33	100.00
	SP	Inf	6.04E+04	Inf	9.68E+04	9.86E+04	1.52E+05	Inf	1.52E+04	1.63E+04	1.78E+03
	h	+	+	+	+	+	+	+	+	+	+
F2	F_{mean}	1.46E+03	2.62E+02	1.42E+00	3.44E+03	8.13E+00	5.71E+02	9.48E-02	9.12E+01	2.08E+04	0.00E+00
	SD	4.82E+02	8.90E+01	6.67E-01	1.33E+03	2.47E+01	1.85E+02	3.60E-02	4.21E+01	1.59E+04	0.00E+00
	SR	0.00	0.00	0.00	0.00	70.00	0.00	0.00	0.00	0.00	100.00
	SP	Inf	Inf	Inf	Inf	1.62E+05	Inf	Inf	Inf	Inf	6.16E+04
	h	+	+	+	+	=	+	+	+	+	+
F3	F_{mean}	4.62E+01	4.22E+01	4.48E+01	5.68E+01	4.77E+01	4.30E+01	4.60E+01	4.76E+01	2.10E+02	4.32E+01
	SD	1.53E+00	2.39E-01	1.04E+00	7.08E+00	8.44E-01	3.18E+00	5.70E-01	4.30E-01	4.34E+02	8.16E+00
	SR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.33
	SP	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	8.20E+06
	h	+	-	+	+	+	-	+	+	+	+
F4	F_{mean}	5.80E-01	2.60E+00	2.12E-04	1.85E+01	1.57E+00	3.32E-01	9.47E-04	9.25E+00	1.15E+02	0.00E+00
	SD	6.29E-01	1.52E+00	6.24E-05	1.02E+01	3.71E+00	6.03E-01	3.02E-04	1.55E+01	7.78E+01	0.00E+00
	SR	0.00	6.67	100.00	0.00	40.00	73.33	100.00	70.00	3.33	100.00
	SP	Inf	3.46E+06	9.72E+04	Inf	3.56E+05	2.97E+05	1.85E+05	4.19E+04	4.70E+05	2.24E+03
	h	+	+	+	+	+	+	+	+	+	+
F5	F_{mean}	3.60E-02	5.58E+00	2.07E-04	1.60E+01	5.70E-01	1.17E+00	1.50E-03	1.25E+01	1.14E+02	0.00E+00
	SD	3.22E-02	2.36E+00	7.51E-05	9.56E+00	8.65E-01	1.15E+00	1.87E-03	1.94E+01	5.81E+01	0.00E+00
	SR	6.67	0.00	100.00	0.00	33.33	40.00	96.67	60.00	3.33	100.00
	SP	2.60E+06	Inf	9.88E+04	Inf	2.66E+05	6.75E+05	2.05E+05	2.50E+04	6.04E+05	2.57E+03
	h	+	+	+	+	+	+	+	+	+	+
F6	F_{mean}	1.70E-01	5.75E-04	8.34E-03	1.86E+00	1.93E-01	0.00E+00	3.86E-03	7.08E-03	3.92E+01	0.00E+00
	SD	8.21E-02	2.21E-03	9.48E-03	9.28E-01	3.47E-01	0.00E+00	6.55E-03	1.85E-02	7.00E+01	0.00E+00
	SR	0.00	100.00	60.00	6.67	70.00	100.00	90.00	86.67	73.33	100.00
	SP	Inf	5.00E+04	1.57E+05	1.60E+05	1.05E+05	1.24E+05	7.23E+04	1.34E+0	1.13E+04	2.20E+03
	h	+	=	+	+	+	=	+	+	+	+
F7	F_{mean}	6.60E-02	3.43E-14	3.55E-03	1.80E+00	1.70E-01	5.09E-15	1.57E-03	7.47E-01	1.21E+01	0.00E+00
	SD	2.57E-02	1.07E-14	5.36E-04	1.10E+00	3.38E-01	1.79E-15	1.99E-04	9.17E-01	5.99E+00	0.00E+00
	SR	0.00	100.00	100.00	3.33	53.33	100.00	100.00	36.67	16.67	100.00
	SP	Inf	4.79E+04	1.59E+05	3.36E+05	1.07E+05	1.25E+05	8.39E+04	1.25E+05	6.96E+04	1.54E+03
	h	+	+	+	+	+	+	+	+	+	+
F8	F_{mean}	5.44E-01	1.88E-05	1.12E-01	3.35E+00	9.80E-01	0.00E+00	2.03E+00	4.69E-01	8.09E+00	0.00E+00
	SD	1.88E-01	8.29E-05	1.16E-02	2.35E+00	9.53E-01	0.00E+00	3.64E-01	1.25E+00	5.90E+00	0.00E+00
	SR	0.00	100.00	0.00	0.00	16.67	100.00	0.00	83.33	16.67	100.00
	SP	Inf	6.67E+04	Inf	Inf	4.59E+05	1.66E+05	Inf	3.15E+04	2.11E+05	1.89E+03
	h	+	=	+	+	+	=	+	+	+	+
F9	F_{mean}	2.01E-01	1.15E-80	2.10E-04	6.28E+01	4.97E-01	2.89E-33	1.16E-05	4.98E-03	4.67E+03	0.00E+00
	SD	1.17E-01	4.42E-80	6.90E-05	6.96E+01	1.06E+00	2.19E-33	4.81E-06	1.58E-02	5.71E+03	0.00E+00
	SR	0.00	100.00	0.00	13.33	73.33	100.00	0.00	86.67	56.67	100.00
	SP	Inf	6.04E+04	Inf	8.24E+04	9.70E+04	1.52E+05	Inf	1.35E+04	2.20E+04	2.21E+03
	h	+	=	+	+	+	+	+	=	+	+

F10	F_{mean}	1.26E+03	2.62E+02	4.92E+00	3.23E+03	8.45E+00	1.92E+03	1.13E-01	9.09E+01	2.57E+04	0.00E+00
	SD	3.22E+02	7.62E+01	3.67E+00	1.79E+03	2.24E+01	4.17E+02	5.78E-02	3.77E+01	1.75E+04	0.00E+00
	SR	0.00	0.00	0.00	0.00	73.33	0.00	0.00	3.33	3.33	100.00
	SP	Inf	Inf	Inf	Inf	1.62E+05	Inf	Inf	1.69E+06	7.47E+05	6.66E+04
	h	+	+	+	+	+	+	+	+	+	+
F11	F_{mean}	5.15E+01	4.55E+01	4.59E+01	5.62E+01	4.85E+01	4.24E+01	4.60E+01	5.03E+01	1.08E+02	4.37E+01
	SD	1.39E+01	3.16E+00	3.60E+00	7.00E+00	5.72E-02	3.73E+00	3.03E+00	8.66E+00	2.25E+02	7.00E-01
	SR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	SP	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
	h	+	+	+	+	+	-	+	+	+	+
F12	F_{mean}	1.83E+02	1.26E+02	1.49E+02	1.80E+02	2.65E+01	9.80E+01	9.07E+01	4.25E+01	1.70E+02	0.00E+00
	SD	5.61E+01	1.76E+01	3.42E+01	5.01E+01	3.39E+01	5.16E+01	2.74E+01	4.64E+01	7.41E+01	0.00E+00
	SR	0.00	0.00	0.00	0.00	53.33	0.00	0.00	40.00	6.67	100.00
	SP	Inf	Inf	Inf	Inf	2.60E+05	Inf	Inf	3.95E+04	3.84E+05	4.13E+04
	h	+	+	+	+	+	+	+	+	+	+
F13	F_{mean}	2.10E+02	1.52E+00	2.67E+02	7.28E+00	1.94E-01	7.58E-01	1.55E+02	3.03E+01	2.04E+02	0.00E+00
	SD	1.01E+02	5.39E-01	9.17E+01	5.62E+00	4.08E-01	2.68E-01	3.95E+01	5.60E+01	2.48E+02	0.00E+00
	SR	0.00	0.00	0.00	6.67	80.00	0.00	0.00	53.33	43.33	100.00
	SP	Inf	Inf	Inf	2.59E+05	8.72E+04	Inf	Inf	2.15E+04	2.81E+04	4.09E+03
	h	+	+	+	+	+	+	+	+	+	+
F14	F_{mean}	2.27E-01	1.44E-13	3.65E-04	1.71E+04	6.20E+00	5.68E-14	1.52E-05	1.44E-01	2.31E+04	1.01E-08
	SD	9.70E-02	4.15E-14	6.12E-04	1.47E+04	3.90E+00	0.00E+00	4.12E-06	5.22E-01	1.79E+04	3.91E-09
	SR	0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00
	SP	Inf	5.85E+04	Inf	Inf	Inf	1.37E+05	Inf	Inf	Inf	2.28E+05
	h	+	-	+	+	+	-	+	+	+	+
F15	F_{mean}	5.92E-01	5.88E+00	2.06E-04	2.08E+02	1.31E+02	1.43E+00	1.34E-01	1.62E+02	2.93E+02	1.75E-07
	SD	7.76E-01	2.51E+00	6.95E-05	4.59E+01	2.93E+01	1.10E+00	4.32E-01	4.08E+01	5.76E+01	5.88E-08
	SR	0.00	0.00	100.00	0.00	0.00	13.33	90.00	0.00	0.00	100.00
	SP	Inf	Inf	1.04E+05	Inf	Inf	1.27E+06	2.55E+05	Inf	Inf	1.29E+05
	h	+	+	+	+	+	+	+	+	+	+
F16	F_{mean}	7.20E-03	1.20E+01	2.06E-04	1.63E+02	1.48E+02	3.00E+00	1.02E-01	2.09E+02	3.14E+02	2.07E-07
	SD	1.06E-02	3.16E+00	6.44E-05	2.82E+01	3.94E+01	1.78E+00	3.05E-01	5.21E+01	6.18E+01	7.68E-08
	SR	86.67	0.00	100.00	0.00	0.00	6.67	90.00	0.00	0.00	100.00
	SP	2.51E+05	Inf	1.15E+05	Inf	Inf	3.45E+06	2.55E+05	Inf	Inf	1.43E+05
	h	+	+	+	+	+	+	+	+	+	+
F17	F_{mean}	0.00E+00	2.05E-03	2.76E+02	1.46E+03	0.00E+00	1.47E-01	5.72E-03	0.00E+00	7.21E+02	0.00E+00
	SD	0.00E+00	3.49E-03	3.86E+02	4.63E+02	0.00E+00	1.07E-01	3.13E-02	0.00E+00	4.54E+02	0.00E+00
	SR	100.00	100.00	56.67	0.00	100.00	10.00	96.67	100.00	13.33	100.00
	SP	1.20E+04	4.88E+04	2.45E+03	Inf	2.47E+03	1.33E+06	8.56E+04	9.09E+02	4.54E+03	6.83E+02
	h	=	+	=	+	=	+	+	=	+	+
F18	F_{mean}	1.49E+00	6.02E-03	1.52E-01	2.06E+02	4.81E+00	3.76E-02	4.96E-03	6.60E-01	7.05E+02	9.33E-03
	SD	1.47E-01	1.04E-02	7.18E-02	1.44E+02	1.78E+00	4.00E-02	7.37E-03	2.61E-01	4.24E+02	1.16E-02
	SR	0.00	83.33	0.00	0.00	0.00	13.33	66.67	0.00	0.00	50.00
	SP	Inf	2.04E+05	Inf	Inf	Inf	2.20E+06	3.08E+05	Inf	Inf	4.93E+05
	h	+	-	+	+	+	+	-	+	+	+
F19	F_{mean}	1.32E+07	1.89E+07	1.32E+07	1.03E+08	1.02E+07	1.82E+07	1.62E+06	1.49E+07	6.11E+08	7.98E+06
	SD	4.09E+06	4.92E+06	8.09E+06	8.26E+07	3.44E+06	5.14E+06	7.45E+05	1.13E+07	6.42E+08	2.55E+06
	SR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	SP	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
	h	+	+	+	+	+	+	-	+	+	+

Table 3 (continued)

<i>f</i>	AFPSO	FLPSO-QIW	FlexiPSO	FPSO	FIPSO	OLPSO-L	HPSO-TVAC	RPPSO	BPSO	PSO-ITC
F20	4.13E+00 1.19E+00 0.00 Inf	4.01E+00 1.46E+00 0.00 Inf	2.25E+00 7.43E-01 0.00 Inf	4.35E+01 8.59E+00 0.00 Inf	2.76E+01 5.79E+00 0.00 Inf	2.98E+00 8.26E-01 0.00 Inf	6.64E+00 2.00E+00 0.00 Inf	4.79E+01 2.19E+01 0.00 Inf	2.64E+04 6.86E+04 0.00 Inf	1.15E+00 4.49E-01 0.00 Inf
w/t/l	+ 19/1/1	+ 17/0/3	+ 20/0/0	+ 20/0/0	+ 19/1/0	+ 15/2/3	+ 18/0/2	+ 19/1/0	+ 20/0/0	+ 20/0/0
+/=/-	19/1/1	14/3/3	20/0/0	20/0/0	18/2/0	15/2/3	18/0/2	18/2/0	20/0/0	20/0/0
#S/#PS/#NS	1/2/17	7/2/11	5/2/13	0/5/15	1/11/8	6/6/8	2/6/12	1/10/9	0/11/9	15/2/3
#BMF	1	1	0	0	1	3	2	1	0	15
#BSP	0	2	2	0	0	0	0	1	0	12

t functions, and loses in *l* functions. #BMF represents the number of the best (i.e., lowest) F_{mean} values achieved by each PSO variant. *t*-test results (*h*) are summarized as “+/=/-” to indicate the number of functions in which PSO-ITC performs significantly better, almost the same as, and significantly worse than its contender, respectively. Finally, the SR and the SP results are summarized as “#S/#PS/#NS” and #BSP, respectively, where the former indicates the numbers of functions that are solved completely (i.e., SR=100%), solved partially (i.e., 0% < SR < 100%), and never solved (i.e., SR=0%) by a particular PSO variant, whereas the latter represents the number of the best (i.e., lowest) SP values attained by the involved PSO variants.

4.5.1. Comparison among the F_{mean} results

Table 3 indicates that the proposed PSO-ITC has the most superior searching accuracy as it outperforms its peers with a large margin in the majority of the problems. Specifically, the PSO-ITC achieves 15 best F_{mean} values out of the 20 employed benchmarks. For the conventional (F1–F8) and the rotated (F9–F13) problems, the proposed PSO-ITC successfully locates the global optima of all problems, except for the functions F3 and F11. More particularly, the PSO-ITC is the only algorithm to solve the conventional functions of F1, F2, F4, F5, and F7. Another significant finding is that except for the PSO-ITC, the remaining PSO variants experience different levels of performance degradation in the rotated problems compared with the conventional counterpart. Our PSO-ITC is the only PSO variant that is robust in the rotation operation, as it is able to find the global optima for all rotated problems, except for function F11. We also observe that the F_{mean} values attained by all involved algorithms in the conventional (F3) and the rotated (F11) Rosenbrock functions are relatively large. The inferior performance of all algorithms in functions F3 and F11 is attributed to the fact that the global optima of these functions are located in a long, narrow, parabolic-shaped valley to test the algorithm’s ability in navigating flat regions with a small gradient. Most of the involved algorithms are able to locate the aforementioned valley but are hardly able to converge towards the global optimum, which causes them to obtain a poor F_{mean} value.

Meanwhile, all the involved algorithms, including PSO-ITC, also suffer performance deterioration in shifted problems (F14–F17), as none of them is able to find the global optima for all shifted problems, except for function F17. Nevertheless, we observe that the PSO-ITC is least affected by the shifting operation, as it produces the three best F_{mean} and one second best F_{mean} values in four shifted problems. Specifically, the PSO-ITC is the only algorithm to successfully obtain the F_{mean} values with an accuracy level of 10^{-7} in functions F15 and F16. On the complex problems (F18–F20), we observe further performance degradation of the involved algorithms, as the inclusion of both rotating and shifting operations (F18–F19) and expanded operation (F20) has significantly increased the problems’ complexities, which makes the problems more difficult to solve. We observe that the performance of the PSO-ITC in complex problems is competitive, as it produces the best, second best, and third best F_{mean} values in functions 20, 19, and 18, respectively. Despite slightly inferior F_{mean} values compared with the FLPSO-QIW and HPSO-ITC in functions 18 and 19, such performance deviations are relatively insignificant when compared with the outstanding performances of the PSO-ITC over the aforementioned peers in other types of problems.

4.5.2. Comparisons among the *t*-test results

The *t*-test results, represented by the *h* signs in Table 3, are largely consistent with the previously reported F_{mean} values, given that the summarized results of “w/t/l” and “+/=/-” are almost the same. Table 3 shows that PSO-ITC performs significantly better than its peers in more problems and achieves significantly worse

results than its peers in fewer problems. Specifically, PSO-ITC significantly outperforms all its contenders in 10 of the 20 employed functions, that is, functions F1, F4, F5, F7, F10, F12, F13, F15, F16, and F20. These observations validate the excellent searching accuracy of the PSO-ITC compared with its peers.

Another noteworthy observation is that despite the largely different F_{mean} values produced by PSO-ITC and FLPSO-QIW in function F6 (i.e., 0 and 5.75E-04, respectively), the t -test's result reveals that such a performance difference is insignificant at a statistical level. Similar ambiguous scenarios could be observed in FLPSO-QIW, FIPSO, RPPSO, and FlexiPSO in functions F6, F2, F9, and F17, respectively. This ambiguity is due to fact that when certain algorithms, such as FLPSO-QIW, run with a predefined number of independent runs, they have a small probability of stagnating in the local optima, thereby producing a relatively large fitness value that can jeopardize the overall F_{mean} value. As shown in Table 3, despite having a relatively large F_{mean} value in function F6, the FLPSO-QIW successfully solves the problem completely, that is, $SR=100\%$.

4.5.3. Comparisons among the SR results

From Table 3, we observe that the PSO-ITC has a more superior searching reliability than its peers, as it is able to completely solve 15 of the 20 employed benchmarks, that is, it is 2.5 times better than the second ranked OLPSO-L that completely solved six problems. Specifically, the PSO-ITC has successfully solved all the conventional and the rotated problems completely, except for the rotated and the non-rotated Rosenbrock problems (i.e., F11 and F3). The PSO-ITC is the only algorithm to completely solve the functions F2, F10, F12, and F13, as well as partially solve function F3. The excellent performance of the PSO-ITC can also be observed in the shifted problems, as it is the only algorithm to completely solve all the shifted problems with the predefined ϵ . For complex problems, we observe that the searching reliabilities of all involved algorithms are compromised, as none of them is able to solve the problems completely or partially, except for function F18, where PSO-ITC achieves the third best SR value. Although none of the involved algorithms is able to solve functions F19 and F20 completely, PSO-ITC performs competitively because it produces prominent F_{mean} values in these aforementioned problems, as reported in Table 3.

4.5.4. Comparisons among the SP results

Obtaining the SP value is impossible if an algorithm never solves a particular problem (i.e., $SR=0\%$) because the SP value is the computational cost required by an algorithm to solve the problem with a pre-specified accuracy level ϵ . In this scenario, an infinity value “Inf” is assigned to the SP value, and only the convergence graphs, as illustrated in Appendix A, are used to justify the algorithms' speed.

From Table 3, we observe that PSO-ITC achieves the best SP values in all conventional problems, which implies that our proposed approach requires the least computational cost to solve the conventional problems with an acceptable ϵ . The excellent convergence of PSO-ITC in these problems is well supported by the corresponding convergence graphs shown in Appendix A. Except for function F3, we observe a typical feature exhibited by the convergence curves of the PSO-ITC in all other conventional problems, that is, a curve that sharply drops off at one point, usually at the early stage [functions F1, and F4–F8, as illustrated in Fig. A1(a) and (d)–(h), respectively] or the middle stage [function F2 as illustrated by Fig. A1(b)] of the optimization. These observations reveal the ability of the PSO-ITC to break out of the local optima and to locate the global optima by consuming a significantly small amount of FEs.

For rotated problems, the PSO-ITC achieves three (out of five) best SP values in functions F9, F10, and F13. The competitive convergence speeds of the PSO-ITC in the rotated problems are also well justified by the convergence graphs in Appendix A. Specifically, the convergence graphs of the PSO-ITC in functions F9 and F13 [illustrated by Fig. A1(i) and (m), respectively] are sharply dropped off at a very early stage of the optimization, which indicates the high efficiency of PSO-ITC in solving these two rotated problems. Meanwhile, as illustrated in Fig. A1(j) and (l), the convergence graphs of PSO-ITC in functions F10 and F12 sharply drop off at one point at the later stage of the optimization. These observations prove the robustness of PSO-ITC in handling premature convergence at local optima. In contrast, the convergence speed of the PSO-ITC in the shifted problems is slightly compromised, as it produces one best, one second best, and two third best SP values in functions F17, F16, F15, and F14, respectively. We speculate that PSO-ITC may require higher computational cost to locate the shifted global optimum regions, thereby leading to the slightly inferior SP values. Nevertheless, we observe that the algorithm with smaller SP values is not guaranteed to have

Table 4
Mean computation time in CPU cycle (in seconds) for 50-D problems.

f	APSO	FLPSO-QIW	FlexiPSO	FPSO	FIPSO	OLPSO-L	HPSO-TVAC	RPPSO	BPSO	PSO-ITC
F1	9.93E+02	4.65E+02	2.71E+02	2.48E+02	2.38E+02	1.32E+02	3.12E+02	2.40E+02	1.79E+02	1.44E+02
F2	1.03E+03	1.48E+03	2.60E+02	2.36E+02	2.28E+02	1.38E+02	2.29E+02	2.30E+02	2.28E+02	1.50E+02
F3	9.92E+02	7.66E+02	2.19E+02	1.95E+02	1.87E+02	1.27E+02	1.89E+02	1.89E+02	1.87E+02	1.30E+02
F4	4.12E+02	6.65E+02	3.52E+02	3.07E+02	1.89E+02	8.97E+01	3.20E+02	2.83E+02	1.75E+02	1.22E+02
F5	1.06E+03	1.24E+03	2.84E+02	2.61E+02	2.52E+02	1.43E+02	2.54E+02	2.54E+02	2.53E+02	1.65E+02
F6	1.03E+03	8.03E+02	2.58E+02	2.35E+02	2.26E+02	1.31E+02	2.28E+02	2.28E+02	2.26E+02	1.46E+02
F7	9.94E+02	7.61E+02	2.23E+02	1.99E+02	1.90E+02	1.31E+02	1.91E+02	1.93E+02	1.92E+02	1.35E+02
F8	7.43E+02	8.93E+02	6.57E+02	6.33E+02	6.24E+02	4.23E+02	6.26E+02	6.37E+02	6.02E+02	6.77E+02
F9	1.05E+03	7.90E+02	2.75E+02	2.50E+02	2.41E+02	1.35E+02	2.43E+02	2.44E+02	2.43E+02	1.50E+02
F10	4.77E+02	1.02E+03	3.92E+02	3.65E+02	3.57E+02	1.57E+02	3.60E+02	3.58E+02	3.30E+02	1.93E+02
F11	1.05E+03	8.28E+02	2.74E+02	2.50E+02	2.41E+02	1.31E+02	2.43E+02	2.44E+02	2.41E+02	1.50E+02
F12	3.12E+02	4.14E+02	2.59E+02	2.45E+02	2.42E+02	1.29E+02	2.42E+02	2.44E+02	1.71E+02	1.06E+02
F13	3.32E+02	4.23E+02	2.82E+02	2.69E+02	2.69E+02	1.63E+02	2.65E+02	2.75E+02	1.92E+02	1.32E+02
F14	4.87E+02	6.81E+02	4.42E+02	4.24E+02	4.24E+02	3.44E+02	4.25E+02	4.26E+02	3.39E+02	3.01E+02
F15	4.39E+02	7.59E+02	3.92E+02	3.75E+02	3.76E+02	3.55E+02	3.76E+02	3.77E+02	2.95E+02	2.89E+02
F16	1.04E+03	1.43E+03	2.67E+02	2.44E+02	2.37E+02	1.56E+02	2.37E+02	2.38E+02	1.71E+02	1.62E+02
F17	3.11E+02	8.21E+02	2.58E+02	2.45E+02	2.42E+02	3.60E+02	2.43E+02	2.42E+02	1.74E+02	1.26E+02
F18	1.32E+03	1.71E+03	5.56E+02	5.23E+02	5.25E+02	4.47E+02	5.27E+02	5.28E+02	4.58E+02	5.95E+02
F19	1.37E+03	3.01E+03	6.17E+02	5.82E+02	5.85E+02	4.60E+02	5.89E+02	5.87E+02	5.12E+02	6.24E+02
F20	7.65E+02	9.61E+02	6.98E+02	6.60E+02	6.63E+02	4.21E+02	6.76E+02	6.66E+02	5.15E+02	6.12E+02

the best searching accuracy, as shown in functions F15 and F16 [illustrated by Fig. A1(o) and (p), respectively]. For example, FlexiPSO achieves smaller SP values than the PSO-ITC in the functions F15 and F16. However, as shown in Table 3 as well as in Fig. A1(o) and (p), the PSO-ITC achieves a significantly better F_{mean} value than FlexiPSO, and the former reaches the solution with the predefined ϵ earlier than the latter. Finally, the PSO-ITC demonstrates a competitive convergence speed in complex problems, as shown in Fig. A1(r) and (t). Specifically, the convergence speed of the PSO-ITC in functions F18 and F20 is significantly faster than that of its peers during the early stage of the optimization.

4.6. Comparison of mean computational time

As shown in the previous SP analysis, the proposed PSO-ITC is more computationally efficient than its peer algorithms. To further verify this finding, we conducted an experiment to compute the

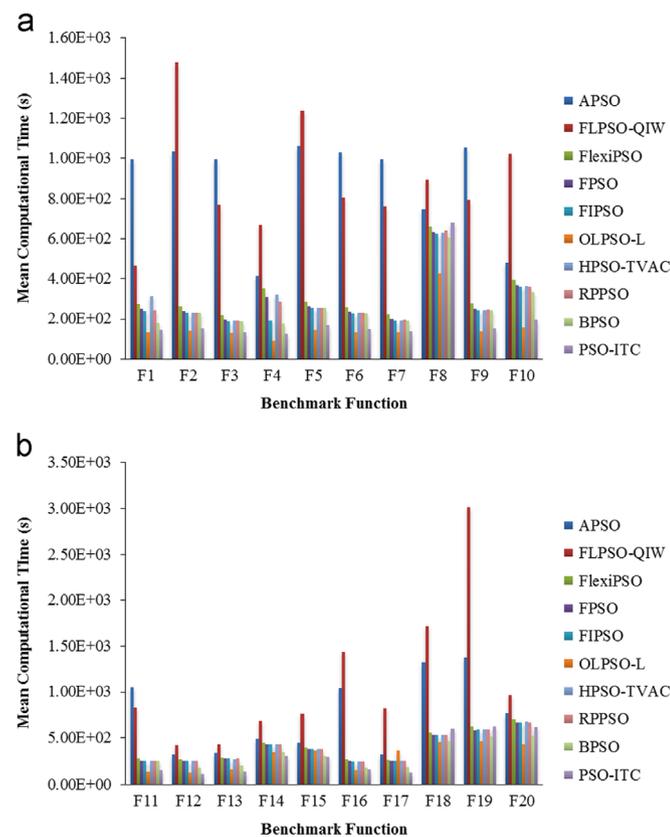


Fig. 11. Mean computation time in CPU cycle (in seconds) for 50-D functions (a) F1–F10. (b) Mean computation time in CPU cycle (in seconds) for 50-D functions (b) F11–F20.

Table 5 Simulation results achieved by BPSO, PSO-ITC1, PSO-ITC2, PSO-ITC3, and PSO-ITC in 50-D problems.

Algorithm		Conventional problems (F1–F8)	Rotated problems (F9–F13)	Shifted problems (F14–F17)	Complex problems (F18–F20)	Overall results
BPSO	#BMF (#GO)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
	%Improve	–	–	–	–	–
PSO-ITC1	#BMF (#GO)	8 (7)	3 (3)	1 (1)	0 (0)	12 (11)
	%Improve	98.0668	89.9635	99.9132	98.7273	96.5093
PSO-ITC2	#BMF (#GO)	7 (7)	3 (3)	1 (1)	1 (0)	12 (11)
	%Improve	97.6837	91.3234	100.0000	99.5629	96.8457
PSO-ITC3	#BMF (#GO)	7 (7)	4 (4)	1 (1)	0 (0)	12 (12)
	%Improve	98.0565	91.4455	99.8533	98.5780	96.8413
PSO-ITC	#BMF (#GO)	7 (7)	5 (4)	4 (1)	2 (0)	18 (12)
	%Improve	97.4250	91.9385	100.0000	99.6095	96.8891

mean computational time (t_{mean}) or runtime of all involved PSO variants on the employed 20 benchmarks. Similar to the previous experiments, the dimensionality level of 50 was considered. The mean computational times for all the algorithms were measured on a PC Intel Core 2 Duo 2.13 GHz with 3.50 GB RAM that runs Windows XP with Matlab implementation. The results are summarized in Table 4 and Fig. 11.

The results in Table 4 and Fig. 11 show that the involved PSO variants exhibit diverse t_{mean} values. In general, both APSO and FLPSO-QIW appear to have higher computational overhead with respect to the other algorithms, as these two algorithms produce 6 and 14 worst t_{mean} values out of the 20 employed benchmarks, respectively. Meanwhile, we observe that the computation overheads of the OLPSO-L and the proposed PSO-ITC are the lowest in a majority of the employed functions. Specifically, OLPSO-L achieves 15 best and 2 s best t_{mean} values, whereas PSO-ITC records 5 best and 11 s best values out of the 20 employed benchmarks. In most of the employed benchmarks, the differences between t_{mean} values produced by OLPSO-L and PSO-ITC are relatively insignificant, which suggests that these two algorithms have a comparable computational overhead. The excellent performance of PSO-ITC in terms of t_{mean} and previously reported SP values confirms that the proposed algorithm is indeed more computationally efficient than its peers.

4.7. Effect of different strategies

The proposed PSO-ITC consists of three strategies, namely ITC module, NS operator, and EBLs. The contribution of each strategy in improving the PSO-ITC's overall performance is worth investigating. To perform this study, we decompose the complete PSO-ITC into (1) PSO-ITC with ITC module only (PSO-ITC1), (2) PSO-ITC with ITC module and EBLs operator (PSO-ITC2), and (3) PSO-ITC with ITC module and NS operator (PSO-ITC3). We compare the F_{mean} values produced by PSO-ITC1, PSO-ITC2, PSO-ITC3, and PSO-ITC with those produced by BPSO. The degree of improvement of each PSO-ITC variant over the BPSO is expressed in terms of percentage improvement (%Improve) calculated as follows (Lam et al., 2012):

$$\%Improve = \frac{F_{mean}(BPSO) - F_{mean}(\zeta)}{|F_{mean}(BPSO)|} \times 100\% \quad (8)$$

where ζ denotes PSO-ITC1, PSO-ITC2, PSO-ITC3, or PSO-ITC. If ζ outperforms BPSO, the %Improve has a positive value. Otherwise, the %Improve is assigned a negative value. The (1) number of best F_{mean} values (#BMF), (2) number of global optima found (#GO), and (3) %Improve values produced by all compared algorithms in the conventional, rotated, shifted, and complex problems are presented in Table 5. The last column of Table 5 summarizes the overall results, that is, the total #BMF, total #GO, and average %Improve values achieved by all involved algorithms.

As reported in Table 5, the performance of all PSO-ITC variants improved significantly compared with that of BPSO, which implies that any strategy, that is, ITC module, NS operator, or EBLs, indeed helps to enhance the algorithm's searching accuracy. Among all these PSO-ITC variants, the complete PSO-ITC achieves the largest average %Improve value, followed by PSO-ITC2, PSO-ITC3, and PSO-ITC1.

Specifically, Table 5 shows that PSO-ITC1 exhibits the least improvement, as it is only able to successfully solve seven (out of eight) conventional problems. The searching accuracy of PSO-ITC1 deteriorates when it is employed to solve rotated, shifted, and complex problems. This deterioration implies that the ITC module alone is insufficient to help the particles to escape from the local minima in other classes of problems. In contrast, the performance of PSO-ITC2 and PSO-ITC3 is good in shifted and rotated problems, respectively. We speculate that the combination of ITC module and EBLs is more effective in tracking the shifted global optima, whereas the combination of ITC module and NS operator plays a major role in solving non-separable problems. However, the performance of these two variants in complex problems (F18–F20) is still unsatisfactory, which implies that the combination of ITC module with any of the NS or EBLs strategies is still insufficient in handling problems with a more complex fitness landscape. Finally, we observe that the complete PSO-ITC outperforms the other improved variants in all types of problems. Specifically, it successfully achieves seven (out of eight), five (out of five), four (out of four), and two (out of three) best F_{mean} values in the conventional, rotated, shifted, and complex problems, respectively. Such an observation is reasonable, as the integration of ITC module with EBLs and NS operator is sufficient for solving the shifted and rotated problems, respectively. The superior performance of the complete PSO-ITC in all tested problems implies that the aforementioned strategies are integrated effectively. None of the contributions of the aforementioned strategies is compromised when the PSO-ITC is used to solve different types of problems.

4.8. Comparison with other state-of-the-art metaheuristic search algorithms

In this section, we compare the proposed PSO-ITC with some cutting-edge metaheuristic search (MS) algorithms, as these MS algorithms are capable of solving optimization problems. Specifically,

we compare our PSO-ITC with real-coded chemical reaction optimization (RCCRO) (Lam et al., 2012), differential evolution with strategy adaption (SaDE) (Qin et al., 2009), orthogonal learning-based artificial bee colony (OCABC) (Gao et al., 2013), group search optimizer (GSO) (He et al., 2009), real-coded biogeography-based optimization (RCBBO) (Gong et al., 2010), and covariance matrix adaptation evolution strategy (CMAES) (Hansen and Ostermeier, 2001). RCCO is a real-coded version of the chemical reaction optimization (Lam and Li, 2010) that was developed based on an analogy to a chemical reaction. SaDE is an improved variant of the differential evolution (DE) (Storn and Price, 1997). OCABC employs orthogonal learning into the artificial bee colony (ABC) to achieve better searching performance. The development of GSO is motivated by an animal's searching behavior, i.e., the producer–scrounger (PS) model. RCBBO is the real-coded version of biogeography-based optimization (Simon, 2008), inspired by the geographical distribution of biological organisms. The CMAES is an improved evolutionary strategy (Beyer and Schwefel, 2002) with the restart and increasing population size mechanism.

To compare the proposed PSO-ITC with the aforementioned MS algorithms, we simulate various 30 dimensional conventional problems and summarize the F_{mean} values produced by all the algorithms in Table 6. All the results of the compared MS algorithms are extracted from their corresponding literature. Thus, we assign the F_{mean} value as “NA” if the algorithm's F_{mean} value for a particular benchmark is not available in its original literature. Table 6 shows that PSO-ITC has the most superior searching accuracy as it successfully solves almost all tested functions. Specifically, the PSO-ITC locates the global optima of eight out of 10 problems, that is, two times better than the second ranked OCABC. Also, the PSO-ITC is the only algorithm to find the global optima for Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, and Ackley functions.

4.9. Comparative study on two real-world engineering design problems

In this section, we study the feasibility of the proposed PSO-ITC in engineering applications. More precisely, we investigate the performance of the PSO-ITC over two real-world engineering design problems, namely (1) the gear train design problem (Sandgren, 1990) and (2) the spread spectrum radar polyphase

Table 6 Comparison of the PSO-ITC with other state-of-the art MS algorithms in 30-D problems.

Function		RCCRO	SaDE	OCABC	GSO	RCBBO	CMAES	PSO-ITC
Sphere	F_{mean}	6.43E-07	3.28E-20	4.32E-43	1.95E-08	1.39E-03	6.09E-29	0.00E+00
	SD	(2.09E-07)	(3.63E-20)	(8.16E-43)	(1.16E-08)	(5.50E-04)	(1.55E-29)	(0.00E+0.0)
Schwefel 2.22	F_{mean}	2.19E-03	3.51E-25	1.17E-22	3.70E-05	7.99E-02	3.48E-14	0.00E+00
	SD	(4.34E-04)	(2.74E-25)	(7.13E-23)	(8.62E-05)	(1.44E-02)	(4.03E-15)	(0.00E+0.0)
Schwefel 1.2	F_{mean}	2.97E-07	NA	NA	5.78E+00	2.27E+01	1.51E-26	0.00E+00
	SD	(1.15E-07)			(3.68E+00)	(1.03E+01)	(3.64E-27)	(0.00E+0.0)
Schwefel 2.21	F_{mean}	9.32E-03	NA	5.67E-01	1.08E-01	3.09E-02	3.99E-15	0.00E+00
	SD	(3.66E-03)		(2.73E-01)	(3.99E-02)	(7.27E-03)	(5.31E-16)	(0.00E+0.0)
Rosenbrock	F_{mean}	2.71E+01	2.10E+01	7.89E-01	4.98E+01	5.54E+01	5.58E-01	2.55E+01
	SD	(3.43E+01)	(7.8E+00)	(6.27E-01)	(3.02E+01)	(3.52E+01)	(1.39E+00)	(1.35E+00)
Step	F_{mean}	0.00E+00	5.07E+01	0.00E+00	1.60E-02	0.00E+00	7.00E-02	0.00E+00
	SD	(0.00E+00)	(1.34E+01)	(0.00E+00)	(1.33E-01)	(0.00E+00)	(2.93E-01)	(0.00E+0.0)
Quartic	F_{mean}	5.41E-03	4.86E-03	4.39E-03	7.38E-02	1.75E-02	2.21E-01	1.43E+01
	SD	(2.99E-03)	(5.21E-04)	(2.03E-03)	(9.26E-02)	(6.43E-03)	(8.65E-02)	(1.76E+01)
Rastrigin	F_{mean}	9.08E-04	2.43E+00	0.00E+00	1.02E+00	2.62E-02	4.95E+01	0.00E+00
	SD	(2.88E-04)	(1.60E+00)	(0.00E+00)	(9.51E-01)	(9.76E-03)	(1.23E+01)	(0.00E+0.0)
Ackley	F_{mean}	1.94E-03	3.81E-06	5.32E-15	2.66E-05	2.51E-02	4.61E+00	0.00E+00
	SD	(4.19E-04)	(8.26E-07)	(1.82E-15)	(3.08E-05)	(5.51E-03)	(8.73E+00)	(0.00E+0.0)
Grienwank	F_{mean}	1.12E-02	2.52E-09	0.00E+00	3.08E-02	4.82E-01	7.40E-04	0.00E+00
	SD	(1.62E-02)	(1.24E-08)	(0.00E+00)	(3.09E-02)	(8.49E-02)	(2.39E-03)	(0.00E+0.0)
w/t/l		8/1/1	6/0/2	4/3/2	9/0/1	8/1/1	8/0/2	
#BMF		1	0	4	0	1	1	8

code design problem (Das and Suganthan, 2010). The descriptions and mathematical models of these two engineering design problems are presented in the following subsections.

4.9.1. Gear train design problem

The gear train design problem aims to optimize the gear ratio for a compound gear train that contains three gears. The objective function of this problem is represented as (Sandgren, 1990)

$$f(x) = \left(\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4} \right)^2 \tag{9}$$

where $x_i \in [12, 60]$, $i=1, 2, 3, 4$. As shown in Eq. (9), the gear ratio must be as close as possible to $1/6.931$ to minimize the cost of gear ratio in the gear train. The bound constraint of this problem restricts the number of teeth of each gear in the range of 12–60.

4.9.2. Spread spectrum radar polyphase code design problem

The spread spectrum radar polyphase code design problem plays an important role in the radar system design. This problem has no polynomial time solution, and its formal statement is defined as follows (Das and Suganthan, 2010):

$$\text{Global min } f(x) = \max \{ \varphi_1(X), \dots, \varphi_{2m}(X) \} \tag{10}$$

where $X = \{ (x_1, \dots, x_D) \in R^D | 0 \leq x_j \leq 2\pi \}$ and $m = 2D - 1$, with

$$\begin{aligned} \varphi_{2i-1}(X) &= \sum_{j=i}^D \cos \left(\sum_{k=|2i-j-1|-1}^j x_k \right), \quad i = 1, 2, \dots, D \\ \varphi_{2i}(X) &= 0.5 + \sum_{j=i+1}^D \cos \left(\sum_{k=|2i-j-1|-1}^j x_k \right), \quad i = 1, 2, \dots, D-1 \\ \varphi_{m+i}(X) &= -\varphi_i(X), \quad i = 1, 2, \dots, m \end{aligned} \tag{11}$$

4.9.3. Experimental settings for the two real-world engineering design problems

In this study, all the 10 PSO variants employed in the previous experiment are tested in these two engineering design problems. The parameter settings of each algorithm remain the same as in the previous experiment. For the gear train design problem, the population size (S) and the maximum fitness evaluation numbers (FE_{max}) are set to 10 and $3.00E+04$, respectively. Meanwhile, we

Table 7
Experimental settings for the two real-world engineering design problems.

Parameters	Gear train design	Spread spectrum radar polyphase design
D	4	20
S	10	20
FE_{max}	$3.00E+04$	$2.00E+05$

Table 8
Comparison between two real-world engineering design problems.

Algorithm	Gear train design			Spread spectrum radar polyphase design		
	$F_{mean} \pm SD$	h	t_{mean} (s)	$F_{mean} \pm SD$	h	t_{mean} (s)
APSO	$1.28E-08 \pm 1.70E-08$	+	$1.11E+02$	$1.33E+00 \pm 1.92E-01$	+	$4.96E+02$
FLPSO-QIW	$3.34E-10 \pm 5.78E-10$	-	$1.17E+02$	$1.02E+00 \pm 6.88E-02$	-	$9.63E+02$
FlexiPSO	$2.36E-09 \pm 5.78E-10$	-	$9.24E+01$	$1.22E+00 \pm 2.48E-01$	+	$2.55E+02$
FPSO	$7.48E-07 \pm 2.43E-06$	=	$9.21E+01$	$1.13E+00 \pm 1.30E-01$	+	$2.48E+02$
FIPSO	$5.59E-09 \pm 9.39E-09$	=	$9.14E+01$	$1.04E+00 \pm 1.47E-01$	-	$2.75E+02$
OLPSO-L	$6.74E-09 \pm 1.25E-08$	+	$3.83E+01$	$1.27E+00 \pm 1.97E-01$	+	$1.80E+02$
HPSO-TVAC	$1.90E-08 \pm 3.70E-08$	=	$9.16E+01$	$1.21E+00 \pm 1.94E-01$	+	$3.68E+02$
RPPSO	$2.43E-07 \pm 3.79E-07$	+	$9.16E+01$	$1.10E+00 \pm 1.73E-01$	+	$5.03E+02$
BPSO	$1.04E-04 \pm 5.67E-04$	=	$6.31E+01$	$1.21E+00 \pm 1.70E-01$	+	$3.62E+02$
PSO-ITC	$4.25E-09 \pm 4.55E-09$	=	$4.15E+01$	$1.10E+00 \pm 1.23E-01$	=	$2.66E+02$

consider the spread spectrum radar polyphase code design problem for $D=20$. The S and FE_{max} of this problem are set to 20 and $2.00E+05$, respectively. The experimental settings for these two problems are summarized in Table 7.

4.9.4. Simulation results of the two real-world engineering design problems

The simulation results over 30 independent runs for the gear train design and spread spectrum radar polyphase design problems are presented in Table 8, which contains the values of mean fitness (F_{mean}), standard deviation (SD), t -test's result (h), and mean computational time (t_{mean}).

For the gear train design problem, almost all PSO variants exhibit excellent searching accuracy, except for the BPSO. Among these PSO variants, our proposed PSO-ITC achieves the third best F_{mean} value, that is, its searching accuracy in solving the gear train design problem outperforms seven other peers, namely APSO, FPSO, FIPSO, OLPSO-L, HPSO-TVAC, RPPSO, and BPSO. Although the F_{mean} value produced by the PSO-ITC in the gear train design problem is slightly inferior to that of FLPSO-QIW and FlexiPSO, the former is at least two times more superior to the latter two in terms of computational overhead (represented by t_{mean}). Meanwhile, all involved PSO variants have a similar searching accuracy in tackling the spread spectrum radar polyphase design problem, as the produced F_{mean} values are relatively similar. As shown in Table 8, the searching accuracy exhibited by the PSO-ITC in this problem is competitive because it achieves the third best F_{mean} values. In terms of searching accuracy, the performance deviation between the PSO-ITC (i.e., $F_{mean}=1.10E+00$) and the first-ranked FLPSO-QIW (i.e., $F_{mean}=1.02E+00$) is relatively insignificant, as the latter is only 1.08 times better than the former. On the other hand, the mean computational time required by our proposed PSO-ITC (i.e., $t_{mean}=2.66E+02$ s) to solve the spread spectrum radar polyphase design problem is significantly less than that of FLPSO-QIW (i.e., $t_{mean}=9.63E+02$ s). More precisely, the proposed PSO-ITC is 3.62 times better than the FLPSO-QIW in terms of computational overhead. Based on the simulation results in Table 8, we conclude that our proposed PSO-ITC achieves a better trade-off between the produced F_{mean} and the t_{mean} values compared with its peers. The prominent performance of the PSO-ITC in terms of searching accuracy and computational overhead proves that its application is indeed feasible in real-world engineering problems.

4.10. Discussion

The simulation results of the benchmark and real-world problems indicate that our proposed PSO-ITC has superior searching accuracy, searching reliability, and convergence speed compared with the other nine well-established PSO variants and six

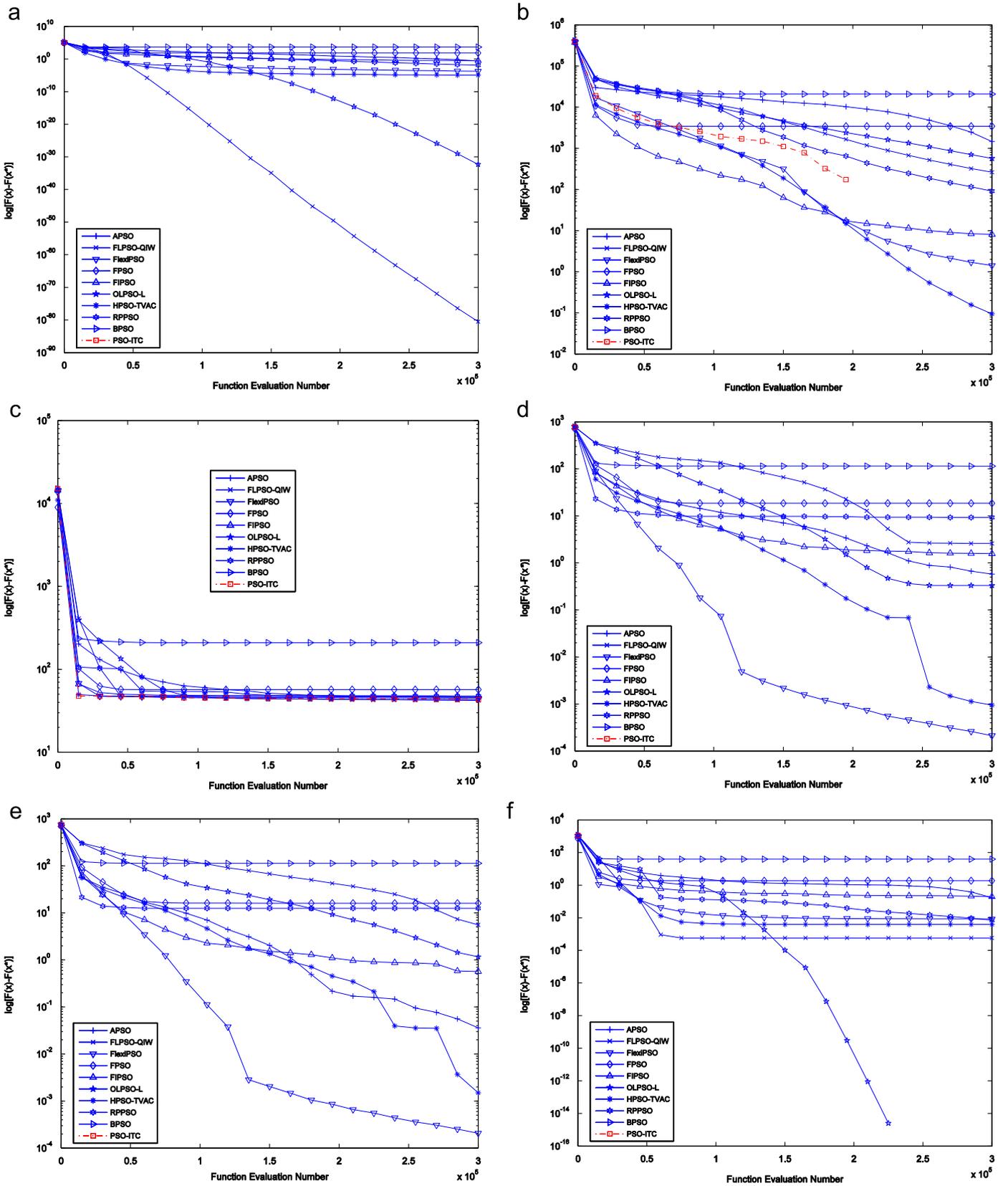


Fig. A1. Convergence curves of 50 dimensional problems: (a) F1 Sphere, (b) F2 Schwefel 1.2, (c) F3 Rosenbrock, (d) F4 Rastrigin, (e) F5 Noncontinuous Rastrigin, and (f) Griewank functions. Convergence curves of 50 dimensional problems: (g) F7 Ackley, (h) F8 Weierstrass, (i) F9 Rotated Sphere, (j) F10 Rotated Schwefel 1.2, (k) F11 Rotated Rosenbrock, and (l) F12 Rotated Rastrigin functions. Convergence curves of 50 dimensional problems: (m) F13 Rotated Griewank, (n) F14 Shifted Sphere, (o) F15 Shifted Rastrigin, (p) F16 Shifted Noncontinuous Rastrigin, (q) F17 Shifted Griewank, and (r) F18 Rotated Shifted Griewank functions. Convergence curves of 50 dimensional problems: (s) F19 Shifted Rotated High Conditioned Elliptic and (t) F20 Shifted Expanded Griewank Rosenbrock functions.

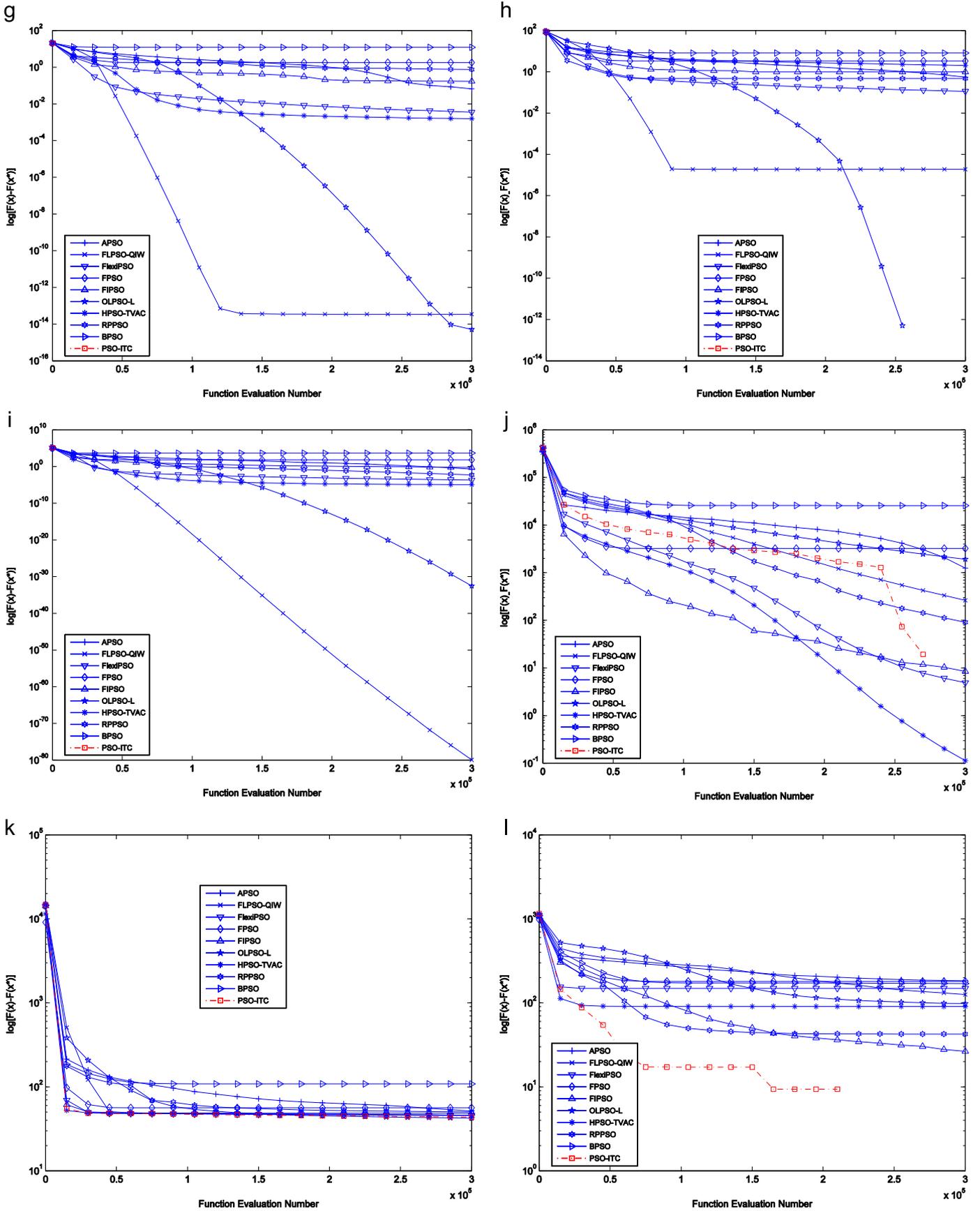


Fig. A1. (continued)

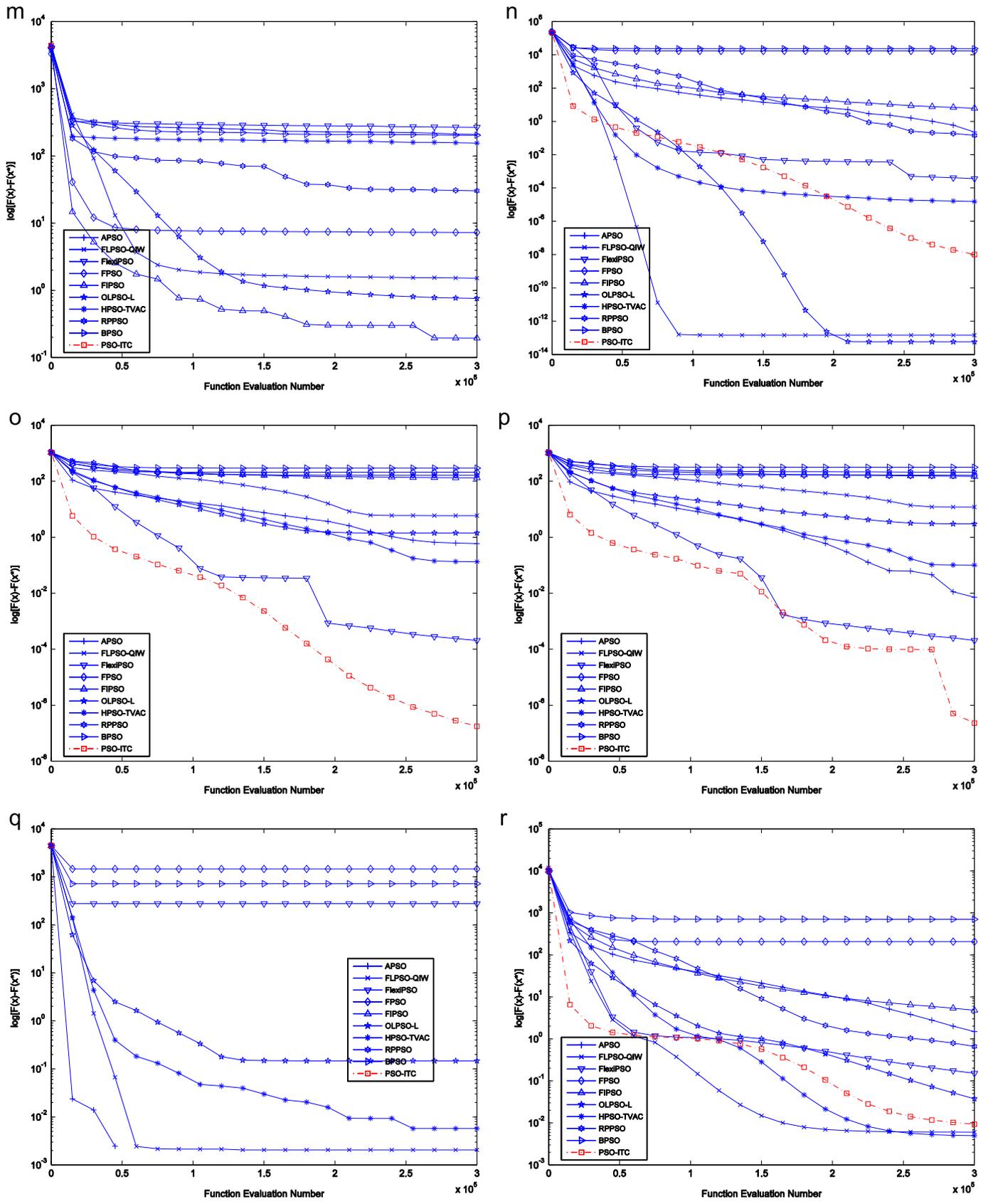


Fig. A1. (continued)

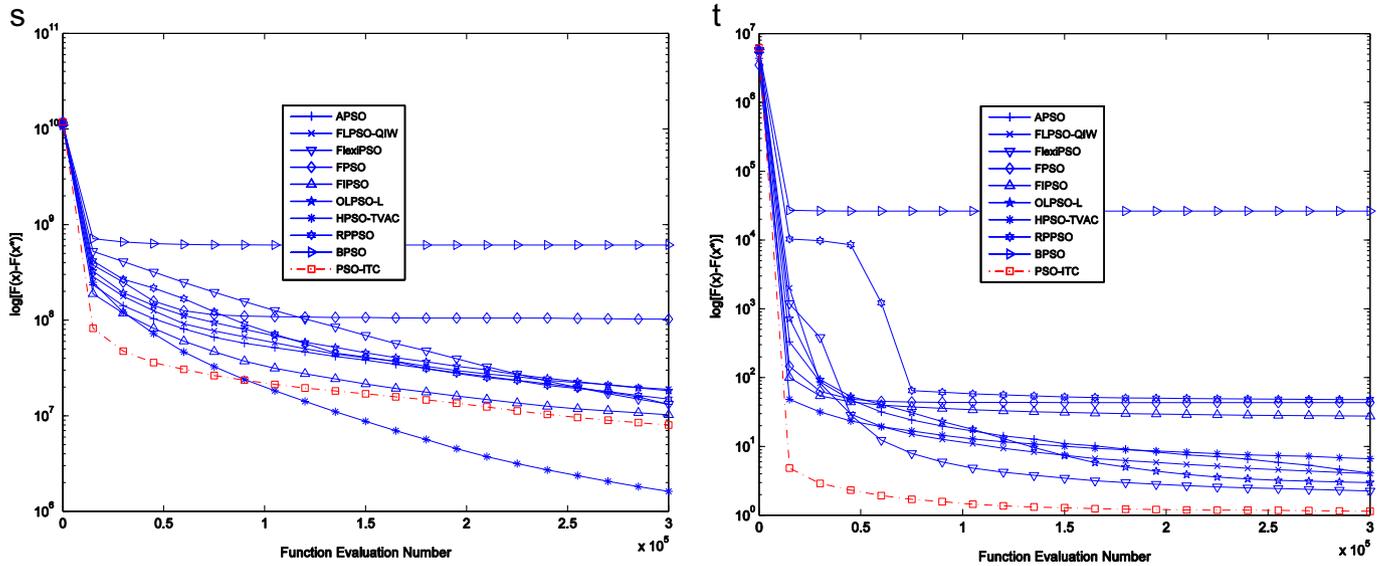


Fig. A1. (continued)

cutting-edge MS algorithms. The excellent performance of PSO-ITC is attributed to the two major contributions proposed in our work, namely the ITC module and the proposed learning framework. The ITC module aims to achieve better control of the exploitation/exploration searches of PSO-ITC particles by linearly increasing the particle's connectivity with time. The linearly increasing scheme is adopted in the ITC module as early studies (Kennedy, 1999; Kennedy and Mendes, 2002; Shi and Eberhart, 1998) revealed that particles at the early stage of optimization need to perform more exploration, which makes small topological connectivity preferable. At the later stage of optimization, particles should exploit the most promising explored region, which requires large topological connectivity. To prevent particle stagnation in the local optima, a shuffling mechanism is incorporated in the ITC module to offer a new searching direction for the particle if it fails to improve the global best fitness for z successive FEs.

The proposed learning framework consists of a new velocity update mechanism and a new NS operator. For the new velocity update mechanism, two exemplars of $c_{exp,i}$ and $s_{exp,i}$ are generated to update the velocity of particle i . Both exemplars are generated from particle i 's neighborhood through the roulette wheel selection to ensure that good quality exemplars are employed to guide particle i to a more prominent search space. If particle i fails to improve its personal best fitness when evolved through the new velocity update mechanism, the NS operator is triggered. In the NS operator, another exemplar, the $o_{exp,i}$ exemplar, is used to further evolve particle i . Unlike the $c_{exp,i}$ and $s_{exp,i}$ exemplars, the $o_{exp,i}$ exemplar is derived from the $c_{guide,i}$ and $s_{guide,i}$ guidance particles contributed by another particle's neighborhood. This mechanism establishes information exchange between the different neighborhoods that exist in the population, thereby allowing particle i to locate potentially better unexplored regions based on the useful information provided by other neighborhoods. In addition, when the particle is evolved through the new velocity update mechanism or the new NS operator, the particle can be attracted towards or repelled from its exemplar depending on its exemplars' fitness. This mechanism automatically assigns different search tasks (i.e., exploration and exploitation) to different particles in the population, thereby resolving the exploration/exploitation balancing issue.

Although the proposed PSO-ITC exhibited superior performance in the previously reported experiments, it is applicable only to unconstrained single-objective (SO) problems with continuous search space. More work needs to be done to further extend the applicability of the

proposed PSO-ITC to a more general class of optimization problems, including those with discrete and mixed search spaces as well as multiple-objective (MO) problems. For example, MO problems have a rather different perspective compared with SO problems because the former contains more than one objective that needs to be achieved simultaneously. Also, unlike SO problems which consist of only one global optimum, a set of solutions, namely the Pareto-optima set, are considered equally important in MO problems. In general, two main aspects need to be considered to adapt the proposed PSO-ITC for MO problems. First, the PSO-ITC needs to guide solutions toward the Pareto-frontier by employing strategies, such as Pareto-ranking or Pareto-sorting (Fonseca and Fleming, 1995). Second, some mechanisms, such as sharing or niche methods (Fonseca and Fleming, 1995), need to be incorporated in the PSO-ITC to ensure that a set of well-distributed solutions are generated across the Pareto-frontier. For more discussions of the extension of an optimization algorithm to facilitate its application in a more general class of optimization problems, refer Page et al. (2012).

In this paper, our main suggestion is to explore the possible benefits of combining the ITC module with the proposed learning framework in the context of PSO algorithms in solving the unconstrained SO optimization problem with continuous search space. Extensive experimental results obtained from our current study prove that the combination of the ITC module and the proposed learning strategy significantly enhances the searching performance of PSO in the aforementioned search space. In our future works, we will extend the applicability of PSO-ITC to a diverse class of optimization problems, such as discrete, mixed, and multi-objective search spaces.

5. Conclusion

In this paper, the PSO-ITC is proposed to solve unconstrained SO optimization problems with continuous search space. The ITC module is developed to vary the particle's topology connectivity during processing, thereby achieving better balance of the exploration/exploitation searches. In addition, a new learning framework, which consists of a new velocity update mechanism and a new NS operator, is incorporated into the PSO-ITC. Both aforementioned strategies aim to improve the searching accuracy of the algorithm by generating the more promising exemplars as the guidance particles. The simulation results reveal that the proposed PSO-ITC significantly outperforms its peers in terms of

searching accuracy, searching reliability, and computation cost. This superior performance implies that the increasing topology approach and the new learning framework are promising ways of enhancing the searching performance of PSO.

Acknowledgments

The authors express their sincere gratitude to the associate editor and the reviewers for their significant contributions to the improvement of the final paper. The authors would also like to thank Abdul Latiff Abdul Tawab, Ahmad Ahzam Latib, Nor Azhar Zabidin, and Amir Hamid for their technical supports. This research was supported by the Universiti Sains Malaysia (USM) Postgraduate Fellowship Scheme and the Postgraduate Research Grant Scheme (PRGS) entitled “Development of PSO Algorithm with Multi-Learning Frameworks for Application in Image Segmentation.”

Appendix A. Convergence curves for 50 dimensional problems

See Fig. A1.

References

- Alonso Zotes, F., Santos Peñas, M., 2012. Particle swarm optimisation of interplanetary trajectories from Earth to Jupiter and Saturn. *Engineering Applications of Artificial Intelligence* 25, 189–199.
- AlRashidi, M.R., El-Hawary, M.E., 2009. A survey of particle swarm optimization applications in electric power systems. *IEEE Transactions on Evolutionary Computation* 13, 913–918.
- Banks, A., Vincent, J., Anyakoha, C., 2007. A review of particle swarm optimization. Part I: background and development. *Natural Computing* 6, 467–484.
- Banks, A., Vincent, J., Anyakoha, C., 2008. A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Natural Computing* 7, 109–124.
- Bastos-Filho, C.J.A., Carvalho, D.F., Figueiredo, E.M.N., de Miranda, P.B.C., 2009. Dynamic clan particle swarm optimization. In: Ninth International Conference on Intelligent Systems Design and Applications (ISDA '09), Pisa, pp. 249–254.
- Beyer, H.-G., Schwefel, H.-P., 2002. Evolution strategies: a comprehensive introduction. *Natural Computing* 1, 3–52.
- Carvalho, D.F., Bastos-Filho, C.J.A., 2008. Clan particle swarm optimization. *IEEE Congress on Evolutionary Computation*, 2008. CEC, pp. 3044–3051.
- Chen, Y.-p., Peng, W.-C., Jian, M.-C., 2007. Particle swarm optimization with recombination and dynamic linkage discovery. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 37, 1460–1470.
- Clerc, M., Kennedy, J., 2002. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* 6, 58–73.
- Das, S., Suganthan, P.N., 2010. Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems. Nanyang Technological University, Singapore.
- del Valle, Y., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J.C., Harley, R.G., 2008. Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Transactions on Evolutionary Computation* 12, 171–195.
- dos Santos Coelho, L., Mariani, V.C., 2008. Particle swarm approach based on quantum mechanics and harmonic oscillator potential well for economic load dispatch with valve-point effects. *Energy Conversion and Management* 49, 3080–3085.
- Eberhart, R.C., Shi, Y., 2001. Particle swarm optimization: developments, applications and resources. In: Congress on Evolutionary Computation. Proceedings of the 2001, vol. 81, pp. 81–86.
- Fonseca, C.M., Fleming, P.J., 1995. An overview of evolutionary algorithms in multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 3, 1–16.
- Fu, Y., Ding, M., Zhou, C., 2012. Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 42, 511–526.
- Gao, W.-f., Liu, S.-y., Huang, L.-L., 2013. A novel artificial bee colony algorithm based on modified search equation and orthogonal learning. *IEEE Transactions on Cybernetics* 43, 1011–1024.
- Gong, W., Cai, Z., Ling, C.X., Li, H., 2010. A real-coded biogeography-based optimization with mutation. *Applied Mathematics and Computation* 216, 2749–2758.
- Guidice, V.G., Venayagamoorthy, G.K., 2003. Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks. In: *IEEE Proceedings of the Swarm Intelligence Symposium, 2003 (SIS '03)*, pp. 110–117.
- Hansen, N., Ostermeier, A., 2001. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation* 9, 159–195.
- He, S., Wu, Q.H., Saunders, J.R., 2009. Group search optimizer: an optimization algorithm inspired by animal searching behavior. *IEEE Transactions on Evolutionary Computation* 13, 973–990.
- Hicks, C.R., 1993. *Fundamental Concepts in the Design of Experiments*, fourth ed. Saunders College Publishing, Texas.
- Ho, S.-Y., Lin, H.-S., Liauh, W.-H., Ho, S.-J., 2008. OPSO: orthogonal particle swarm optimization and its application to task assignment problems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 38, 288–298.
- Holden, N., Freitas, A.A., 2008. A hybrid PSO/ACO algorithm for discovering classification rules in data mining. *Journal of Artificial Evolution and Applications* 2008, 1–11.
- Hsieh, S.-T., Sun, T.-Y., Liu, C.-C., Tsai, S.-J., 2009. Efficient population utilization strategy for particle swarm optimizer. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 39, 444–456.
- Huang, C.-J., Chuang, Y.-T., Hu, K.-W., 2009. Using particle swarm optimization for QoS in ad-hoc multicast. *Engineering Applications of Artificial Intelligence* 22, 1188–1193.
- Huang, H., Qin, H., Hao, Z., Lim, A., 2012. Example-based learning particle swarm optimization for continuous optimization. *Information Sciences* 182, 125–138.
- Kathrada, M., 2009. The flexi-PSO: towards a more flexible particle swarm optimizer. *OPSEARCH* 46, 52–68.
- Kennedy, J., 1999. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In: *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 1933, p. 1938.
- Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: *Proceedings IEEE International Conference on Neural Networks*, vol. 1944, pp. 1942–1948.
- Kennedy, J., Mendes, R., 2002. Population structure and particle swarm performance. In: *Proceedings of IEEE Congress on Evolutionary Computation (CEC '02)*, pp. 1671–1676.
- Kiranyaz, S., Ince, T., Yildirim, A., Gabbouj, M., 2010. Fractional particle swarm optimization in multidimensional search space. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 40, 298–319.
- Lam, A.Y.S., Li, V.O.K., 2010. Chemical-reaction-inspired metaheuristic for optimization. *IEEE Transactions on Evolutionary Computation* 14, 381–399.
- Lam, A.Y.S., Li, V.O.K., Yu, J.J.Q., 2012. Real-coded chemical reaction optimization. *IEEE Transactions on Evolutionary Computation* 16, 339–353.
- Leu, M.-S., Yeh, M.-F., 2012. Grey particle swarm optimization. *Applied Soft Computing* 12, 2985–2996.
- Liang, J., Suganthan, P.N., 2005. Dynamic multi-swarm particle swarm optimizer. In: *IEEE Proceedings of Swarm Intelligence Symposium (SIS 2005)*, pp. 124–129.
- Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S., 2006. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation* 10, 281–295.
- Lin, C.-J., Chen, C.-H., Lin, C.-T., 2009. A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 39, 55–68.
- Liu, D., Tan, K.C., Goh, C.K., Ho, W.K., 2007. A multiobjective memetic algorithm based on particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 37, 42–50.
- Liu, L., Liu, W., Cartes, D.A., 2008. Particle swarm optimization-based parameter identification applied to permanent magnet synchronous motors. *Engineering Applications of Artificial Intelligence* 21, 1092–1100.
- Mariani, V.C., Duck, A.R.K., Guerra, F.A., Coelho, L.D.S., Rao, R.V., 2012. A chaotic quantum-behaved particle swarm approach applied to optimization of heat exchangers. *Applied Thermal Engineering* 42, 119–128.
- Marinakis, Y., Marinaki, M., 2013. A hybridized particle swarm optimization with expanding neighborhood topology for the feature selection problem. In: Blesa, M., Blum, C., Festa, P., Roli, A., Sampels, M. (Eds.), *Hybrid Metaheuristics*. Springer, Berlin, Heidelberg, pp. 37–51.
- Marinakis, Y., Marinaki, M., Dounias, G., 2010. A hybrid particle swarm optimization algorithm for the vehicle routing problem. *Engineering Applications of Artificial Intelligence* 23, 463–472.
- Mendes, R., Kennedy, J., Neves, J., 2004. The fully informed particle swarm: simpler, maybe better. *IEEE Transactions on Evolutionary Computation* 8, 204–210.
- Mirjalili, S., Mohd Hashim, S.Z., Moradian Sardroudi, H., 2012. Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Applied Mathematics and Computation* 218, 11125–11137.
- Modares, H., Alfi, A., Naghibi Sistani, M.-B., 2010. Parameter estimation of bilinear systems based on an adaptive particle swarm optimization. *Engineering Applications of Artificial Intelligence* 23, 1105–1111.
- Montes de Oca, M.A., Stutzle, T., Birattari, M., Dorigo, M., 2009. Frankenstein's PSO: a composite particle swarm optimization algorithm. *Evolutionary Computation*, *IEEE Transactions on* 13, 1120–1132.
- Montes de Oca, M.A., Stutzle, T., Van den Eenden, K., Dorigo, M., 2011. Incremental social learning in particle swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 41, 368–384.
- Mousa, A.A., El-Shorbagy, M.A., Abd-El-Wahed, W.F., 2012. Local search based hybrid particle swarm optimization algorithm for multiobjective optimization. *Swarm and Evolutionary Computation* 3, 1–14.

- Nasir, M., Das, S., Maity, D., Sengupta, S., Halder, U., Suganthan, P.N., 2012. A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization. *Information Sciences* 209, 16–36.
- Neyestani, M., Farsangi, M.M., Nezamabadi-pour, H., 2010. A modified particle swarm optimization for economic dispatch with non-smooth cost functions. *Engineering Applications of Artificial Intelligence* 23, 1121–1126.
- Omkar, S.N., Venkatesh, A., Mudigere, M., 2012. MPI-based parallel synchronous vector evaluated particle swarm optimization for multi-objective design optimization of composite structures. *Engineering Applications of Artificial Intelligence* 25, 1611–1627.
- Özbakır, L., Delice, Y., 2011. Exploring comprehensible classification rules from trained neural networks integrated with a time-varying binary particle swarm optimizer. *Engineering Applications of Artificial Intelligence* 24, 491–500.
- Page, S.F., Chen, S., Harris, C.J., White, N.M., 2012. Repeated weighted boosting search for discrete or mixed search space and multiple-objective optimisation. *Applied Soft Computing* 12, 2740–2755.
- Paoli, A., Melgani, F., Pasolli, E., 2009. Clustering of hyperspectral images based on multiobjective particle swarm optimization. *IEEE Transactions on Geoscience and Remote Sensing* 47, 4175–4188.
- Parsopoulos, K.E., Vrahatis, M.N., 2002. Recent approaches to global optimization problems through Particle Swarm Optimization. *Natural Computing* 1, 235–306.
- Pontes, M.R., Neto, F.B.L., Bastos-Filho, C.J.A., 2011. Adaptive clan particle swarm optimization. In: *IEEE Symposium on Swarm Intelligence (SIS)*, pp. 1–6.
- Qin, A.K., Huang, V.L., Suganthan, P.N., 2009. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation* 13, 398–417.
- Ratnaweera, A., Halgamuge, S.K., Watson, H.C., 2004. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation* 8, 240–255.
- Sahoo, N.C., Ganguly, S., Das, D., 2011. Simple heuristics-based selection of guides for multi-objective PSO with an application to electrical distribution system planning. *Engineering Applications of Artificial Intelligence* 24, 567–585.
- Sahoo, N.C., Ganguly, S., Das, D., 2012. Fuzzy-Pareto-dominance driven possibilistic model based planning of electrical distribution systems using multi-objective particle swarm optimization. *Expert Systems with Applications* 39, 881–893.
- Sakthivel, V.P., Bhuvaneshwari, R., Subramanian, S., 2010. Multi-objective parameter estimation of induction motor using particle swarm optimization. *Engineering Applications of Artificial Intelligence* 23, 302–312.
- Salomon, R., 1996. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. *Biosystems* 39, 263–278.
- Sandgren, E., 1990. Nonlinear integer and discrete programming in mechanical design optimization. *Journal of Mechanical Design* 112, 223–229.
- Sarath, K.N.V.D., Ravi, V., 2013. Association rule mining using binary particle swarm optimization. *Engineering Applications of Artificial Intelligence* 26, 1832–1840.
- Sharma, K.D., Chatterjee, A., Rakshit, A., 2009. A hybrid approach for design of stable adaptive fuzzy controllers employing Lyapunov theory and particle swarm optimization. *IEEE Transactions on Fuzzy Systems* 17, 329–342.
- Shi, Y., Eberhart, R., 1998. A modified particle swarm optimizer. In: *Proceedings IEEE World Congress on Computational Intelligence*, pp. 69–73.
- Shih, T.-F., 2006. Particle swarm optimization algorithm for energy-efficient cluster-based sensor networks. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E89-A, 1950–1958.
- Simon, D., 2008. Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation* 12, 702–713.
- Storn, R., Price, K., 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, 341–359.
- 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*.
- Sun, J., Chen, W., Fang, W., Wun, X., Xu, W., 2012. Gene expression data analysis with the clustering method based on an improved quantum-behaved Particle Swarm Optimization. *Engineering Applications of Artificial Intelligence* 25, 376–391.
- Sun, J., Fang, W., Wu, X., Xie, Z., Xu, W., 2011. QoS multicast routing using a quantum-behaved particle swarm optimization algorithm. *Engineering Applications of Artificial Intelligence* 24, 123–131.
- Sun, J., Feng, B., Xu, W., 2004. Particle swarm optimization with particles having quantum behavior. In: *Proceedings of Congress on Evolutionary Computation (CEC2004)*, Portland, OR, USA, vol. 321, pp. 325–331.
- Tang, Y., Wang, Z., Fang, J.-a., 2011. Feedback learning particle swarm optimization. *Applied Soft Computing* 11, 4713–4725.
- van den Bergh, F., Engelbrecht, A.P., 2004. A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation* 8, 225–239.
- Van Der Merwe, D.W., Engelbrecht, A.P., 2003. Data clustering using particle swarm optimization. In: *Proceedings of the 2003 Congress on Evolutionary Computation (CEC '03)*, vol. 211, pp. 215–220.
- Wachowiak, M.P., Smolikova, R., Yufeng, Z., Zurada, J.M., Elmaghraby, A.S., 2004. An approach to multimodal biomedical image registration utilizing particle swarm optimization. *IEEE Transactions on Evolutionary Computation* 8, 289–301.
- Wang, C., Liu, Y., Zhao, Y., 2013. Application of dynamic neighborhood small population particle swarm optimization for reconfiguration of shipboard power system. *Engineering Applications of Artificial Intelligence* 26, 1255–1262.
- Wang, L., Singh, C., 2009. Reserve-constrained multiarea environmental/economic dispatch based on particle swarm optimization with local search. *Engineering Applications of Artificial Intelligence* 22, 298–307.
- Wang, Z., Sun, X., Zhang, D., 2007. A PSO-based classification rule mining algorithm. In: Huang, D.-S., Heutte, L., Loog, M. (Eds.), *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*. Springer, Berlin, Heidelberg, pp. 377–384.
- Yaghini, M., Khoshraftar, M.M., Fallahi, M., 2013. A hybrid algorithm for artificial neural network training. *Engineering Applications of Artificial Intelligence* 26, 293–301.
- Yang, F., Sun, T., Zhang, C., 2009. An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization. *Expert Systems with Applications* 36, 9847–9852.
- Yao, X., Liu, Y., Lin, G., 1999. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation* 3, 82–102.
- Zaslavsky, G.M., 1978. The simplest case of a strange attractor. *Physics Letters A* 69, 145–147.
- Zhan, Z.-H., Zhang, J., Li, Y., Chung, H.S.H., 2009. Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 39, 1362–1381.
- Zhan, Z.-H., Zhang, J., Li, Y., Shi, Y.-H., 2011. Orthogonal learning particle swarm optimization. *IEEE Transactions on Evolutionary Computation* 15, 832–847.
- Zhou, D., Gao, X., Liu, G., Mei, C., Jiang, D., Liu, Y., 2011. Randomization in particle swarm optimization for global search ability. *Expert Systems with Applications* 38, 15356–15364.