Swarm and Evolutionary Computation 24 (2015) 11–24

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Regular Paper

Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation

Nandar Lynn, Ponnuthurai Nagaratnam Suganthan *

School of Electrical & Electronics Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 20 December 2014
Received in revised form
2 May 2015
Accepted 26 May 2015
Available online 23 June 2015

Keywords:
Comprehensive learning (CL)
Exploration
Exploitation
Particle swarm optimization (PSO)
Heterogeneous

ABSTRACT

This paper presents a comprehensive learning particle swarm optimization algorithm with enhanced exploration and exploitation, named as "heterogeneous comprehensive learning particle swarm optimization" (HCLPSO). In this algorithm, the swarm population is divided into two subpopulations. Each subpopulation is assigned to focus solely on either exploration or exploitation. Comprehensive learning (CL) strategy is used to generate the exemplars for both subpopulations. In the exploration-subpopulation, the exemplars are generated by using personal best experiences of the particles in the exploration-subpopulation itself. In the exploitation-subpopulation, the personal best experiences of the entire swarm population are used to generate the exemplars. As the exploration-subpopulation does not learn from any particles in the exploitation-subpopulation, the diversity in the exploration-subpopulation can be retained even if the exploitation-subpopulation converges prematurely. The heterogeneous comprehensive learning particle swarm optimization algorithm is tested on shifted and rotated benchmark problems and compared with other recent particle swarm optimization algorithms to demonstrate superior performance of the proposed algorithm over other particle swarm optimization variants.

1. Introduction

In order to solve multimodal, discontinuous, non-convex and non-differentiable optimization problems, researchers have developed population-based algorithms such as particle swarm optimization (PSO), genetic algorithm (GA), differential evolution (DE), evolutionary strategy (ES), evolutionary programming (EP) and so on. In population-based algorithms, finding the optimal solution of a problem is based on two cornerstones, namely exploration: global search, exploring all over the search space to find promising regions and exploitation: local search, exploiting the identified promising regions to fine tune the search for the optimal solution. Good convergence behavior of a population-based algorithm can be obtained when an appropriate balance between exploration and exploitation processes is found. Emphasizing on exploration will lead to waste of time searching over inferior regions of the search space and slow down the convergence rate. On the other hand, emphasizing on exploitation will cause loss of diversity early in the search process, thereby possibly getting stuck into a local optimum. Therefore, in the population-based evolutionary algorithms, it is important to obtain the balance between exploration and exploitation of the search space [1,2].

Among population-based algorithms, PSO is easy to implement and has performed well on many optimization problems. PSO is also known for having the ability to quickly converge to the optimal [5]. However, in PSO, all particles share its swarm's best experience (the global best) that can lead the particles to cluster around the global best. In case, if the global best is located near a local minimum, escaping from local optimum becomes difficult and PSO suffers diversity loss near the local minimum [5]. In order to balance the exploration behavior of global search and exploitation nature of local search in PSO, inertia weight w was firstly proposed by Shi and Eberhart [4]. Clerc and Kennedy also developed another control parameter called constriction coefficient χ to control the convergence tendency of the particle swarm, including exploration and exploitation abilities [6]. In [7], self-organizing hierarchical PSO (HPSO-TVAC) was introduced with time varying acceleration coefficients. With decreasing cognitive component and increasing social component, global exploration is enhanced to avoid premature convergence in the early stages and local exploitation is enhanced to converge to the global optimum solution during the latter stages of the search.

The neighborhood topologies also control PSO's exploration and exploitation abilities according to the information sharing among the particles in the swarm [8–10]. Based on the findings in

* Corresponding author at: School of Electrical and Electronics Engineering Nanyang Technological University, Singapore.
E-mail addresses: nandar001@ntu.edu.sg (N. Lynn), epnsugan@ntu.edu.sg (P.N. Suganthan).

http://dx.doi.org/10.1016/j.swevo.2015.05.002

[8,10], a fully-informed PSO (FIPS) was proposed in which information from fully connected neighborhood was used [11] and different neighborhood topologies were studied in [12]. In [13], best experiences from local neighborhood and global neighborhood were used in unified particle swarm optimization (UPSO) algorithm by combining their exploration and exploitation abilities. The paper mentioned that the neighborhood size should be selected properly to get trade-off between exploration and exploitation.

Instead of using neighborhood topology to learn the information from other particles, Liang proposed comprehensive learning particle swarm optimizer (CLPSO) in which each particle learnt from other particles' best experiences for different dimensions via a comprehensive learning strategy [14]. In CLPSO, learning probability curve is set so that the particles have different levels of exploration and exploitation abilities. In orthogonal learning particle swarm optimization (OLPSO), orthogonal learning strategy was developed in which a particle learnt from the combination of its own best experience (cognitive learning) and its neighborhood best experience (social learning) to compromise the balance between the exploration and exploitation search [15]. Efficient population utilization strategy for particle swarm optimization (EPUS-PSO) was presented in [16]. In EPUS-PSO, solution sharing and search range sharing strategies are proposed to share best information among the particles and to avoid the particles from getting trapped in a local optimum. Population size is varied by a population manager according to the status of the solution search [16].

The new learning strategy called scatter learning strategy was presented in a scattering learning particle swarm optimization algorithm (SLPSOA) [17]. In scatter learning strategy, the exemplar pool (EP) is constructed which is composed of a certain number of relatively high-quality solutions scattered in the solution search space and enables the particles to explore different regions. Then, the particles select their exemplars from EP using roulette wheel rule and the selected exemplar is used for a certain number of iterations to exploit the corresponding region thoroughly. A competitive swarm optimizer (CSO) was developed where neither personal best position nor global best position was involved in updating the particles' positions [18]. In CSO, the two particles are randomly selected to compete and the loser will update its position by learning from the winner and mean position of current swarm. The empirical analysis of exploration and exploitation abilities showed that CSO achieved a good balance between exploration and exploitation [18].

In [19], as a new approach to balance the exploration and exploitation in PSO, predator prey optimizer was developed, combining PSO idea with predator prey strategy. In predator prey optimizer (PPO), one predator particle is introduced to attract the best particle in the swarm while prey particles are repelled from the predator and the best. The balance between the two processes is influenced and controlled by the interactions of the predator and prey particles. In [20], attractive and repulsive PSO (ARPSO) was introduced with negative entropy into original PSO, encouraging high diversity and discouraging premature convergence in order to obtain trade-off between the two. Blackwell and Bentley also introduced the repulsive force to PSO and proposed the atomic swarm that is composed of equal number of charged and neutral particles so that there is a balance between exploration and exploitation [21].

In order to address the exploration and exploitation trade-off problem, heterogeneous particle swarm optimization was proposed in [22,23]. In heterogeneous PSO (HPSO) [23], the particles in heterogeneous swarms were allowed to follow different velocity and position updating rules from a behavior pool, thereby having the ability to explore and exploit throughout the problem search space. In [24], a multi-swarm PSO using charged particles (PSO-2S) was developed in which the search space was partitioned and two kinds of swarms were used, called main and auxiliary. In PSO-2S, the auxiliary swarms are initialized in different partitioned areas, using charged particles. After certain number of generations, main swarm is formed with the best individual of the auxiliary swarms to search for the optimum. In [25], a cooperative approach was applied to PSO (CPSO-S_k) in which the dimensionality of the search space was split and different swarms were used to search over different dimensions of a solution cooperatively. Multi-swarm idea have also been used for locating multiple optimal solution in [26] and for dynamic environments in [21,24,27].

From the PSO variants mentioned above, it is obvious that the main issue in PSO is to keep the balance between the exploration and exploitation and researchers addressed this issue by suggesting different methods. Inspired by those methods, a CLPSO with two subpopulation groups, called Heterogeneous CLPSO (HCLPSO), is proposed in this paper. Instead of relying on one method to balance the exploration and exploitation ability of PSO, this paper addresses the issue by the following methods: by using adaptive control parameters, by controlling the information sharing (or topology) among the particles, by using a learning strategy and by using heterogeneous swarm rather than homogeneous.

In this paper, a heterogeneous swarm is used where the swarm is divided into two subpopulations. Each subpopulation is assigned to carry out the exploration and exploitation search separately. Exploration and exploitation processes are enhanced without one process crippling the other. Comprehensive learning strategy is used to generate an exemplar for the particles to learn. In PSO, learning from the two exemplars, personal best and the whole swarm's best, can cause two problems. One is "oscillation phenomenon" [28] which can occur if the two experiences were in opposite directions. This makes the search ability inefficient and slows down the convergence speed of the algorithm. Another is "two steps forward, one step back phenomenon" [25] which causes the solution vector to be improved on some dimensions and to be declined on other dimensions as one exemplar may have good values on some dimensions and others may have good values on some other dimensions. Thus, in order to extract such useful information from different dimensions of different particles in the swarm, comprehensive learning (CL) strategy is used to generate a promising exemplar in the proposed algorithm.

Via comprehensive learning (CL) strategy, the exploration-subpopulation group learns for different dimensions from its own members' previous best experiences and its particles have high level of exploration ability. The exploitation-subpopulation benefits by learning from the best experiences of all particles in the swarm including the whole swarm's best experience and therefore, its particles have strong exploitation ability. Different learning probability values are specified for each particle in the swarm such that the particles from the exploration-subpopulation are not influenced by the exploitation-subpopulation. In this way, the information sharing among the particle is controlled and at the same time, the exploitation-subpopulation is able to exploit instantly new good regions discovered by the exploration-subpopulation. Besides, adaptive control parameters are used in the subpopulation groups to enhance exploration and exploitation. Therefore, this novel heterogeneous subpopulation structure is able to emphasize exploration and exploitation simultaneously without one process unfavorably influencing the other.

This paper is organized as follows: original PSO is introduced in Section 2 and the proposed HCLPSO is presented in Section 3. In Section 4, the performance of the proposed HCLPSO algorithm is evaluated using the benchmark problems and compared with other state-of-art PSO algorithms. The research limitation and future works are also discussed in Section 4. Finally, the paper is concluded in Section 5.

2. Particle swarm optimization

Particle swarm optimization (PSO) is a population-based optimization technique introduced by Eberhart and Kennedy [3]. PSO was inspired by swarm behavior such as bird flocking and fish schooling. Without colliding with each other, a flock of birds or a school of fish is able to search for the food or shelter. Members share information within the group. Each member updates its direction by using their own findings and group's information. Imitating this social behavior, the PSO algorithm was developed in [3]. In PSO algorithm, each particle in the swarm represents a potential solution to a given problem. The particles navigate by adjusting their flying directions using their own and other swarm members' best experiences to find the optimum of the problem. This phenomenon was formulated in [3] as follows:

$$V_i^d = V_i^d + c_1 * rand1_i^d * \left(pbest_i^d - X_i^d\right) + c_2 * rand2_i^d * (gbest^d - X_i^d). \quad (1)$$

$$X_i^d = X_i^d + V_i^d \quad (2)$$

where i represents each particle in the population ($i=1, 2, \ldots, N$) and d represents a dimension ($d=1, 2, \ldots, D$). X_i^d and V_i^d are the position and velocity components of ith particle in the population, respectively. $pbest_i^d$ is the best position of ith particle and $gbest^d$ is the best position found by the whole swarm population. c_1 and c_2 are acceleration coefficients. $rand1_i^d$ and $rand2_i^d$ are randomly generated numbers in the range of [0, 1]. In order to perform search in a controlled manner, the updated velocity V_i^d is limited with a maximum magnitude V_{max}. If $|V_i^d|$ exceeds V_{max}, then $V_i^d = sign(V_i^d) * V_{max}$ for such dimensions. To achieve a balance between global exploration and local exploitation, a new parameter 'w' called inertia weight was introduced into PSO and used to control the flying velocity as follows [4]:

$$V_i^d = w * V_i^d + c_1 * rand1_i^d * \left(pbest_i^d - X_i^d\right) + c_2 * rand2_i^d * (gbest^d - X_i^d). \quad (3)$$

In [4,5], inertia weight w was defined as a linearly decreasing function of the run time. In addition, as mentioned in Section 1, Clerc and Kennedy also developed another velocity update using χ called the constriction coefficient to control dynamic characteristics of the particle swarm, including its exploration and exploitation tendencies as follows [5]:

$$V_i^d = \chi \left[V_i^d + c_1 * rand1_i^d * \left(pbest_i^d - X_i^d\right) + c_2 * rand2_i^d * \left(gbest^d - X_i^d\right)\right] \quad (4)$$

where constriction coefficient χ is calculated as

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad \text{with } \varphi > 4. \quad (5)$$

$$\varphi = c_1 + c_2 \quad (6)$$

The constriction χ is set as $\chi=0.729$ with $c_1=c_2=2.05$ and $\varphi=4.1$ [6]. The inertia weight and construction coefficients are mathematically equivalent when (5) and (6) are intercepted [29]. In this paper, PSO with inertia weight is also used in balancing exploration and exploitation processes of the proposed PSO algorithm.

3. Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation

In this paper, to curtail the adverse influence of exploration and exploitation on each other, the swarm population is divided into two subpopulations with one each for exploration and exploitation, respectively. In order to determine the exemplar for the particles in each subpopulation, comprehensive learning (CL) strategy [14] is chosen among simple single population PSO algorithms [5,7,11,13]. In CLPSO algorithm, instead of following the best individual alone, every particle in the swarm is able to learn from all other particles' best experiences for different dimensions. Besides, exploration and exploitation level of the particles can be specified through learning probability curve. Therefore, CL strategy is selected in the proposed algorithm to generate the exemplar for the particles in both exploration and exploitation subpopulation groups. The comprehensive learning strategy is briefly introduced in Section 3.1. The proposed algorithm is named as Heterogeneous CLPSO (HCLPSO) and is presented in Section 3.2.

3.1. Comprehensive learning particle swarm optimizer

The flying direction of a particle is guided by its own $pbest_i^d$ and $gbest^d$ in the original PSO. However, *gbest* may be far from the global optimum and may represent an inferior local optimum of a multimodal problem. To resolve this situation, a comprehensive learning strategy was proposed. In CLPSO, a particle's velocity is updated using all the particles' *pbests*. Each dimension of a particle learns from different particles' *pbests* instead of learning from the same exemplar for all dimensions. This enhances the diversity of the population. The velocity of ith particle is updated with the following equation as in [14]:

$$V_i^d = wV_i^d + c * rand_i^d * \left(pbest_{fi(d)}^d - X_i^d\right) \quad (7)$$

where $f_i(d) = [f_i(1), f_i(2), \ldots, f_i(D)]$ indicates if the ith particle follows its own or other's $pbest_i^d$ for each dimension d. The exemplar for each dimension is decided according to learning probability *Pc* values (different *Pc* values for different particles, Pc_i vs. *particle id*) and explorative particles and exploitative particles are specified according to these learning probability values. The *Pc* value for each particle is calculated with the following equation [14]:

$$Pc_i = a + b * \frac{(\exp((10(i-1))/(ps-1)) - 1)}{(\exp(10) - 1)} \quad (8)$$

where '*ps*' represents the population size, $a=0.05$, $b=0.45$. To choose either its own or other's $pbest_i^d$ for each corresponding dimension of the ith particle, a random number is generated for each dimension and compared with its learning probability Pc_i value. If random number is smaller than Pc_i value, the ith particle is guided by other particle's $pbest_i^d$ position which is determined by tournament selection of size 2, i.e. two particles are selected randomly and the particle with better fitness is chosen for the corresponding dimension. If random number is larger than Pc_i, the particle will follow its own *pbest* position for that dimension. Therefore, the exemplar $pbest_{fi(d)}$ is a new position where each dimension learns from several particles' *pbest* positions. In order to ensure particle's movement improves its *pbest*, a certain number of evaluations is defined as refreshing gap m in CLPSO and a new $pbest_{fi(d)}$ will be generated if there is no improvement for m (the refreshing gap) consecutive moves. The search range is also restricted in CLPSO with the bound $[X_{min}, X_{max}]$. If the updated position of the particle is out of the bound, its fitness value and its *pbest* are not updated.

3.2. Heterogeneous CLPSO with enhanced exploration and exploitation

Exploration emphasizes on finding various potential solution regions of the entire search space and exploitation focuses on refining the promising solutions in the potential solution regions to attain the optimal solution. With the exploration and exploitation ability, the particles can fly throughout the search space to find the global optimum. In CLPSO, each dimension of the particle learns from either its own best position or other particles' best position. The exemplar selection was decided by comparing the random number with the learning probability *Pc* curve. With different *Pc* values, the particles have different level of exploration and exploitation ability. However, particles with high exploration tendency can be adversely influenced by particles with high exploitation tendency. Therefore, in order to address this issue and balance the exploration and exploitation search, CLPSO is enhanced with exploration-subpopulation and exploitation-subpopulation group and the algorithm named Heterogeneous CLPSO (HCLPSO) is proposed in this paper.

In HCLPSO, the swarm is divided into two heterogeneous subpopulations. The first subpopulation is enhanced for exploration and the second subpopulation is enhanced for exploitation. In both exploration and exploitation subpopulations, the exemplar is generated using comprehensive learning (CL) strategy with the learning probability *Pc* curve as shown in Fig. 1 below. The learning probability values for each particle in the swarm are calculated using Eq. (8) with $a=0$, $b=0.25$. The velocity of the exploration-enhanced subpopulation is updated using Eq. (7), while the velocity of exploitation-enhanced subpopulation is updated using the following equation:

$$V_i^d = w*V_i^d + c_1*rand1_i^d*\left(pbest_{fi(d)}^d - X_i^d\right) + c_2*rand2_i^d*\left(gbest^d - X_i^d\right) \tag{9}$$

The inertia weight w [4] is used as a linearly decreasing function of run time in the range of 0.99–0.2. In subpopulation group 1, time varying acceleration coefficient $c=3$–1.5 is used in Eq. (7) to enhance the exploration and in subpopulation group 2, the time varying acceleration coefficients $c_1=2.5$–0.5 and $c_2=0.5$–2.5 [7] are used in Eq. (9) to enhance exploitation.

As shown in Eqs. (7) and (9), all the particles in both subpopulations are guided by the exemplars obtained from comprehensive learning (CL) strategy. As described in Section 3.1, the random number is generated for each dimension of a particle and compared with its respective learning probability Pc_i value. If random number is smaller than Pc_i value, the particle will learn from another particle's *pbest*. The exemplar is determined with the tournament selection procedure in which the two particles are randomly selected from the subpopulation group 1 and the corresponding dimension will learn from the particle with better fitness. In the case of the random number is larger than Pc_i value, the corresponding dimension will learn from its own *pbest*. According to the learning probability *Pc* curve illustrated in Fig. 1, the subpopulation group 1 has low learning probability values close to zero. If random number generated for each dimension of a particle were compared with low learning probability values, the particle in the subpopulation group 1 will learn mostly from its own *pbest* for most of the dimensions. If all dimensions of a particle were its own *pbest*, we will randomly choose one dimension to learn from another particle's *pbest*'s corresponding dimension from group 1. Besides, time varying acceleration coefficient *c*, started at 3 and linearly reduced to 1.5, is also used to enhance the exploration ability of the particles. Therefore, the particles are explorative and the subpopulation group 1 has strong exploration ability.

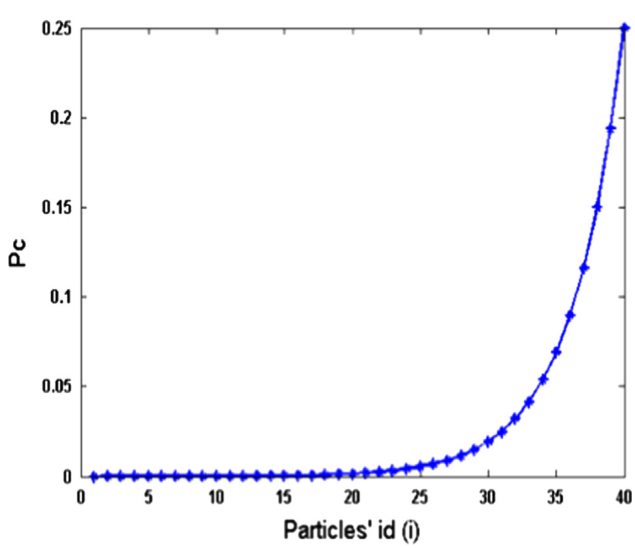

Fig. 1. Learning probability *Pc* curve calculated by using Eq. (8) with $ps=40$, $a=0$ and $b=0.25$.

Unlike the subpopulation group 1, the particles from the subpopulation group 2 learn not only from the exemplar generated by using CL strategy, but also from the swarm's best experience *gbest* as described in Eq. (9). Thus, the subpopulation group 2 has high exploitation ability. If the particles are learning from their own *pbests* and *gbest*, it has higher chance of getting stuck into local optima. To avoid such condition, a particle will adopt other particles' *pbests* (up to 25%) as shown in Fig. 1. However, as 25% probability is still low, one dimension will be adopted from another particle if a particle happens to learn from its own particle for all dimensions. For the acceleration coefficients from Eq. (9), c1 is used in the range of 2.5–0.5 to preserve the diversity satisfactorily in the early search stages. The acceleration coefficients c_2 is used in the range of 0.5–2.5 to emphasize on increasing exploitation of the entire swarm's best experience. Thus, the particles are exploitative and the subpopulation group 2 has strong exploitation ability.

Therefore, the swarm is composed of explorative particles and exploitative particles and the exploration and exploitation processes are performed by the first subpopulation and the second subpopulation, respectively. Since the explorative particles are not allowed to access the information of the exploitative particles, there is no information flow from the exploitation subpopulation group to the exploration subpopulation group. Thus, rapid information flow is avoided and even if the exploitation group suffers premature convergence, the exploration group has the potential to rescue the exploitation oriented group from the local optimum. Therefore, a compromise between exploration and exploitation is achieved in the proposed HCLPSO algorithm. As in CLPSO, if there is no improvement for refreshing gap m (*number of iterations*), a new $pbest_{fi(d)}$ will be generated by learning from its own population itself for the subpopulation group 1 and by learning from the whole population for the subpopulation group 2.

Swarm diversity can be used to identify if populations are conducting exploration or exploitation [30]. Thus, in this paper, diversity of each subpopulation group and whole population is studied for one unimodal function (shifted sphere function) and one multimodal function (shifted rotated Griewank's function) on 30 dimensions. The diversity measure of the swarm is taken according to Eqs. (10) and (11) [30,31]. The number of function evaluation of 300,000 and population size of 40 are used in the diversity study. The diversity graphs from a single test run are shown in Fig. 2. It can be clearly seen in diversity graphs that diversity of the exploration-subpopulation group is relatively higher than that of exploitation-subpopulation group and the diversity is preserved even if exploitation-subpopulation group converges to the acceptable solution. The performance of proposed PSO algorithm is evaluated by comparing with other state-of-art PSO algorithms and the experimental results are discussed in the next sections.

$$Diversity\ (S(t)) = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\sum_{d=1}^{D}\left(X_i^d(t) - \overline{X^d(t)}\right)^2} \tag{10}$$

$$\overline{X^d(t)} = \frac{\sum_{i=1}^{N} X_i^d(t)}{N} \tag{11}$$

where S is the swarm and $N=|S|$ is the swarm size. D is the dimensionality of the problem. X_i^d is the dth value of ith particle in

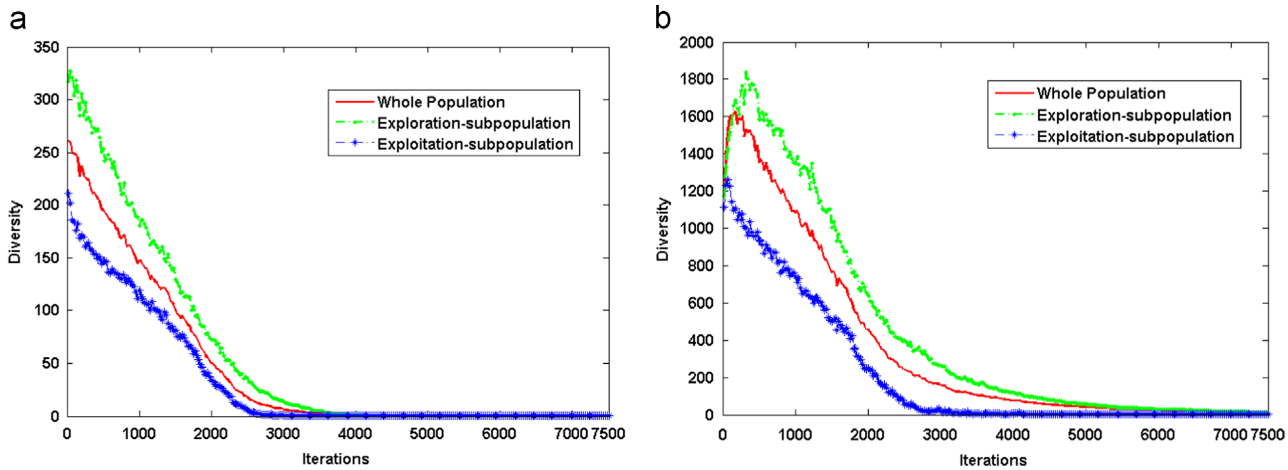

Fig. 2. Diversity comparisons between exploration-subpopulation, exploitation subpopulation and the whole swarm on unimodal shifted sphere function in (a) and on multimodal shifted rotated Griewank's function in (b).

Table 1
Cec 2005 test functions.

Function type	Function name	Initialization range	Search range	Global optimum (x^*)	$F(x^*)$ f_bias
Unimodal functions	F_1: Shifted sphere function	$[-100,100]^D$	$[-100,100]^D$	$\boldsymbol{o}$	−450
	F_2: Shifted Schwefel's problem 1.2	$[-100,100]^D$	$[-100,100]^D$	$\boldsymbol{o}$	−450
	F_3: Shifted rotated high conditioned elliptic function	$[-100,100]^D$	$[-100,100]^D$	$\boldsymbol{o}$	−450
	F_4: Shifted Schwefel's problem 1.2 with noise in fitness	$[-100,100]^D$	$[-100,100]^D$	$\boldsymbol{o}$	−450
	F_5: Schwefel's problem 2.6 with global optimum on bounds	$[-100,100]^D$	$[-100,100]^D$	$\boldsymbol{o}$	−310
Multimodal functions	F_6: Shifted Rosenbrock's function	$[-100,100]^D$	$[-100,100]^D$	$\boldsymbol{o}$	390
	F_7: Shifted rotated Griewank's function without bounds	$[0,600]^D$	$[-600,600]^D$	$\boldsymbol{o}$	−180
	F_8: Shifted rotated Ackley's function with global optimum on bounds	$[-32,32]^D$	$[-32,32]^D$	$\boldsymbol{o}$	−140
	F_9: Shifted Rastrigin's function	$[-5,5]^D$	$[-5,5]^D$	$\boldsymbol{o}$	−330
	F_{10}: Shifted rotated Rastrigin's function	$[-5,5]^D$	$[-5,5]^D$	$\boldsymbol{o}$	−330
	F_{11}: Shifted rotated Weierstrass function	$[-0.5, 0.5]^D$	$[-0.5, 0.5]^D$	$\boldsymbol{o}$	90
	F_{12}: Schwefel's problem 2.13	$[-100,100]^D$	$[-100,100]^D$	$\boldsymbol{o}$	−460
Expanded functions	F_{13}: Expanded extended Griewank's plus Rosenbrock's function(F8F2)	$[-3,1]^D$	$[-3,1]^D$	$\boldsymbol{o}$	−130
	F_{14}: Shifted rotated expanded Scaffer's F6	$[-100,100]^D$	$[-100,100]^D$	$\boldsymbol{o}$	−300
Hybrid composition functions	F_{15}: Hybrid composition function	$[-5,5]^D$	$[-5,5]^D$	$\boldsymbol{o}$	120
	F_{16}: Rotated hybrid composition function	$[-5,5]^D$	$[-5,5]^D$	$\boldsymbol{o}$	120
	F_{17}: Rotated hybrid composition function with noise in fitness	$[-5,5]^D$	$[-5,5]^D$	$\boldsymbol{o}$	120
	F_{18}: Rotated hybrid composition function	$[-5,5]^D$	$[-5,5]^D$	$\boldsymbol{o}$	10
	F_{19}: Rotated hybrid composition function with a narrow basin for the global optimum	$[-5,5]^D$	$[-5,5]^D$	$\boldsymbol{o}$	10
	F_{20}: Rotated hybrid composition function with the global optimum on the bounds	$[-5,5]^D$	$[-5,5]^D$	$\boldsymbol{o}$	10
	F_{21}: Rotated hybrid composition function	$[-5,5]^D$	$[-5,5]^D$	$\boldsymbol{o}$	360
	F_{22}: Rotated hybrid composition function with high condition number matrix	$[-5,5]^D$	$[-5,5]^D$	$\boldsymbol{o}$	360
	F_{23}: Non-continuous rotated hybrid composition function	$[-5,5]^D$	$[-5,5]^D$	$\boldsymbol{o}$	360
	F_{24}: Rotated hybrid composition function	$[-5,5]^D$	$[-5,5]^D$	$\boldsymbol{o}$	260
	F_{25}: Rotated hybrid composition function without bounds	$[-2,5]^D$	$[-5,5]^D$	$\boldsymbol{o}$	260

* $\boldsymbol{o}=[o_1, o_2, o_3, . . , o_D]$: shifted global optimum.

the population and $\overline{X^d(t)}$ is the average of dth dimension over all the particles in the swarm.

4. Performance evaluation

4.1. Test functions and compared PSOs

In this paper, shifted and rotated CEC 2005 benchmark functions are used to evaluate the performance of the proposed HCLPSO algorithm. The set of CEC 2005 benchmark functions is composed of all types of unimodal, multimodal, expanded and hybrid composite functions [32]. All 25 CEC 2005 benchmark functions are listed with their global optimum, search range, initialization range, bias values and shown in Table 1. The acceptable tolerance of each function is also defined in Table 1. If the result is obtained within the acceptable tolerance of the global optimum, it is defined as the run is successful. The performance of the proposed algorithm is evaluated with other state-of-art PSO algorithms such as

- global version of PSO (PSO) [5];
- fully-informed PSO (FIPS) [11];
- unified PSO (UPSO) [13];
- comprehensive learning PSO (CLPSO) [14];
- self-organizing hierarchical PSO with time varying acceleration coefficients (HPSO-TVAC) [7];
- orthogonal learning PSO (OLPSO) [15]; and
- static heterogeneous swarm optimization (sHPSO) [23].

Table 2
Parameter settings.

Algorithm	Inertia weight w	Constriction coefficients χ	Acceleration coefficients c_1, c_2, c	Population size	Function evaluations (FES)	Reference
PSO	0.9–0.4	–	$c_1=2$, $c_2=2$	40	300,000	[4]
FIPS	–	0.729	$c=2$	40	300,000	[11]
UPSO	–	0.729	$c=1.49445$	40	300,000	[13]
CLPSO	0.9–0.4	–	$c=1.49445$	40	300,000	[14]
OLPSO	0.9–0.4	–	$c=2$	40	300,000	[15]
HPSO-TVAC	–	–	$c_1=2.5–0.5$, $c_2=0.5–2.5$	40	300,000	[7]
sHPSO	0.72	–	$c_1=2.5–0.5$, $c_2=0.5–2.5$	40	300,000	[23]
HCLPSO	0.99–0.2	–	$c_1=2.5–0.5$, $c_2=0.5–2.5$, $c=3–1.5$	40	300,000	

Table 3
Calibration of two subpopulation group sizes (g_1 and g_2) for the proposed HCLPSO algorithm.

Functions	Criteria	5+35	10+30	15+25	20+20	25+15	30+10	35+5
F_1	Mean	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	Std	0	0	0	0	0	0	0
	Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**
F_2	Mean	**8.69E–10**	2.09E–08	1.70E–06	2.34E–04	1.82E–02	0.35	3.61
	Std	1.35E–09	3.97E–08	1.71E–06	3.67E–04	1.48E–02	0.41	4.77
	Rank	**1**	2	3	4	5	6	7
F_3	Mean	**5.37E+05**	5.89E+05	6.42E+05	8.78E+05	1.41E+06	2.39E+06	3.09E+06
	Std	1.73E+05	1.86E+05	2.61E+05	3.56E+05	6.14E+05	9.61E+05	1.39E+06
	Rank	**1**	2	3	4	5	6	7
F_4	Mean	**2.20E+02**	4.52E+02	5.22E+02	6.35E+02	8.24E+02	1.57E+03	2.43E+03
	Std	1.21E+02	2.60E+02	3.09E+02	4.82E+02	3.86E+02	7.59E+02	9.70E+02
	Rank	**1**	2	3	4	5	6	7
F_5	Mean	**2.96E+03**	3.04E+03	2.97E+03	3.14E+03	3.38E+03	3.39E+03	3.49E+03
	Std	5.81E+02	6.08E+02	4.55E+02	4.20E+02	6.49E+02	4.87E+02	4.80E+02
	Rank	**1**	3	2	4	5	6	7
F_6	Mean	<u>10.28</u>	3.22	2.39	2.33	0.72	**0.31**	0.43
	Std	15.29	3.53	4.27	3.76	0.97	0.58	0.64
	Rank	7	6	5	4	3	**1**	2
F_7	Mean	**0.01**	0.02	0.02	0.02	0.02	0.02	0.02
	Std	0.01	0.02	0.02	0.02	0.02	0.02	0.02
	Rank	**1**	2	2	2	2	2	2
F_8	Mean	**20.82**	20.86	20.87	20.91	20.91	20.91	20.93
	Std	0.09	0.07	0.09	0.06	0.06	0.09	0.05
	Rank	**1**	2	2	4	5	6	7
F_9	Mean	<u>1.19</u>	<u>0.10</u>	**0**	**0**	**0**	**0**	**0**
	Std	1.12	4.01E–01	0	0	0	0	0
	Rank	3	2	**1**	**1**	**1**	**1**	**1**
F_{10}	Mean	59.33	60.84	56.08	**55.34**	58.35	68.88	74.53
	Std	16.96	14.08	12.90	12.13	12.73	17.13	15.88
	Rank	4	5	2	**1**	3	6	7
F_{11}	Mean	**19.47**	20.15	20.32	20.04	20.90	21.29	22.77
	Std	4.16	3.86	2.94	2.86	2.54	2.15	2.00
	Rank	**1**	3	5	2	7	4	6
F_{12}	Mean	3.87E+03	**3.20E+03**	3.91E+03	3.58E+03	4.92E+03	6.74E+03	9.75E+03
	Std	3.24E+03	3.43E+03	3.69E+03	3.50E+03	3.75E+03	3.42E+03	4.75E+03
	Rank	3	**1**	3	2	4	5	6
F_{13}	Mean	1.74	1.54	**1.45**	1.65	1.72	2.00	2.14
	Std	0.28	0.28	0.28	0.32	0.35	0.37	0.26
	Rank	5	2	**1**	3	4	6	7
F_{14}	Mean	**11.58**	11.86	11.93	11.95	12.01	12.16	12.29
	Std	0.57	0.46	0.58	0.45	0.54	0.44	0.44
	Rank	**1**	2	3	4	5	6	7
Average rank		2.21	2.50	2.57	2.86	3.93	4.43	5.29
Final rank		**1**	2	3	4	5	6	7

The first algorithm, global version of PSO, used inertia weight to balance the exploration and exploitation ability in finding the global optimum. In FIPS algorithm, information of all neighbors was used to guide a particle and various topological structures of the population are tested to control the exploration and exploration process. UPSO used both the swarm's best experience and the

Table 4
Calibration of acceleration coefficient c of exploration-subpopualtion group with $g_1=15$, $g_2=25$.

Functions	Criteria	$c_1=2.5$–0.5, $c_2=0.5$–2.5, $c=1.5$	$c_1=2.5$–0.5, $c_2=0.5$–2.5, $c=3$–1.5	$c_1=2.5$–0.5, $c_2=0.5$–2.5, $c=3$
F_1	Mean	**0**	**0**	**0**
	Std	0	0	0
	Rank	**1**	**1**	**1**
F_2	Mean	8.60E–06	1.70E–06	**2.27E–08**
	Std	1.36E–05	1.71E–06	3.89E–08
	Rank	3	2	**1**
F_3	Mean	8.77E+05	6.42E+05	**5.42E+05**
	Std	4.85E+05	2.61E+05	2.41E+05
	Rank	3	2	**1**
F_4	Mean	4.57E+02	5.22E+02	**2.46E+02**
	Std	2.68E+02	3.09E+02	1.30E+02
	Rank	2	3	**1**
F_5	Mean	3.11E+03	2.97E+03	**2.96E+03**
	Std	4.97E+02	4.55E+02	5.49E+02
	Rank	3	2	**1**
F_6	Mean	2.86	**2.39**	10.72
	Std	6.54	4.27	14.62
	Rank	2	**1**	3
F_7	Mean	**0.02**	0.02	0.02
	Std	0.01	0.02	0.02
	Rank	**1**	2	2
F_8	Mean	**20.87**	20.87	20.87
	Std	0.08	0.09	0.09
	Rank	**1**	2	2
F_9	Mean	0.10	**0**	0.07
	Std	0.40	**0**	2.52E–01
	Rank	3	**1**	2
F_{10}	Mean	66.37	**56.08**	58.10
	Std	24.35	12.90	17.18
	Rank	3	**1**	2
F_{11}	Mean	22.15	20.32	**19.96**
	Std	2.34	2.94	3.19
	Rank	3	2	**1**
F_{12}	Mean	5.65E+03	**3.91E+03**	4.43E+03
	Std	6.31E+03	3.69E+03	3.94E+03
	Rank	2	**1**	1
F_{13}	Mean	1.59	**1.45**	1.60
	Std	0.24	0.28	0.31
	Rank	2	**1**	3
F_{14}	Mean	11.97	11.93	**11.78**
	Std	0.60	0.58	0.55
	Rank	3	2	**1**
Average rank		2.29	1.64	1.57
Final rank		3	2	**1**

particle's neighborhood best experience to adjust the exploration and exploitation tendencies. CLPSO used learning probability curve and CL strategy that encouraged a particle to learn from different particles with different levels of exploration and exploitation ability. HPSO-TVAC introduced time varying acceleration coefficients to encourage the particles for stronger exploration in the early search and for stronger exploitation at the end of the search. In OLPSO, a particle is guided by an exemplar constructed from its personal best and the global best using an orthogonal learning strategy. In static heterogeneous particle swarm optimizer (sHPSO), different velocity updating rules of PSO [3], cognitive-only velocity update [33], social-only velocity update [33], barebones PSO [34], modified barebones PSO [34] were used. The behaviors are randomly assigned to the particles and kept unchanged throughout the search process in sHPSO algorithm. All the algorithms are tested on all 25 benchmark functions and run 30 times using the same number of function evaluations and population size. The detail parameter settings of the proposed HCLPSO and other algorithms are presented in Table 2 and refreshing gap m is set at five for all the experiments in this paper.

4.2. Parameter tuning

In the proposed HCLPSO algorithm, there are four main parameters to be adjusted for balancing exploration and exploitation between two subpopulation groups. The first parameter to tune is subpopulation group size and the next parameters to adjust are the acceleration coefficients, c, from exploration-subpopulation velocity update Eq. (7) and c1 and c2 from exploitation-subpopulation velocity update Eq. (9). The parameters are tuned using the first 14 CEC 2005 test functions with the same parameter settings of run times 30, population size 40, FES 300,000 and dimension 30. The results are ranked based on error mean and standard deviation values and summarized with final rank resulted from averaging over the ranks.

The experiment results of tuning subpopulation group sizes are shown in Table 3. According to the final rank, the subpopulation size of ($g_1=5$, $g_2=35$) obtained the best performance for all 14 benchmark functions compared to other subpopulation settings. However, as underlined in Table 3, this parameter setting provides unsatisfactory performance on test functions F_6 and F_9 as the second best parameter setting ($g_1=10$, $g_2=30$) does on function F_9. On the other hand, the parameter setting ($g_1=15$, $g_2=25$) provides the satisfactory and consistent performance throughout 14 problems.

For acceleration coefficient c from exploration-subpopulation velocity update Eq. (7), c is used as a constant value of 1.49445 in [14]. In this paper, c is calibrated using constant and time varying values and its calibration results are presented in Table 4. For c1 and c2 from exploitation-subpopulation velocity update Eq. (9), time varying values of c_1 started at 2.5, linearly reduced to 0.5 and c_2 started at 0.5, linearly increased to 2.5 from literature [7] are used. In order to balance the search between exploration-subpopulation and exploitation-subpopulation groups, different c values are calibrated by pairing with $c_1=2.5$–0.5, $c_2=0.5$–2.5 from [7]. In calibration Table 4, $c=3$ provides the best performance according to final rank value. However, it performs poorly on functions F_6 and F_9 as underlined. Among three different settings, the second best parameter setting $c=3$–1.5 provides superior performance consistently for all 14 test functions.

Therefore, the subpopulation sizes ($g_1=15$, $g_2=25$) and time varying acceleration coefficients ($c_1=0.5$–2.5, $c_2=2.5$–0.5 and $c=3$–1.5) are selected for further performance evaluation of the proposed HCLPSO algorithm.

4.3. Comparative study with PSO variants

In this section, the proposed HCLPSO algorithm is compared with other state-of-the-art PSO algorithms and its performance is evaluated using several criteria. Firstly, the performance of each algorithm is measured and ranked in terms of error mean and standard deviation values of their solutions. The final rank for each algorithm is defined according to their average rank values over 25 benchmark problems. Secondly, in order to examine the statistical difference between HCLPSO and other PSO algorithms, non-parametric Wilcoxon signed-rank test [35–38] is conducted between the results of proposed HCLPSO and the results of the other PSO variants with the significance level of 5%. In this single-problem analysis, pairwise comparison is performed over the results obtained over 30 simulation runs on 10 and 30 dimensional problems. The symbol (+) represents that HCLPSO performs significantly better than the compared algorithm, the symbol

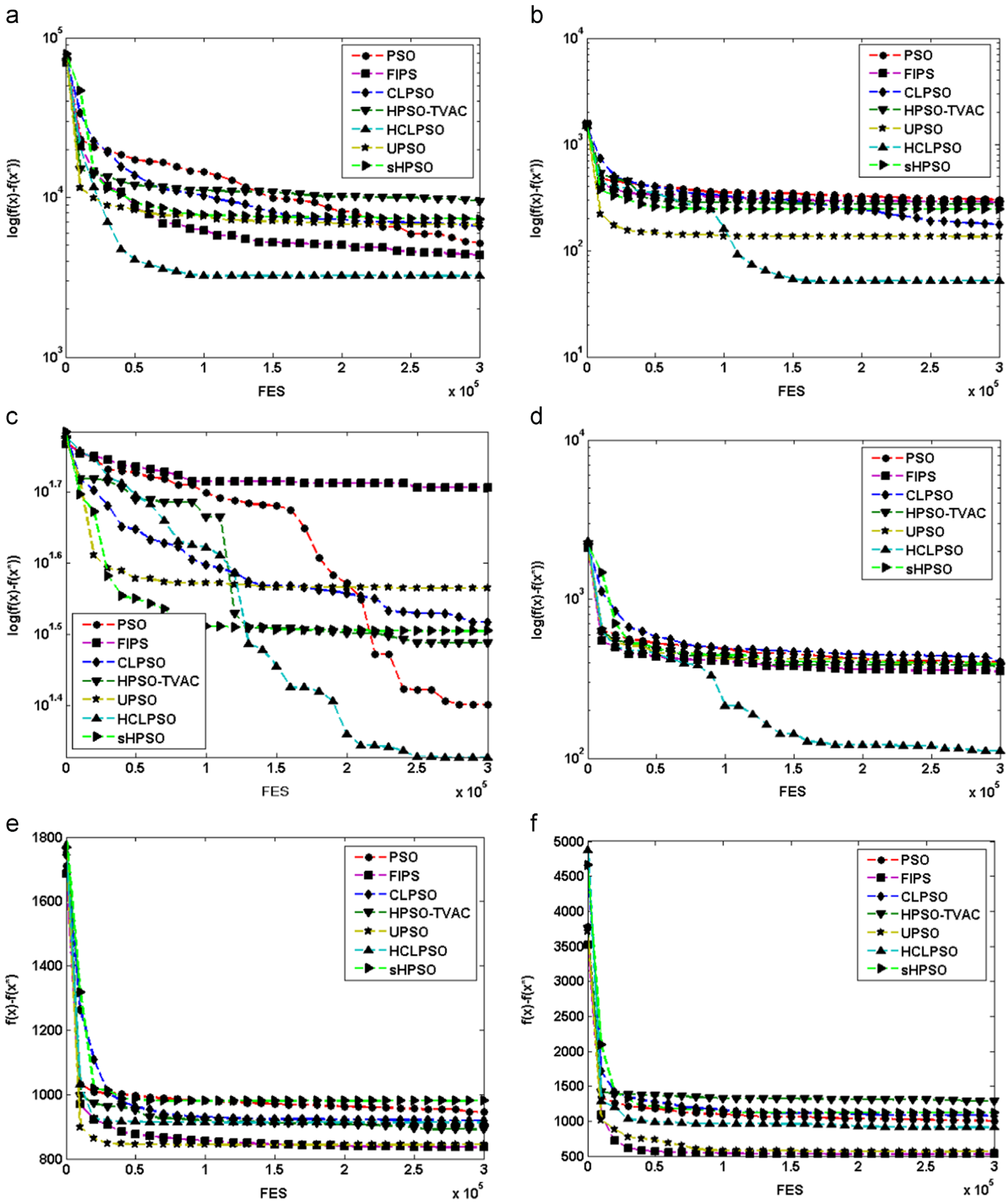

Fig. 3. Median convergence characteristics graphs of 30 dimensional CEC 2005 benchmark functions: (a) F_5: shwefel's problem 2.6, (b) F_{10}: shifted rotated Rastrigin's function, (c) F_{11}: shifted rotated Weierstrass function, (d) F_{17}: rotated hybrid composition function with noise in fitness, (e) F_{19}: rotated hybrid composition function with a narrow basin for the global optimum, (f) F_{22}: rotated hybrid composition function with high condition number matrix.

(0) represents that there is no significant difference between HCLPSO and the compared algorithm and the symbol (−) represents that the compared algorithm performs significantly better than HCLPSO algorithm. In addition, the convergence progress of all the algorithms is also analyzed for unimodal, multimodal and hybrid composite functions. As specified in [32], the median performance of 30 runs is used to analyze the convergence performance. Among 25 test cases, some convergence characteristics graphs are presented in Fig. 3.

Table 5
Comparison of experimental results of PSO algorithms for 10 dimensional CEC 2005 test functions with $g_1=8$ and $g_2=12$.

Functions	Criteria	HCLPSO	PSO	FIPS	UPSO	CLPSO	HPSO-TVAC	sHPSO
F_1	Mean	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	Std	0	0	0	0	0	0	0
	Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**
F_2	Mean	**0**	**0**	**0**	**0**	1.86E–02	**0**	**0**
	Std	0	0	0	0	2.24E–02	0	0
	Rank	**1**	**1**	**1**	**1**	2	**1**	**1**
F_3	Mean	6.83E+04	1.60E+05	2.24E+05	1.05E+05	3.49E+05	**6.36E+04**	8.82E+04
	Std	4.63E+04	1.10E+05	1.22E+05	1.16E+05	2.21E+05	3.81E+04	6.48E+04
	Rank	2	5	6	4	7	**1**	3
F_4	Mean	8.59E–03	**0**	**0**	1.47E+02	5.20	2.15E+02	2.31E+03
	Std	1.29E–02	**0**	**0**	3.38E+02	11.02	2.96E+02	2.23E+03
	Rank	2	**1**	**1**	4	3	5	6
F_5	Mean	7.51	1.04E+03	1.32E+02	6.03E+02	**6.95**	4.96E+02	1.27E+03
	Std	8.53	2.07E+02	1.11E+02	5.34E+02	16.01	3.61E+02	2.10E+03
	Rank	2	6	3	5	**1**	4	7
F_6	Mean	**0.81**	3.47	4.70	18.50	0.86	11.83	786.51
	Std	1.17	7.76	11.11	87.60	1.94	22.29	4017.41
	Rank	**1**	3	4	6	2	5	7
F_7	Mean	**0.11**	0.22	0.22	0.19	0.21	2.56	11.58
	Std	0.06	0.14	0.13	0.15	0.11	1.84	15.61
	Rank	**1**	5	4	2	3	6	7
F_8	Mean	20.21	20.29	20.36	20.36	20.27	20.17	**20.14**
	Std	0.07	0.09	0.08	0.06	0.05	0.11	0.12
	Rank	3	4	7	6	5	2	**1**
F_9	Mean	0.10	3.55	0.23	7.89	**0**	1.76	18.67
	Std	0.30	1.40	0.57	4.26	0	1.47	15.16
	Rank	2	5	3	6	**1**	4	7
F_{10}	Mean	9.43	14.36	14.27	15.95	**9.24**	34.76	34.22
	Std	3.04	5.21	6.40	5.75	2.97	11.37	12.14
	Rank	2	4	3	5	**1**	7	6
F_{11}	Mean	**3.29**	3.36	4.55	5.98	4.53	5.91	6.92
	Std	1.27	1.42	1.61	0.94	0.78	1.19	1.69
	Rank	**1**	2	4	7	3	5	6
F_{12}	Mean	**1.38E+01**	1.83E+03	2.61E+02	7.78E+01	7.22E+01	5.70E+02	1.43E+03
	Std	2.62E+01	4.31E+03	3.62E+02	2.71E+02	5.20E+01	7.52E+02	3.21E+03
	Rank	**1**	7	4	3	2	5	6
F_{13}	Mean	0.29	0.63	1.17	0.81	**0.28**	0.66	1.61
	Std	0.06	0.21	0.25	0.23	0.06	0.22	1.00
	Rank	2	3	6	5	**1**	4	7
F_{14}	Mean	**2.70**	2.86	2.93	3.30	3.00	3.20	3.60
	Std	0.45	0.41	0.34	0.33	0.27	0.35	0.41
	Rank	**1**	2	3	6	4	5	7
F_{15}	Mean	**2.51E+00**	2.34E+02	2.08E+02	2.63E+02	4.81E+00	2.15E+02	3.82E+02
	Std	1.38E+01	1.76E+02	1.76E+02	1.07E+02	1.46E+01	2.04E+02	1.93E+02
	Rank	**1**	5	3	6	2	4	7
F_{16}	Mean	**1.08E+02**	1.29E+02	1.12E+02	1.35E+02	1.20E+02	1.91E+02	1.87E+02
	Std	1.63E+01	1.64E+01	9.91E+00	2.45E+01	9.92E+00	3.91E+01	4.58E+01
	Rank	**1**	4	2	5	3	7	6
F_{17}	Mean	**1.13E+02**	1.42E+02	1.24E+02	1.49E+02	1.25E+02	1.95E+02	2.05E+02
	Std	1.36E+01	6.81E+01	1.46E+01	1.86E+01	1.29E+01	4.47E+01	7.31E+01
	Rank	**1**	4	2	5	3	6	7
F_{18}	Mean	**6.80E+02**	7.10E+02	8.06E+02	8.97E+02	6.85E+02	1.01E+03	9.79E+02
	Std	2.16E+02	2.53E+02	1.34E+02	1.20E+02	1.87E+02	8.67E+01	7.28E+01
	Rank	**1**	3	4	5	2	7	6
F_{19}	Mean	6.76E+02	6.99E+02	7.59E+02	8.62E+02	**6.54E+02**	9.95E+02	9.89E+02
	Std	2.03E+02	2.45E+02	1.69E+02	1.31E+02	1.88E+02	9.26E+01	1.42E+02
	Rank	2	3	4	5	**1**	7	6
F_{20}	Mean	6.24E+02	**6.16E+02**	7.81E+02	8.89E+02	7.46E+02	9.86E+02	1.01E+03
	Std	2.38E+02	2.66E+02	1.35E+02	9.67E+01	1.62E+02	1.12E+02	8.42E+01
	Rank	2	**1**	3	5	4	6	7
F_{21}	Mean	**3.93E+02**	4.51E+02	7.65E+02	8.23E+02	4.52E+02	1.04E+03	1.09E+03
	Std	1.41E+02	1.93E+02	2.84E+02	3.22E+02	1.40E+02	2.83E+02	2.23E+02
	Rank	**1**	2	4	5	3	6	7
F_{22}	Mean	**7.24E+02**	7.64E+02	7.98E+02	8.12E+02	7.27E+02	8.86E+02	8.86E+02

Table 5 (*continued*)

Functions	Criteria	HCLPSO	PSO	FIPS	UPSO	CLPSO	HPSO-TVAC	sHPSO
	Std	1.15E+02	9.30E+01	6.13E+01	7.19E+01	1.42E+02	1.30E+02	1.06E+02
	Rank	**1**	3	4	5	2	7	6
F_{23}	Mean	**5.58E+02**	6.77E+02	8.57E+02	1.01E+03	5.58E+02	1.12E+03	1.17E+03
	Std	5.55E+01	1.43E+02	2.66E+02	2.38E+02	7.16E+01	2.63E+02	1.69E+02
	Rank	**1**	3	4	5	2	6	7
F_{24}	Mean	2.10E+02	2.80E+02	3.77E+02	4.20E+02	**2.00E+02**	8.23E+02	6.62E+02
	Std	5.48E+01	1.56E+02	3.90E+00	1.11E+02	1.63E−08	4.59E+02	4.56E+02
	Rank	2	3	4	5	**1**	7	6
F_{25}	Mean	**2.00E+02**	3.10E+02	3.77E+02	4.46E+02	2.00E+02	7.87E+02	7.38E+02
	Std	1.21E−12	1.75E+02	2.33E+00	1.95E+02	2.59E−09	4.37E+02	4.85E+02
	Rank	**1**	3	4	5	2	7	6
Average rank		**1.44**	3.32	3.52	4.68	2.44	5.00	5.72
Final rank		**1**	3	4	5	2	6	7
Best/2nd Best/Worst		**15/9/0**	4/3/1	3/2/1	2/1/1	7/7/2	3/1/7	3/0/12
Algorithms		HCLPSO	PSO	FIPS	UPSO	CLPSO	HPSO-TVAC	sHPSO

4.3.1. *Results for 10 dimensional problems*

The experiment of 10 dimensional CEC 2005 test function is performed using population size 20 and the results are illustrated in Table 5. The exploration-subpopulation group size $g_1=8$ and exploitation-subpopulation group size $g_2=12$ are employed in the experiment of 10 dimension problems. As shown in comparison Table 5, on 10 dimensional unimodal problems, all PSO algorithms perform equally on functions F_1 and F_2. HPSO-TVAC provides the best solution on function F_3 as PSO and FIPS does the best on function F_4. CLPSO yields the best solution on function F_5. The proposed HCLPSO algorithm performs the second best and ranked 2 on those three unimodal functions F_3, F_4 and F_5.

On multimodal, expanded and hybrid composite functions, HCLPSO algorithm outperforms other algorithms on function F_6, F_7, F_{11}, F_{12}, F_{14}–F_{18}, F_{21}–F_{23}, F_{25} i.e. 13 out of 20 test functions. CLPSO also obtains the best performance on function F_9, F_{10}, F_{13}, F_{19} and F_{24}, i.e. 5 out of 20 functions. sHPSO does the best on function F_8 as PSO offers the best solution on function F_{20}. The number of (Best/2nd Best/Worst) is counted for each algorithm and worst ranks are underlined in Table 5. It can be observed that HCLPSO algorithm has no worst performance on any benchmark functions. Overall, the proposed HCLPSO algorithm provides the best performance on 15 out of 25 benchmark problems and rank first over other PSO algorithms except for OLPSO algorithm. The 10 dimensional Wilcoxon signed rank test results of HCLPSO algorithm against other algorithms are shown in Table 6. The number of (+/0/−) are presented in last row of Wilcoxon table and it can be observed that HCLPSO performs significantly better than other PSO algorithms.

Table 6
Wilcoxon signed rank test results of single-problem analysis with a significance level of $\alpha=0.05$ for 10 dimensional problems.

Functions	Pairwise comparison HCLPSO versus					
	PSO	FIPS	UPSO	CLPSO	HPSO-TVAC	sHPSO
F_1	+	+	+	0	+	+
F_2	+	+	+	0	+	+
F_3	+	+	+	0	+	+
F_4	0	+	+	0	+	+
F_5	+	+	+	+	+	+
F_6	+	+	+	0	+	+
F_7	+	+	+	+	+	+
F_8	0	+	+	+	+	+
F_9	+	+	+	+	+	+
F_{10}	+	+	+	+	+	+
F_{11}	+	+	+	0	+	+
F_{12}	+	+	+	0	+	+
F_{13}	+	+	+	0	+	+
F_{14}	0	+	+	0	+	+
F_{15}	0	+	+	0	+	+
F_{16}	+	+	+	+	+	+
F_{17}	+	+	+	0	+	+
F_{18}	+	+	+	0	+	+
F_{19}	+	+	+	+	+	+
F_{20}	+	+	+	+	+	+
F_{21}	+	+	+	+	+	+
F_{22}	+	+	+	+	+	+
F_{23}	0	0	+	0	+	+
F_{24}	+	+	+	+	+	+
F_{25}	+	+	+	+	+	+
+/0/−	20/5/0	24/1/0	25/0/0	12/13/0	25/0/0	25/0/0

4.3.2. *Results for 30 dimensional problems*

As described in comparison Table 7, all algorithms perform well on the shifted sphere function F_1. UPSO obtains the best performance on function F_2. HCLPSO achieves the best solution on function F_3. PSO performs the best on function F_4 and FIPS provides the best performance on function F_5. The proposed HCLPSO algorithm is consistently performed well and obtained the best and the second best performance on all unimodal functions F_1–F_5.Overall, it provides outstanding and robust performance over other PSO algorithms.

In multimodal functions, HCLPSO performs well on all the shifted and rotated multimodal functions. The proposed HCLPSO provides the best performance on multimodal F_6, F_9, F_{10}, F_{11}, F_{12} and second best on multimodal F_7 and F_8. On the other hand, CLPSO yields the best solution on F_6 as OLPSO and HPSO-TVAC does the best on function F_7 and sHPSO does on function F_8. HPSO-TVAC offers the best solution on F_{12}. Hence, HCLPSO is consistently performing well throughout multimodal problems and obtains the best overall performance. In expanded hybrid composite test functions, HCLPSO performs the best on both expanded extended function F_{13} and shifted rotated expanded function F_{14}.

In hybrid composite test functions, HCLPSO yields the best performance on hybrid composition function of F_{16}, F_{17}, F_{21}, F_{23}, F_{24} and F_{25}. On the other hand, FIPS gives the best results on function F_{18}, F_{19}, F_{20} and F_{22}. Similarly, OLPSO performs the best on F_{21}, F_{23} and F_{24} and CLPSO does the best on F_{15}, F_{21}, F_{23}, F_{24} and F_{25}. However, HCLPSO performs consistently within top three ranges throughout 11 problems except for functions F_{19} and F_{20}. According to final rank and the number of Best/2nd Best/Worst, proposed HCLPSO ranked first by performing the best 15 out of 25 functions

Table 7
Comparison of experimental results of PSO algorithms for 30 dimensional CEC 2005 test functions with $g_1=15$ and $g_2=25$.

F	Criteria	HCLPSO	PSO	FIPS	UPSO	CLPSO	OLPSO	HPSO-TVAC	sHPSO
F_1	Mean	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	Std	0	0	0	0	0	0	0	0
	Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
F_2	Mean	1.70E–06	0.37	77.94	**2.65E–07**	1.14E+03	13.79	3.79E–06	1.44E–02
	Std	1.71E–06	0.32	27.05	2.42E–07	2.53E+02	8.33	2.82E–06	7.10E–02
	Rank	2	5	7	**1**	8	6	3	4
F_3	Mean	**6.42E+05**	6.53E+06	2.45E+07	1.54E+06	1.22E+07	1.60E+07	7.72E+05	8.75E+05
	Std	2.61E+05	4.17E+06	6.29E+06	4.75E+05	3.34E+06	7.04E+06	2.96E+05	5.34E+05
	Rank	**1**	5	8	4	6	7	2	3
F_4	Mean	5.22E+02	**3.81E+02**	1.15E+03	7.28E+03	8.77E+03	2.18E+03	2.48E+04	2.02E+04
	Std	3.09E+02	3.31E+02	3.73E+02	2.79E+03	1.85E+03	1.09E+03	5.71E+03	9.94E+03
	Rank	2	**1**	3	5	6	4	8	7
F_5	Mean	2.97E+03	3.85E+03	**2.22E+03**	6.32E+03	4.47E+03	3.30E+03	9.20E+03	6.94E+03
	Std	4.55E+02	8.00E+02	5.14E+02	1.63E+03	4.26E+02	3.75E+02	1.81E+03	1.43E+03
	Rank	2	4	**1**	6	5	3	8	7
F_6	Mean	**2.39**	70.16	37.70	68.20	**2.39**	20.68	50.41	115.27
	Std	4.27	95.09	35.03	96.41	3.84	24.97	50.54	228.56
	Rank	**1**	6	3	5	**1**	2	4	7
F_7	Mean	0.02	0.76	0.03	0.02	0.70	**0.01**	**0.01**	0.04
	Std	0.02	1.41	0.02	0.01	0.15	0.01	0.01	0.04
	Rank	2	6	3	2	5	**1**	**1**	4
F_8	Mean	20.87	20.90	20.94	20.95	20.92	20.96	20.71	**20.18**
	Std	0.09	0.07	0.06	0.05	0.06	0.08	0.15	0.19
	Rank	3	4	6	7	5	8	2	**1**
F_9	Mean	**0**	19.00	57.11	85.16	**0**	**0**	10.71	82.54
	Std	0	5.37	14.55	16.90	0	0	4.96	24.35
	Rank	**1**	3	4	6	**1**	**1**	2	5
F_{10}	Mean	**56.08**	100.32	177.86	93.63	99.78	109.23	275.73	242.83
	Std	12.90	58.63	9.25	18.82	12.48	18.94	45.49	89.17
	Rank	**1**	4	6	2	3	5	8	7
F_{11}	Mean	**20.32**	23.18	38.36	30.94	24.42	25.05	30.62	32.29
	Std	2.94	2.81	1.52	1.96	1.78	3.14	2.58	3.84
	Rank	**1**	2	8	6	3	4	5	7
F_{12}	Mean	**3.91E+03**	5.07E+04	5.62E+04	4.31E+03	1.47E+04	1.22E+04	4.47E+03	2.43E+04
	Std	3.69E+03	3.50E+04	2.00E+04	3.44E+03	3.45E+03	5.42E+03	4.66E+03	2.66E+04
	Rank	**1**	7	8	2	5	4	3	6
F_{13}	Mean	**1.45**	3.02	12.51	5.78	1.86	1.86	3.85	6.14
	Std	0.28	0.69	0.96	1.59	0.18	0.28	1.11	2.31
	Rank	**1**	4	8	6	2	3	5	7
F_{14}	Mean	**11.93**	12.65	13.12	12.79	12.64	13.13	12.31	13.07
	Std	0.58	0.43	0.21	0.33	0.22	0.20	0.37	0.39
	Rank	**1**	4	7	5	3	8	2	6
F_{15}	Mean	88.04	299.71	283.02	465.59	**40.10**	244.88	321.29	457.16
	Std	113.02	107.62	39.65	70.22	37.90	84.09	107.12	138.94
	Rank	2	5	4	8	**1**	3	6	7
F_{16}	Mean	**104.07**	235.84	201.66	199.42	163.61	135.54	336.21	376.70
	Std	36.41	152.98	14.63	61.91	27.87	55.48	97.20	99.46

Table 7 (*continued*)

F	Criteria	HCLPSO	PSO	FIPS	UPSO	CLPSO	OLPSO	HPSO-TVAC	sHPSO
	Rank	**1**	6	5	4	3	2	7	8
F_{17}	Mean	**109.59**	262.63	224.39	278.90	219.31	204.56	391.68	367.25
	Std	34.01	153.27	10.71	79.14	40.09	46.67	97.34	133.86
	Rank	**1**	5	4	6	3	2	8	7
F_{18}	Mean	894.42	927.77	**832.24**	840.92	907.76	906.63	908.24	997.99
	Std	43.04	2.86	1.78	6.18	26.75	20.23	79.00	60.04
	Rank	3	7	**1**	2	5	4	6	8
F_{19}	Mean	913.49	926.69	**831.09**	841.77	915.09	906.73	912.45	982.94
	Std	2.39	2.49	1.50	7.42	1.36	20.25	79.78	78.69
	Rank	5	7	**1**	2	6	3	4	8
F_{20}	Mean	914.03	927.06	**831.86**	841.62	911.24	906.40	905.92	974.13
	Std	2.36	2.35	1.66	6.22	20.60	20.19	74.72	67.96
	Rank	6	7	**1**	2	5	4	3	8
F_{21}	Mean	**500.00**	510.00	862.53	761.78	**500.00**	**500.00**	1173.50	996.01
	Std	0	54.77	1.03	174.96	0	0	229.74	328.97
	Rank	**1**	2	4	3	**1**	**1**	6	5
F_{22}	Mean	910.68	938.66	**521.17**	615.37	974.32	941.70	1216.89	1123.41
	Std	15.75	12.96	1.27	113.50	17.22	16.30	73.68	103.68
	Rank	3	4	**1**	2	6	5	8	7
F_{23}	Mean	**534.16**	574.17	865.57	816.07	**534.16**	**534.16**	1258.30	996.79
	Std	4.07E−04	122.10	1.03	147.72	1.53E−04	5.04E−04	34.57	323.48
	Rank	**1**	2	4	3	**1**	**1**	6	5
F_{24}	Mean	**200.00**	**200.00**	214.39	225.24	**200.00**	**200.00**	1276.11	995.63
	Std	0	0	0.58	4.87	0	0	205.10	409.92
	Rank	**1**	**1**	2	3	**1**	**1**	5	4
F_{25}	Mean	**200.00**	1113.45	214.21	225.56	**200.00**	207.46	1331.03	1037.43
	Std	0	14.41	0.50	7.19	0	7.09	26.00	423.59
	Rank	**1**	6	3	4	**1**	2	7	5
Average rank		1.8	4.32	4.12	3.88	3.48	3.4	4.8	5.76
Final rank		**1**	6	5	4	3	2	7	8
Best/2nd Best/Worst		**15/5/0**	3/3/1	5/1/4	1/7/2	8/1/1	6/4/2	2/4/9	2/0/5
Algorithms		HCLPSO	PSO	FIPS	UPSO	CLPSO	OLPSO	HPSO-TVAC	sHPSO

Table 8
Wilcoxon signed rank test results of single-problem analysis with a significance level of α=0.05 for 30 dimensional problems.

Functions	Pairwise comparison HCLPSO versus						
	PSO	FIPS	UPSO	CLPSO	OLPSO	HPSO-TVAC	sHPSO
F_1	0	0	0	0	0	0	0
F_2	+	+	−	+	+	+	0
F_3	+	+	+	+	+	0	0
F_4	0	+	+	+	+	+	+
F_5	+	−	+	+	+	+	+
F_6	+	+	+	0	+	+	+
F_7	+	+	0	+	0	0	0
F_8	0	+	+	+	+	−	+
F_9	+	+	+	0	0	+	+
F_{10}	+	+	+	+	+	+	+
F_{11}	+	+	+	+	+	+	+
F_{12}	+	+	+	+	+	+	+
F_{13}	+	+	+	+	+	+	+
F_{14}	+	+	+	+	+	+	+
F_{15}	+	+	+	0	+	+	+
F_{16}	+	+	+	+	0	+	+
F_{17}	+	+	+	+	+	+	+
F_{18}	+	−	−	+	0	0	+
F_{19}	+	−	−	+	−	0	+
F_{20}	+	−	−	0	−	−	+
F_{21}	+	+	+	0	0	+	+
F_{22}	+	−	−	+	+	+	+
F_{23}	+	+	+	−	0	+	+
F_{24}	0	+	+	0	0	+	+
F_{25}	+	+	+	+	+	+	+
+/0/−	21/4/0	19/1/5	18/2/5	17/7/1	15/8/2	18/5/2	21/4/0

and achieved comparable performance on other unimodal, multimodal, expanded and hybrid composite functions. The last row in Wilcoxon sign rank test Table 8 also shows that HCLPSO performs better than other PSO algorithms on most of the 30 dimensional problems.

Among unimodal functions, the convergence graph of Schwefel's Problem 2.6 with Global Optimum on Bounds (F_5) is presented in Fig. 3(a). It can be clearly observed that the proposed HCLPSO obtains better result than other algorithms. Among multimodal functions, the convergence graphs of the rotated and shifted Rastrigin's function and Weierstrass functions (F_{10} and F_{11}) are presented in Fig. 3(b) and (c) respectively. In both cases, the proposed HCLPSO algorithm is able to jump out of local minimum and achieves the best performance.

Among hybrid composite functions, the convergence graphs of F_{17}, F_{19} and F_{20} are presented in Fig. 3(d), (e) and (f) respectively. Fig. 3(d) shows that the proposed HCLPSO achieves the outstanding performance. However, the HCLPSO obtains the comparable performance to other algorithms in F_{19} and F_{22}. It can be observed in Fig. 3(e) and (f). In summary, the convergence analysis shows that the proposed algorithm, HCLPSO, outperforms in most test cases.

4.4. Discussions

Experimental results and comparisons verify that HCLPSO overall performs better than traditional PSO and other improved PSO variants on shifted rotated unimodal, multimodal, expanded and hybrid composite problems, in terms of solution accuracy (mean and standard deviation), convergence test and statistical analysis results. HCLPSO performs well consistently and achieves high performance on all shifted rotated unimodal and multimodal problems. The algorithm also offers the best performance on expanded and shifted rotated expanded problems as well as on hybrid and rotated hybrid composition problems. However, the algorithm does not perform well on two rotated hybrid composition functions. This can be explained by No Free Lunch Theorem (NFLT) [39]. According to no free lunch theorem, a general-purpose universal optimization algorithm is theoretically impossible. Therefore, no strategy can be expected to outperform another on all types of optimization problems.

When comparing with CLPSO, the results show that HCLPSO can find more accurate solutions with faster convergence speed in all unimodal, multimodal, expanded and hybrid composite functions. Thus, HCLPSO offers higher solution accuracy and faster convergence speed than CLPSO. Compared to other PSO algorithms, HCLPSO performs the best in 15 out of 25 problems and comparable performance on the rest.

This implies that HCLPSO has gained benefits from four methods described in Section 1. By using heterogeneous subpopulations which are assigned for exploration and exploitation processes separately, it enhances the exploration and exploitation processes without crippling one another. By using CL learning strategy and its velocity update equation [14], the particles can learn from different dimensions of different particles and avoid two problems, "oscillation phenomenon" [28] and "two steps forward, one step back phenomenon" [25]. With generating the exemplar for the exploration subpopulation by using its own members' best experiences, information sharing among the particles is controlled and able to prevent premature convergence. Lastly, by using time varying acceleration coefficients [7] for each subpopulation group, their respective exploration/exploitation ability is enhanced. Therefore, the trade-off between exploration and exploitation is achieved and the HCLPSO performs better or comparable to other PSO variants on shifted rotated unimodal, multimodal, expanded and hybrid composition functions.

One of the limitations on HCLPSO algorithm is using CL strategy for solving unimodal problems. Although learning of different dimensions from different exemplars is efficient for searching the global optimum of multimodal problems, tuning the algorithm to find the optimal solution on unimodal problems is ineffective and costs low convergence speed. This is expected as especially, in the light of no free lunch theorem, there is no general-purpose optimization algorithm which can solve all classes of problems and all the algorithms have pros and cons on solving optimization problems. Secondly, the HCLPSO is more complex compare to other algorithms and requires additional computations. However, the computational resources are mostly spent on the evaluating of objective function in EA algorithms. Hence, the additional computational requirement of the optimization strategy for HCLPSO is negligible. In addition, the HCLPSO offers faster convergence rate and this can be translated into additional saving on computation time.

Though HCLPSO is a promising optimization algorithm, as mentioned above comprehensive learning strategy has to be tuned to get better performance. To address this issue, adaptive or self-learning strategy will be investigated in the future.

5. Conclusion

In this paper, heterogeneous CLPSO with enhanced exploration and exploitation is presented in which the entire population is divided into two heterogeneous subpopulation groups and each subpopulation group is especially designed to enhance exploration and exploitation performances, respectively. In addition, the explorative particles do not interact with the exploitative particles to prevent loss of diversity. Furthermore, inertia weight and time varying acceleration coefficients are used in both groups to bias the exploration and exploitation balance from exploration in early stages to exploitation in the later stages. Intensive performance evaluation of the proposed HCLPSO algorithm is conducted using benchmark optimization problems. The experiment results showed that the proposed algorithm overall outperforms the current state-of-art PSO variants.

Acknowledgment

The authors would like to thank Zhi-hui Zhan, a co-author of [15], for providing simulation result of OLPSO algorithm on CEC 2005 30 dimensional problems.

References

[1] E. Eiben, C. Schippers, On evolutionary exploration and exploitation, Fundam. Inf. 35 (1998) 35–50.
[2] M. Črepinšek, S-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: a survey, ACM Comput. Surv. (CSUR) 45 (no. 3) (2013), article no. 35, pp. 1-33.
[3] J. Kennedy, R.C. Eberhart, Particle swarm optimization, Proc. IEEE Int. Conf. Neural Netw. 4 (1995) 1942–1948.
[4] Y.H. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE Congress on Evolutionary Computation, 1998, pp. 69–73.
[5] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation, 1999, pp. 1945–1950.
[6] M. Clerc, J. Kennedy, The particle swarm – explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (no. 1) (2002) 58–73.
[7] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (no. 3) (2004) 240–255.
[8] J. Kennedy, Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance, in: Proceedings of the IEEE Congress on Evolutionary Computation, 1999, pp. 1931–1938.
[9] P.N. Suganthan, Particle swarm optimizer with neighborhood operator, in: Proceedings of the IEEE Congress on Evolutionary Computation, Washington, DC, 1999, pp. 1958–1962.
[10] J. Kennedy, R. Mendes, Population structure and particle swarm performance, in: Proceedings of the IEEE Congress on Evolutionary Computation, Honolulu, HI, 2002, pp. 1671–1676.
[11] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, IEEE Trans. Evol. Comput. 8 (2004) 204–210.
[12] J. Kennedy, R. Mendes, Neighborhood topologies in fully informed and best-of-neighborhood particle swarm, IEEE Trans. Syst., Man Cybern. C 36 (no. 4) (2006) 515–519.
[13] K.E. Parsopoulos, M.N. Vrahatis, editors, UPSO: a Unified Particle Swarm Optimization Scheme, Lecture Series on Computational Sciences, 2004, pp. 868–873.
[14] J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (no. 3) (2006) 281–295.
[15] Z.H. Zhan, J. Zhang, Y. Lin, Y.H. Shi, Orthogonal learning particle swarm optimization, IEEE Trans. Evol. Comput. 15 (no. 6) (2011) 832–847.
[16] S. Hsieh, T. Sun, C. Liu, S. Tsai, Efficient population utilization strategy for particle swarm optimizer, IEEE Trans. Syst., Man, Cybern. B, Cybern 39 (no. 2) (2009) 444–456.
[17] Z. Ren, A. Zhang, C. Wen, Z. Feng, A scatter learning particle swarm optimization algorithm for multimodal problems, IEEE Trans. Cybern. 44 (no. 7) (2014) 1127–1140.
[18] R. Cheng, Y. Jin, A competitive swarm optimizer for large scale optimization, IEEE Trans. Cybern. 45 (no. 2) (2015) 191–204.
[19] A. Silva, A. Neves, E. Costa, An empirical comparison of particle swarm and predator prey optimization, in: Proceedings of the 13th Irish International Conference on Artificial Intelligence and Cognitive Science (AICS 2002), 2002, pp. 103–110.
[20] J. Riget, J.S. Vesterstrom, Diversity-Guided Particle Swarm Optimizer – The ARPSO, Department of Computer Science, University of Aarhus, Denmark, EVA Life Technical Report no. 2002, 2002.
[21] T.M. Blackwell, P.J. Bentley, Dynamic search with charged swarms, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2002, pp. 19–26.
[22] M.A. Montes de Oca, J. Pe~na, T. Sẗutzle, C. Pinciroli, M. Dorigo, Heterogeneous particle swarm optimizers, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2009, pp. 698–705.
[23] A.P. Engelbrecht, Heterogeneous Particle Swarm Optimization, in: Proceedings of the 7th International Conference on Swarm Intelligence (ANTS 2010), 2010, pp. 191–202.
[24] E. Dor, M. Clerc, P. Siarry, A multi-swarm PSO using charged particles in a partitioned search space for continuous optimization, Comput. Optim. Appl. 53 (no.1) (2012) 271–295.
[25] F. van den Bergh, A.P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Trans. Evol. Comput. 8 (no. 3) (2004) 225–239.
[26] R. Brits, A.P. Engelbrecht, F. van den Bergh, A niching particle swarm optimizer, in: Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, 2002, pp. 692–696.
[27] T.M. Blackwell, J. Branke, Multi-swarms, exclusion, and anti-convergence in dynamic environments, IEEE Trans. Evol. Comput. 10 (no. 4) (2006) 459–472.
[28] K.E. Parsopoulos, M.N. Vrahatis, On the computation of all global minimizers through particle swarm optimization, IEEE Trans. Evol. Comput. 8 (no. 3) (2004) 211–224.
[29] R.C. Eberhart, Y.H. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2000, pp. 84–88.
[30] O. Olorunda, A. Engelbrecht, Measuring exploration/exploitation in particle swarms using swarm diversity, in: Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, 2008, pp. 1128–1134.
[31] A.P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, John Wiley & Sons Ltd., 2005.
[32] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.-P. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2005, pp. 1–50.
[33] J. Kennedy, The particle swarm: social adaptation of knowledge, in: Proceedings of the IEEE International Conference on Evolutionary Computation, 1997, pp. 303–308.
[34] J. Kennedy, Bare bones particle swarms, in: Proceedings of the IEEE Swarm Intelligence Symposium, 2003, pp. 80–87.
[35] T. Bartz-Beielstein, Experimental Research in Evolutionary Computation, Springer, New York, 2006.
[36] M. Hollander, D. Wolfe, Nonparametric Statistical Methods, 2nd ed., John Wiley & Sons Inc., 1999.
[37] J.D. Gibbons, S. Chakraborti, Nonparametric Statistical Inference, 5th ed., Chapman & Hall, 2010.
[38] J. Derrac, S. Garcia, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (no. 1) (2011) 3–18.
[39] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82.