Applied Soft Computing 55 (2017) 533–548

Contents lists available at ScienceDirect

Applied Soft Computing

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

Ensemble particle swarm optimizer

Nandar Lynn, Ponnuthurai Nagaratnam Suganthan*

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

ARTICLE INFO

Article history:
Received 30 July 2015
Received in revised form
28 December 2016
Accepted 2 February 2017
Available online 8 February 2017

Keywords:
Ensemble
Particle swarm optimization
Real-parameter optimization
Self-adaptive
Strategy adaptation

ABSTRACT

According to the "No Free Lunch (NFL)" theorem, there is no single optimization algorithm to solve every problem effectively and efficiently. Different algorithms possess capabilities for solving different types of optimization problems. It is difficult to predict the best algorithm for every optimization problem. However, the ensemble of different optimization algorithms could be a potential solution and more efficient than using one single algorithm for solving complex problems. Inspired by this, we propose an ensemble of different particle swarm optimization algorithms called the ensemble particle swarm optimizer (EPSO) to solve real-parameter optimization problems. In each generation, a self-adaptive scheme is employed to identify the top algorithms by learning from their previous experiences in generating promising solutions. Consequently, the best-performing algorithm can be determined adaptively for each generation and assigned to individuals in the population. The performance of the proposed ensemble particle swarm optimization algorithm is evaluated using the CEC2005 real-parameter optimization benchmark problems and compared with each individual algorithm and other state-of-the-art optimization algorithms to show the superiority of the proposed ensemble particle swarm optimization (EPSO) algorithm.

1. Introduction

Optimization problems can be found in all areas of science and engineering. Depending on the nature of the optimization problems (variables, objective functions, constraints and certainty/uncertainty), it can be classified into continuous/discrete, constrained/unconstrained, single/multi objective, convex/non-convex, differentiable/non-differentiable and deterministic/stochastic optimization problems. In order to solve these diverse problems, many optimization techniques, such as particle swarm optimization (PSO), genetic algorithm (GA), differential evolution (DE), evolutionary strategy (ES), evolutionary programming (EP), ant colony optimization (ACO) algorithm and artificial bee colony algorithm (ABC) have been developed. The performance of the evolutionary algorithms (EA) and swarm intelligence (SA) algorithms had been demonstrated using real-parameter optimization benchmark problems [1–7] and successfully applied to solve real-world applications [8–13]. However, it is difficult to draw a conclusion that those successful and well-known optimization algorithms will perform well on every optimization problem. In order to find the best algorithm for a particular problem, numerous trial-and-error runs are required and this trail-and-error process requires unrealistic computational time. This problem has been highlighted in "No Free Lunch (NFL) theorem" [14] which states that no single state-of-the-art optimization algorithm can be expected to perform better than any other algorithm on all classes of optimization problems. Instead of such a trial-and-error-based search and relying on a single optimization algorithm, the ensemble of different optimization algorithms could potentially bring their advantages to light and enable us to solve a variety of complex optimization problems. Motivated by this observation, an ensemble of PSO algorithms with strategy adaptation called the ensemble particle swarm optimizer (EPSO) is proposed in this paper. In the proposed EPSO algorithm, the self-adaptive evaluation strategy is employed to encourage meritocracy among the competing algorithms. Different PSO variants are promoted based on their merits to take the leading role in finding better solutions. Instead of combining of different EAs, our approach is completely focused on ensemble of different PSO variants to encourage the simplicity and ease of implementation for real-world applications.

In the proposed EPSO, different PSO variants, namely *inertia weight* PSO [15], comprehensive learning PSO (CLPSO) [16], fitness-distance-ratio based PSO (FDR-PSO) [17], self-organizing hierarchical PSO (HPSO-TVAC) [18] and distance-based locally informed PSO (LIPS) [19] are hybridized to complement each other. PSO algorithm with inertia weight is used to ensure fast convergence speed. HPSO-TVAC is used to prevent the premature

* Corresponding author.
E-mail addresses: nandar001@ntu.edu.sg (N. Lynn), epnsugan@ntu.edu.sg (P.N. Suganthan).

http://dx.doi.org/10.1016/j.asoc.2017.02.007
1568-4946/

convergence. In addition, the CLPSO is also used to maintain the population diversity throughout the optimization process. Furthermore, FDR-PSO is employed to enhance local search capability. Finally, LIPS is deployed to search multiple local optima in multimodal optimization. The details of these algorithms are discussed in Section 3. In terms of the problem type, *inertia weight* PSO and HPSO-TVAC are effective for solving unimodal problems while CLPSO, LIPS and FDR-PSO are effective for solving multimodal problems. Therefore, it can be easily seen that different PSO variants with complementary capabilities are combined in the proposed EPSO to enhance generalization and robust performance.

In order to select the most suitable PSO variant in the current phase of the search process, the self-adaptive mechanism from [20] is introduced in the proposed EPSO algorithm. In EPSO, a pool of PSO strategies is constructed and a probabilistic selection is performed based on the success rate of each PSO strategy in generating improved solutions. In the beginning of the search, each PSO strategy has equal probability to be selected and stochastic universal sampling selection [21] is used to select a PSO strategy from the pool. In EPSO, a fix number of generations called learning period (LP) is defined for PSO strategies as in [20]. Within LP period, if a PSO strategy can generate improved solutions, it is recorded in its success memory. On the other hand, if it fails to generate improved solutions, it is recorded in its failure memory. After LP generations, based on those success and failure memories, the success rate of each PSO strategy is updated at each subsequent generation, followed by updating the selection probability. The PSO strategy with higher success rate will have larger probability to be chosen. In this way, a suitable PSO strategy can be determined adaptively to match the current stage of the search process and assigned to each individual particle in the swarm for updating their search direction.

This paper is organized as follows: The overview of PSO is described in Section 2.1 and a literature review on ensemble approaches involving PSO and other EAs are presented in Sections 2.2 and 2.3, respectively. The proposed EPSO algorithm is introduced in Section 3. Different PSO strategies employed in EPSO are briefly explained in Section 3.1 and EPSO's self-adaptive selection strategy is presented in Section 3.2. In Section 4, the proposed EPSO algorithm is evaluated using a set of shifted and/or rotated benchmark functions and compared with its individual variants and a heterogeneous PSO algorithm. Finally, the paper is concluded in Section 5.

2. Review of PSO and its related ensemble approaches

2.1. Particle swarm optimization

Among the population-based evolutionary and swarm intelligence algorithms, PSO has attracted considerable attention due to its simplicity and fast convergence speed [22,23] and PSO variants have been successfully applied to solve real world optimization problems [24–28]. Particle swarm optimization (PSO) is a population-based search optimization technique that was introduced by Eberhart and Kennedy [23]. The basic principle of PSO mimics swarm social behavior such as bird flocking and fish schooling. In PSO, each particle in the swarm represents the potential solution to an optimization problem. Initially, the particles are randomly distributed over the search space with random velocity values. Then, each particle's velocity is updated using its own previous best experience called *pbest* and the whole swarm's best experience called *gbest*. This phenomenon was formulated in [23] as follows:

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

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

where, $d = 1,2,\ldots,D$ and D represents the dimension. The i represents each particle in the population ($i = 1, 2,\ldots, N$). X_i and V_i are the position and velocity components of i^{th} particle respectively. $pbest_i$ is the previous best position of the i^{th} particle and *gbest* is the whole swarm's best position. The c_1 and c_2 are acceleration coefficient. $rand1_i$ and $rand2_i$ are randomly generated numbers in the range of [0,1]. In this way, PSO searches the global best solution by simply adjusting the trajectory of each particle towards its own best and toward the best particle of the swarm at each generation. Unfortunately, the originally PSO is well known for its premature convergence to a local optimum.

In light of premature convergence of the original PSO, Shi and Eberhart incorporated a new parameter called inertia weight w into the PSO algorithm [29]. Inertia weight w is defined as a linearly decreasing function of time to adjust the balance between the global search and local search. In *inertia weight* PSO, the position and velocity of the particles are updated according to the following equation [29]:

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

The position of the particle is updated using Eq. (2). A large value of inertia weight w encourages the global search while a small value of w enhances the local search during the search process. By decreasing inertia weight w linearly from a relatively large value to a small value through the optimization process, *inertia weight* PSO has more global search ability at the early stages of the optimization process and more local search ability at the latter stages. Compared to the original PSO, *inertia weight* PSO converged quickly to the global optima and performed significantly better [29]. However, with the decreasing small w value, *inertia weight* PSO lacks the global search ability at the latter stages of the search even when the global search ability is required to jump out of the local optimum, especially in the case of solving multimodal problems.

Many PSO variants have been proposed to improve the PSO performance. However, it is extremely difficult to find a generalized PSO variant that performs well on various types of optimization problems. For instance, *inertia weight* PSO performs well on the unimodal problems, whereas it performs very badly on complex multimodal problems [15,29] due to its premature convergence to a local optimum. As an alternative, popular comprehensive learning particle swarm optimizer offers best performance on multimodal problems though its convergence rate on the unimodal problems is unsatisfactory [16]. The recent global orthogonal learning PSO algorithm performs well on the unimodal problems, whereas local orthogonal learning PSO provides better performance on the multimodal problems [30]. As highlighted, different algorithms have their own pros and cons. Each PSO algorithm is suitable for the specific nature of the problem. By hybridizing these different PSO variants together, they can benefit from each other and enable us to solve a wide range of optimization problems. In light of this, various ensemble approaches have been developed and these approaches are reviewed in next section.

2.2. Ensemble approaches used in PSO

To address universality and robust performance issue of PSO algorithms, recently the researchers have introduced the ensemble approach to PSO algorithms. Du and Li proposed multi-strategy ensemble PSO (MEPSO) for dynamic optimization [31]. In [31], the particles were divided into two parts, part I and II, and the two different strategies were implemented in each part. Gaussian local search strategy was introduced in part I to enhance the convergence ability and differential mutation strategy was used in part II to enhance the diversity. The experimental results showed that multi-strategy ensemble PSO offered good performance on all different

dynamic optimization problems and outperformed other dynamic optimization PSO algorithms.

In [32], Engelbrecht proposed the heterogeneous PSO (HPSO) algorithm. In HPSO, a pool of different PSO search behaviors is constructed and the particles are allocated by randomly selecting the velocity update rules from the behavior pool. The empirical results proved that HPSO with ensemble of different PSO search behaviors outperformed homogeneous PSO algorithms in terms of solution quality and scalability. The static and dynamic heterogeneous PSO algorithms were also proposed in [32,33]. Nepomuceno proposed the frequency-based heterogeneous PSO called f_k-PSO algorithm in [34]. As in the HPSO algorithm, different PSO variants are employed to construct the behavior pool. However, unlike HPSO algorithm, the success counter is assigned to each behavior. During the search process, the behavior that frequently performs well has higher probability to be selected to perform the search [34]. The performance of f_k-PSO algorithm was tested using a set of different benchmark optimization problems and the results showed that f_k-PSO obtained superior performance against compared single PSO algorithms.

Therefore, it can be clearly observed that hybridizing these different PSO algorithms together could be beneficial to solve a wide range of optimization problems.

2.3. Ensemble approaches used in EAs

Olorunda and Engelbrecht investigated the possibility of combing different evolutionary algorithms with the concept of cooperation. The authors proposed heterogeneous cooperative algorithm in which DE, GA and PSO algorithms are combined using the cooperative model from [35] and the useful search characteristics of each participating algorithm are exploited to find the solutions of various optimization problems [36]. Recently, the researchers have used self-adaptive mechanism to enhance the performance of evolutionary optimization algorithms. Vrugt introduced a concept of self-adaptive multi-method evolutionary search and proposed the multialgorithm genetically adaptive method (AMALGAM) [37]. In [37], different optimization methods such as covariance matrix adaptation evolutionary strategy, parental-centric recombination operator, PSO, GA and DE were used for population evolution and the optimization methods with the high productive search were adaptively selected to generate more offspring during the search. The performance of AMALGAM algorithm was evaluated using a set of standard benchmark functions [2] and the experimental results demonstrated that the robustness and efficiency of the evolutionary search was improved by the concept of the self-adaptive multi-method optimization [37].

A heterogeneous meta-hyper-heuristic (HMHH) algorithm was proposed in [38] where GA, PSOs (standard PSO and bare bones PSO) and DE were used as meta-heuristic sub-algorithms. The selection strategy was applied to select the most appropriate sub-algorithm at each generation and an acceptance strategy was employed to determine whether the new candidate solution should replace the old candidate solution. In [38], the random, performance-based [38] and tabu-search based [39] approaches were investigated as the alternative selection strategies. Deterministic and stochastic were studied as the alternative acceptance strategies. The performance of HMHH algorithm was tested on a set of five benchmark problems and HMHH algorithm obtained promising results in terms of solution quality and algorithm robustness [38].

Spears applied self-adaptive strategy to GA algorithm to select between two different crossovers and it was observed that a larger set of crossover operators may be beneficial for evolutionary algorithms [40]. This curiosity whether the combinations of multiple crossovers outperform over the single crossover or not was investigated in [41]. The synergy of multiple operators was studied and the experiments were conducted on travelling salesman problem and graph bisection problem. The study showed that the careful combinations of multiple crossover operators produced synergy and provided better results compared to using a single crossover operator [41]. In order to avoid the time-consuming trial-and-error process, Qin and Suganthan proposed self-adaptive differential DE algorithm in which different trial vector generation strategies and their associated parameter settings were gradually self-adapted by learning from their previous experiences in generating promising solutions [20,42]. As a result, the better trial vector generation strategy along with its associated parameters is selected adaptively to match different stages of the search process. Ensemble strategies with adaptive EP was introduced in [43] where the benefits of different mutation strategies were exploited for the different stages of the search process.

Multi-strategy ensemble ABC (MEABC) algorithm was recently proposed [44,45]. In [44], a strategy pool of was constructed by employing three different ABC solution search strategies. The strategy for each food source is dynamically changed based on the quality of the new solutions during the search process. If the quality of a new candidate solution was better than the quality of the current candidate solution, the current strategy was selected for the follow-up search process. If it failed to keep improving the quality of the solutions, the new strategy was randomly selected from the strategy pool. The performance of MEABC algorithm was evaluated using shifted and rotated benchmark problems [5] and the experimental results demonstrated that the ensemble of multi-ABC-strategy offered higher chances of finding better new candidate solutions than using a single ABC strategy.

3. Ensemble particle swarm optimizer (EPSO)

Depending on the characteristics of the algorithm as well as of the problem type, the performance of an optimization algorithm varies. Consequently, it is very difficult to identify the suitable optimization algorithm for a given problem. Hence, a high-level heuristic optimization technique that aims to solve different classes of problems is proposed in this paper by forming an ensemble of PSO algorithms, namely *inertia weight* PSO, HPSO-TVAC, FDR-PSO, LIPS and CLPSO algorithms. *Inertia weight* PSO is briefly described in Section 2.1 while other PSO strategies used in the proposed EPSO algorithm are briefly explained in Section 3.1.

In the proposed EPSO algorithm, the population is divided into two subpopulation groups, small and large subpopulations, to maintain the diversity throughout the optimization process. The CLPSO algorithm is assigned for the small subpopulation whereas the inertia weight PSO, HPSO-TVAC, FDR-PSO, LIPS and *gbest*-guided CLPSO algorithms are employed in the large subpopulation of the ensemble approach. This population structure will enable the proposed algorithm to maintain diversity throughout the optimization process while maintaining a competitive convergence rate as well. To update a particle, a candidate PSO strategy in the large subpopulation is selected according to the success ratio of each PSO strategy in the recent past iterations in a self-adaptive manner [20]. Details are presented in Section 3.2.

3.1. PSO variants employed in EPSO

3.1.1. Comprehensive learning particle swarm optimizer (CLPSO)

In PSO, the trajectory towards the global optimum is adjusted by the particles' own $pbest_i^d$ and $gbest^d$. Though *gbest* is the whole population's best experience, it may be an inferior local optimum for a multimodal problem and far from the global optimum. To address this issue, a comprehensive learning particle swarm optimizer (CLPSO) was proposed in [16]. In CLPSO, all particles' best

experiences are used to guide the search of a particle. A particle learns from different particles' *pbests* for different dimensions. The velocity of i^{th} particle is updated using the following equation [16]:

$$V_i^d = wV_i^d + c * rand_i^d * \left(pbest_{fi(d)}^d - X_i^d\right) \quad (4)$$

where, $f_i(d) = [f_i(1), f_i(2), \ldots, f_i(D)]$ defines which particle's *pbest*, the particle *i* should follow. The decision whether the i^{th} particle follows its own or other's $pbest_i{}^d$ for each dimension *d* is determined based on learning probability *Pc* values. Different *Pc* values are defined for different particles. The *Pc* value for each particle is calculated using the following equation [16]:

$$Pc_i = a + b * \frac{\left(\exp\left(\frac{10(i-1)}{ps-1}\right) - 1\right)}{(\exp(10) - 1)} \quad (5)$$

Where, '*ps*' represents the population size, $a = 0.05$, $b = 0.45$. For each dimension of a particle, a random number is generated and compared with *Pc* value. If random number is larger than *Pc* value, the particle will learn from the corresponding dimension of its own *pbest*. If random number is smaller than *Pc* value, the particle will learn from another particle's *pbest* for that dimension. The guidance particle is decided by employing a tournament selection, i.e. two particles are selected randomly and the particle with better fitness is chosen for the corresponding dimension. Therefore, the exemplar $pbest_{fi(d)}$ is a new position which can lead the particles to the new direction in the search space. In order not to waste function evaluations in a wrong direction, a certain number of evaluations is defined as refreshing gap *m* and the particle learns from the exemplar until the particle ceases improving. In this way, all the particles' historical best information is used to update a particle's flying direction. This enables the CLPSO to preserve the population diversity and discourage premature convergence. In [16], CLPSO algorithm was evaluated using different types of benchmark problems and showed significant improved performance, especially when solving multimodal problems.

Beside CLPSO velocity update rule, the following new CLPSO velocity update rule is also used in the proposed EPSO algorithm:

$$V_i^d = w * V_i^d + c_1 * rand1_i^d * \left(pbest_{fi(d)}^d - X_i^d\right) + c_2 * rand2_i^d * \left(gbest^d - X_i^d\right) \quad (6)$$

By combining the CL strategy with the whole swarm's best experience, the new *gbest*-guided CLPSO velocity update rule will encourage the particles to converge to the global optimum while maintaining the population diversity.

3.1.2. Fitness-distance-ratio based particle swarm optimization (FDR-PSO)

FDR-PSO was proposed to tackle the problem of premature convergence observed in PSO [17]. In comparison with PSO, FDR-PSO added a social learning component that learns from the neighboring particle's experience (*nbest*). The neighboring particles are selected with two criteria: (1) they must be near the particle being updated and (2) they must have better fitness than the particle being updated. Whether a neighboring particle satisfies these criteria or not is decided by the ratio of fitness distance to one-dimensional distance called Fitness-Distance-Ratio [17]. For maximization problem, the particle which can maximize Fitness-Distance-Ratio is selected as the neighboring particle. Let the neighboring particle *nbest* be the particle *j* which maximizes the following Fineness-Distance-Ratio:

$$\frac{Fitness\left(P_j\right) - Fitness\left(X_j\right)}{|P_j^d - X_j^d|} \quad (7)$$

where, $|\ldots|$ is the absolute value. For a minimization problem, the particle which can minimize Fitness-Distance-Ratio is selected as the neighbor particle and $Cost\left(P_j\right) - Cost\left(X_j\right)$ is used in the numerator of the above expression. After selecting the neighboring particle *nbest*, the velocity component in d^{th} dimension of the i^{th} particle is updated using the following equation:

$$V_i^d = w * V_i^d + c_1 * rand1_i^d * \left(pbest_i^d - X_i^d\right) + c_2 * rand2_i^d * \left(gbest^d - X_i^d\right) + c_3 * \left(nbest_j^d - X_i^d\right) \quad (8)$$

where, $c_1 = 1$, $c_2 = 1$ and $c_3 = 2$. *pbest* is the cognitive component, the particle's own experience and *gbest* is the social component, the best experience found so far by the whole population, *nbest* is the experience of neighboring particle which maximize/minimize Fitness-Distance-Ratio. FDR-PSO was tested on six benchmark functions and the experimental results showed that FDR-PSO performed significantly better than PSO, being capable of avoiding premature convergence and less likely to be stuck in a local optimum [17].

3.1.3. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients (HPSO-TVAC)

In order to efficiently control the local search and convergence to the global optimum solution, Ratnaweera et al. introduced self-organizing hierarchical PSO with time-varying acceleration coefficients (HPSO-TVAC) in which only cognitive and social parts are considered to estimate the new velocity of each particle and the particles are reinitialized whenever they are stagnated in the search space [18]. Authors claims that in the absence of the previous velocity term, the particles rapidly move toward a local optimum and stagnate due to lack of the momentum. Therefore, HPSO-TVAC is introduced to provide the required momentum for the particles to find the global optimum solution, in the absence of the previous velocity term in Eq. (1). In HPSO-TVAC algorithm, the velocity of a particle is updated as follows:

$$V_i^d = c_1 * rand1_i^d * \left(pbest_i^d - X_i^d\right) + c_2 * rand2_i^d * \left(gbest^d - X_i^d\right) \quad (9)$$

where, the previous velocity term from the Eq. (1) is kept at zero and the modulus of velocity vector of a particle is reinitialized with a random velocity, i.e. $V_i^d = rand * v(reinitialization\ velocity)$ should it stagnate ($V_i^d = 0$) in the search space. *rand* values are uniform distribution random numbers in the range of [0,1]. Therefore, in HPSO-TVAC, a series of particle swarm optimizers are automatically generated inside the main particle swarm optimizer according to the behavior of the particles in the search space [18]. The reinitialization velocity is set to be proportional to the maximum allowable velocity V_{max} in the algorithm. In [18], two unimodal and three multimodal functions were used to evaluate the performance of the HPSO-TVAC algorithm. Though HPSO-TVAC performs much better than PSO, the performance of the algorithm was found to be poor in solving multimodal problems [18].

3.1.4. Distance-based locally informed PSO (LIPS)

Instead of the swarm's best experience (*gbest*), LIPS uses *local bests*, the best experiences of the neighboring particles to guide the particles for finding the optimum over the search space [19]. The distances between the particle and other particles are measured by Euclidean distance. Then, the nearest neighboring particles are selected to guide the particle. Using the local information from its nearest neighbors, the particle's velocity is adjusted using the formula [46] as follows:

$$V_i^d = \chi * \left(V_i^d + \varphi\left(P_i^d - X_i^d\right)\right) \quad (10)$$

Where,

$$P_i = \frac{\sum_{j=1}^{nsize}\left(\varphi_j * nbest_j\right)/nsize}{\varphi} \quad (11)$$

Table 1
Parameter Settings.

Algorithm	Inertia weight w	Constriction Coefficients χ	Acceleration Coefficients c_1, c_2, c	Neighborhood size	Reference
Inertia weight PSO	0.9–0.2	–	$c_1 = 2$, $c_2 = 2$	–	[29]
LIPS	–	0.729	$c = 2$	3	[19]
FDR-PSO	0.9–0.2	0.729	$c_1 = 1$, $c_2 = 1$, $c_3 = 2$	–	[17]
CLPSO	0.9–0.2	–	$c = 3$–1.5	–	[16]
OLPSO	0.9–0.4	–	$c = 2$	–	[30]
HPSO-TVAC	–	–	$c_1 = 2.5$–0.5, $c_2 = 0.5$–2.5	–	[18]
sHPSO	0.72	–	$c_1 = 2.5$–0.5, $c_2 = 0.5$–2.5	–	[32]
CLPSO with gbest	0.9–0.2	–	$c_1 = 2.5$–0.5, $c_2 = 0.5$–2.5	–	–

Table 2
CEC2005 Test functions.

Test Functions	Initialization Range	Search Range	$F(x^*)$ *f_bias*	Fixed Accuracy Level
Unimodal Functions				
F_1: Shifted Sphere Function	$[-100,100]^D$	$[-100,100]^D$	−450	−450 + 1e-6
F_2: Shifted Schwefel's Problem 1.2	$[-100,100]^D$	$[-100,100]^D$	−450	−450 + 1e-6
F_3: Shifted Rotated High Conditioned Elliptic Function	$[-100,100]^D$	$[-100,100]^D$	−450	−450 + 1e-6
F_4: Shifted Schwefel's Problem 1.2 with Noise in Fitness	$[-100,100]^D$	$[-100,100]^D$	−450	−450 + 1e-6
F_5: Schwefel's Problem 2.6 with Global Optimum on Bounds	$[-100,100]^D$	$[-100,100]^D$	−310	−310 + 1e-6
Multimodal Functions				
F_6: Shifted Rosenbrock's Function	$[-100,100]^D$	$[-100,100]^D$	390	390 + 1e-2
F_7: Shifted Rotated Griewank's Function without Bounds	$[0,600]^D$	$[-600,600]^D$	−180	−180 + 1e-2
F_8: Shifted Rotated Ackley's Function with Global Optimum on Bounds	$[-32,32]^D$	$[-32,32]^D$	−140	−140 + 1e-2
F_9: Shifted Rastrigin's Function	$[-5,5]^D$	$[-5,5]^D$	−330	−330 + 1e-2
F_{10}: Shifted Rotated Rastrigin's Function	$[-5,5]^D$	$[-5,5]^D$	−330	−330 + 1e-2
F_{11}: Shifted Rotated Weierstrass Function	$[-0.5, 0.5]^D$	$[-0.5, 0.5]^D$	90	90 + 1e-2
F_{12}: Schwefel's Problem 2.13	$[-100,100]^D$	$[-100,100]^D$	−460	−460 + 1e-2
Expanded Functions				
F_{13}: Expanded Extended Griewank's plus Rosenbrock's Function(F8F2)	$[-3,1]^D$	$[-3,1]^D$	−130	−130 + 1e-2
F_{14}: Shifted Rotated Expanded Scaffer's F6	$[-100,100]^D$	$[-100,100]^D$	−300	−300 + 1e-2
Hybrid Composition Functions				
F_{15}: Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	120	120 + 1e-2
F_{16}: Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	120	120 + 1e-2
F_{17}: Rotated Hybrid Composition Function with Noise in Fitness	$[-5,5]^D$	$[-5,5]^D$	120	120 + 1e-2
F_{18}: Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	10	10+ 1e-1
F_{19}: Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum	$[-5,5]^D$	$[-5,5]^D$	10	10+ 1e-1
F_{20}: Rotated Hybrid Composition Function with the Global Optimum on the Bounds	$[-5,5]^D$	$[-5,5]^D$	10	10+ 1e-1
F_{21}: Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	360	360 + 1e-1
F_{22}: Rotated Hybrid Composition Function with High Condition Number Matrix	$[-5,5]^D$	$[-5,5]^D$	360	360 + 1e-1
F_{23}: Non-Continuous Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	360	360 + 1e-1
F_{24}: Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	260	260 + 1e-1
F_{25}: Rotated Hybrid Composition Function without Bounds	$[-2,5]^D$	$[-5,5]^D$	260	260 + 1e-1

x^*:global optimum, D = dimension

$$\varphi_j \sim U\left(0, \frac{4.1}{nsize}\right) \tag{12}$$

$$\varphi = \sum_{j=1}^{nsize} \varphi_j \tag{13}$$

where, $nbest_j$ is the j^{th} nearest neighborhood to i^{th} particle's *pbest*. *nsize* is the neighborhood size and it is dynamically increased from 2 to 5. χ is defined as constriction coefficient and $\chi = 0.7298$. φ_j is random positive number drawn from a uniform distribution in the range of (0, 4.1/*nize*). φ is defined as acceleration weight and summation of φ_j. The position is updated using the same equation as in (2). Euclidean distance based neighborhood selection is$O\left(N^2\right)$, where N is the number of particles in the current population. With the Euclidean distance based neighborhood selection, LIPS uses the information of the neighbors from the same region and this improves local search ability of the algorithm, especially at the latter stages of the search. This ability enables the particles to quickly converge to global optimum with high accuracy. Compared to other PSO variants, LIPS performed well on multimodal optimization [19].

3.2. Self-adaptive selection strategy

In the proposed EPSO algorithm, *inertia weight* PSO, LIPS, FDR-PSO, HPSO-TVAC and CLPSO-*gbest* variant are employed. The Eq. (3) is used to update the velocity in *inertia weight* PSO. The velocity of the CLPSO-*gbest* variant is updated using the Eq. (6). As mentioned above, the Eqs. (8)–(10) are employed for FDR-PSO, HPSO-TVAC and LIPS respectively. The PSO strategies are selected in a self-adaptive manner [18] based on their previous success and failure in finding good solutions. In the self-adaptive selection strategy, a fixed

Table 3
Comparison of experimental results of PSO algorithms for 10 dimensional CEC2005 test functions.

Functions	Criteria	EPSO	PSO	LIPS	CLPSO	HPSO-TVAC	FDR-PSO	sHPSO
F_1	mean	**0**	0	**0**	**0**	0	0	0
	std	0	3.20E-14	0	0	1.17E-13	2.88E-14	7.22E+02
	rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	h		+	=	=	+	+	+
F_2	mean	**0**	0	6.46E-07	1.86E-02	0	0	0
	std	6.46E-14	1.07E-13	3.54E-06	2.24E-02	7.52E-14	4.72E-13	2.69E+02
	rank	**1**	**1**	6	7	**1**	**1**	**1**
	h		+	+	+	**+**	**=**	+
F_3	mean	7.00E+04	1.60E+05	7.45E+05	3.49E+05	6.36E±04	**3.60E+04**	8.82E+04
	std	4.35E+04	1.10E+05	2.56E+06	2.21E+05	2.58E+04	3.81E+04	6.48E+04
	rank	3	5	7	6	2	**1**	4
	h		+	+	+	**=**	–	=
F_4	mean	1.97E-08	**2.08E-13**	1.16E+03	5.20	2.15E+02	3.74E-10	2.31E+03
	std	9.90E-08	1.19E-13	8.06E+02	11.02	1.97E-09	2.96E+02	2.23E+03
	rank	3	**1**	6	4	5	2	7
	h		–	+	+	+	–	+
F_5	mean	0.01	1.04E+03	2.60E+03	6.95	4.96E+02	**1.19E-09**	1.27E+03
	std	0.02	2.07E+02	8.97E+02	16.01	2.73E-09	3.61E+02	2.10E+03
	rank	2	5	7	3	4	**1**	6
	h		+	**+**	**+**	+	–	+
F_6	mean	1.43	3.47	13.58	0.86	11.83	**0.54**	786.51
	std	2.92	7.76	31.47	1.94	1.34	22.29	4017.41
	rank	3	4	6	2	5	**1**	7
	h		+	+	=	+	–	+
F_7	mean	**0.15**	0.22	3.67	0.21	2.56	0.20	11.58
	std	0.08	0.14	3.76	0.11	0.10	1.84	15.61
	rank	**1**	4	6	3	5	2	7
	h		+	+	=	+	**=**	+
F_8	mean	20.34	20.29	20.32	20.27	20.17	20.30	**20.14**
	std	0.06	0.09	0.12	0.05	0.07	0.11	0.12
	rank	7	4	6	3	2	5	**1**
	h		–	=	–	–	–	–
F_9	mean	0.03	3.55	10.58	**0**	1.76	3.12	18.67
	std	0.18	1.40	4.30	0	1.824039	1.47	15.16
	rank	2	5	6	**1**	3	4	7
	h		+	+	–	+	+	+
F_{10}	mean	**8.19**	14.36	10.31	9.24	34.76	10.68	34.22
	std	3.03	5.21	5.61	2.97	3.99	11.37	12.14
	rank	**1**	5	3	2	7	4	6
	h		+	**=**	+	+	+	+
F_{11}	mean	3.21	3.36	3.97	4.53	5.91	**2.27**	6.92
	std	1.01	1.42	1.09	0.78	1.05	1.19	1.69
	rank	2	3	4	5	6	**1**	7
	h		=	+	+	+	–	+
F_{12}	mean	**3.98E+01**	1.83E+03	2.73E+03	7.22E+01	5.70E+02	1.56E+03	1.43E+03
	std	8.38E+01	4.31E+03	5.03E+03	5.20E+01	2.79E+03	7.52E+02	3.21E+03
	rank	**1**	6	7	2	3	5	4
	h		+	+	+	+	+	+
F_{13}	mean	0.47	0.63	0.82	**0.28**	0.66	0.65	1.61
	std	0.12	0.21	0.42	0.06	0.16	0.22	1.00
	rank	2	3	6	**1**	5	4	7
	h		+	–	–	+	+	+
F_{14}	mean	**2.44**	2.86	3.28	3.00	3.20	2.81	3.60
	std	0.49	0.41	0.49	0.27	0.65	0.35	0.41
	rank	**1**	3	6	4	5	2	7
	h		+	+	+	+	+	+
F_{15}	mean	9.40E±00	2.34E+02	2.17E+02	**4.81E+00**	2.15E+02	2.57E+02	3.82E+02
	std	2.34E+01	1.76E+02	1.05E+02	1.46E+01	2.36E+02	2.04E+02	1.93E+02
	rank	2	5	4	**1**	3	6	7
	h		+	+	=	+	+	+
F_{16}	mean	**1.10E+02**	1.29E+02	1.21E+02	1.20E+02	1.91E+02	1.33E+02	1.87E+02
	std	9.82E+00	1.64E+01	1.89E+01	9.92E+00	5.63E+01	3.91E+01	4.58E+01
	rank	**1**	4	3	2	7	5	6
	h		+	+	+	+	+	+
F_{17}	mean	**1.12E+02**	1.42E+02	1.38E+02	1.25E+02	1.95E+02	1.48E+02	2.05E+02

Table 3 (*Continued*)

Functions	Criteria	EPSO	PSO	LIPS	CLPSO	HPSO-TVAC	FDR-PSO	sHPSO
	std	1.00E+01	6.81E+01	4.21E+01	1.29E+01	9.13E+01	4.47E+01	7.31E+01
	rank	**1**	4	3	2	6	5	7
	h		+	+	+	+	+	+
F_{18}	mean	**6.18E+02**	7.10E+02	8.43E+02	6.85E+02	1.01E+03	8.79E+02	9.79E+02
	std	2.03E+02	2.53E+02	2.10E+02	1.87E+02	1.25E+02	8.67E+01	7.28E+01
	rank	**1**	3	4	2	7	5	6
	h		=	+	=	+	+	+
F_{19}	mean	6.45E±02	6.99E+02	8.76E+02	**6.54E+02**	9.95E+02	8.01E+02	9.89E+02
	std	2.11E+02	2.45E+02	1.75E+02	1.88E+02	2.20E+02	9.26E+01	1.42E+02
	rank	**1**	3	5	2	7	4	6
	h		=	+	=	+	+	+
F_{20}	mean	6.66E±02	**6.16E+02**	8.83E+02	7.46E+02	9.86E+02	8.35E+02	1.01E+03
	std	2.02E+02	2.66E+02	1.53E+02	1.62E+02	1.84E+02	1.12E+02	8.42E+01
	rank	2	**1**	5	3	6	4	7
	h		**=**	+	=	+	+	+
F_{21}	mean	**3.87E+02**	4.51E+02	7.37E+02	4.52E+02	1.04E+03	6.95E+02	1.09E+03
	std	1.22E+02	1.93E+02	2.76E+02	1.40E+02	3.70E+02	2.83E+02	2.23E+02
	rank	**1**	2	5	3	6	4	7
	h		=	+	+	+	+	+
F_{22}	mean	7.59E±02	7.64E+02	8.26E+02	**7.27E+02**	8.86E+02	7.82E+02	8.86E+02
	std	1.36E+01	9.30E+01	6.53E+01	1.42E+02	3.76E+01	1.30E+02	1.06E+02
	rank	2	3	5	**1**	6	4	6
	h		+	+	=	+	+	+
F_{23}	mean	5.91E±02	6.77E+02	9.64E+02	**5.58E+02**	1.12E+03	8.98E+02	1.17E+03
	std	9.55E+01	1.43E+02	2.51E+02	7.16E+01	2.36E+02	2.63E+02	1.69E+02
	rank	2	3	5	**1**	6	4	7
	h		+	+	=	+	+	+
F_{24}	mean	**2.00E+02**	2.80E+02	3.34E+02	2.00E+02	8.23E+02	4.33E+02	6.62E+02
	std	8.09E-13	1.56E+02	2.00E+02	1.63E-08	3.00E+02	4.59E+02	4.56E+02
	rank	**1**	3	4	**1**	7	5	6
	h		+	+	=	+	+	+
F_{25}	mean	**2.00E+02**	3.10E+02	4.36E+02	2.00E+02	7.87E+02	2.90E+02	7.38E+02
	std	8.15E-13	1.75E+02	3.72E+02	2.59E-09	1.95E+02	4.37E+02	4.85E+02
	rank	**1**	4	5	**1**	7	3	6
	h		+	+	+	+	**=**	+
best/2nd		13/8/1	4/2/0	1/0/3	8/6/1	2/2/6	6/3/0	3/0/12
best/worst ranking								
+/=/−			19/4/2	22/3/0	9/11/3	23/1/1	16/3/6	23/1/1
Algorithms		EPSO	PSO	LIPS	CLPSO	HPSO-TVAC	FDR-PSO	sHPSO

number of generations are defined as learning period (LP) and EPSO keeps track of the success and failure of each PSO strategy during the leaning period. Based on the recorded success and failure memories, the success rate of each PSO strategy is calculated and the selection probability of each PSO strategy is updated at each subsequent generation after LP generation. Then, the PSO strategies are selected proportional to their success rate for each particle in the current iteration. The detailed procedure of PSO self-adaptive selection strategy in EPSO is presented step by step as follows:

Let p_k be the probability of each PSO strategy and initialize each p_k as $1/K$ so that they have equal probability to be chosen, where $k = 1, 2, \ldots, K$ and K = total number of PSO strategies from the pool.

Step I: Stochastic universal selection method [21] is used to select the candidate PSO strategy for the particles.

Step II: For generation G, if the selected PSO strategy can generate improved solutions, the selected PSO strategy is promoted by recording in success memories $ns_{k,G}$. Otherwise, the PSO strategy is demoted by recording as failure memories $nf_{k,G}$.

Step II: The success and failure memories are updated for a fixed number of generations called the learning period (LP). If the memories overflow after LP generations, the earliest record stored in the memories will be removed for the numbers obtained in the current generation to be stored in the memory.

Step IV: At subsequent generation after LP generations, the probability of selecting PSO strategies from the pool will be updated as follows:

$$p_{k,G} = \frac{S_{k,G}}{\sum_{K=1}^{K} S_{k,G}} \tag{14}$$

where,

$$S_{k,G} = \frac{\sum_{g=G-LP}^{G-1} ns_{k,G}}{\sum_{g=G-LP}^{G-1} ns_{k,g} + \sum_{g=G-LP}^{G-1} nf_{k,g}} + \varepsilon \tag{15}$$

where, $k = 1, 2, \ldots, K$, $G > LP$ and $S_{k,G}$ represents the success rate of the solution generated by the k^{th} PSO strategy. $\varepsilon = 0.01$ is used to avoid the possible null success rate. The PSO strategy with the higher success rate will have the larger probability to be selected to guide the particle at the current generation. In other words, the suitable PSO strategy for a given problem is gradually selected by self-adaptive selection strategy.

To study the success rate of each PSO strategy on different optimization problems, the proposed ensemble PSO (EPSO) algorithm is experimented with unimodal and multimodal problems from CEC 20005 benchmark functions [2]. The search behavior profiles of the proposed EPSO algorithm on different benchmark problems

are presented in the next section to indicate when and where the different PSO strategies in the ensemble PSO (EPSO) algorithm kicks in to gain insights into the dynamics of the optimization process. The comparative study of the proposed EPSO and its individual PSO variants, as well as recent state-of-the-art PSO algorithms, is conducted for further performance investigation. The algorithm computational complexity has been evaluated in accordance with CEC2005 benchmark competition [2] and the performance of EPSO is also investigated based on the computational cost. The performance investigation results are discussed in Section 4.

4. Performance evaluation

4.1. Parameter settings

In this paper, the performance of the proposed ensemble PSO algorithm is evaluated using the shifted and rotated CEC2005 benchmark functions [2]. The set of CEC2005 benchmark functions consist of 25 different types of unimodal, multimodal, expanded and hybrid composition functions. The search range, initialization range, bias values and fixed accuracy values of all benchmark functions are described in Table 2. If the result is obtained within fixed accuracy values, it is defined as the run is successful. The experiments are conducted using Matlab and performed on a PC with an Intel Core i7-2600 @ 3.40 GHz CPU and 8 GB RAM. The performance of the proposed ensemble PSO algorithm is evaluated with its individual PSO variants and the other two state-of-the-art PSO algorithms.

Inertia weight PSO (PSO) [29]

- Comprehensive Learning PSO (CLPSO) [16]
- Self-organizing hierarchical PSO with time varying acceleration coefficients (HPSO-TVAC) [18]
- Fitness-Distance-Ratio based PSO (FDR-PSO) [17]
- Distance-based locally informed PSO (LIPS) [19]
- Orthogonal Learning PSO (OLPSO) [30]
- Static Heterogeneous Swarm Optimization (sHPSO) [32]

The first five PSO algorithms are PSO strategies employed in the proposed EPSO algorithm.[1] The next is recent start-of-art OLPSO algorithm. In OLPSO, orthogonal learning strategy is used to generate the exemplar from the particle's previous best experience and the swarm's best experience [30]. The last is a static heterogeneous PSO (HPSO) with characteristics similar to the proposed ensemble PSO (EPSO) algorithm. In the sHPSO algorithm, PSO [23], cognitive-only velocity update [47], social-only velocity update [47], barebones PSO [48] were used and randomly assigned to the particles. Unlike the proposed EPSO algorithm, the behaviors are kept unchanged throughout the search process. The parameter settings of each PSO algorithm are listed in Table 1. The same parameter setting is used for the PSO variants which are included in the proposed EPSO algorithm.

4.2. Experimental results

In this section, the proposed EPSO algorithm is compared with its individual PSO variants as well as static HPSO (sHPSO) and recent OLPSO variant. The performance is evaluated in terms of error mean and standard deviation values of their solutions. The best results obtained are highlighted in bold and the second-best results are highlighted by underlying in the performance comparison tables. Non-parametric Wilcoxon signed-rank test [49] is conducted between the results of proposed EPSO 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 EPSO performs significantly better than the compared algorithm, the symbol "=" represents that there is no significant difference between EPSO and the compared algorithm and the symbol "-" represents that the compared algorithm performs significantly better than EPSO algorithm. The algorithms are ranked based on their average mean error values. The number of (best/2nd best/worst ranking) and the number of +/=/− are counted for each algorithm and presented in the last two rows of Table 3. In addition, the convergence progress of all algorithms is also analyzed for unimodal, multimodal, expanded and hybrid composition functions. The median performance of 30 runs is used to analyze the convergence performance. The convergence characteristics graphs of EPSO and compared PSO algorithms are presented in Figs. 3 and 4.

The small/large subpopulation group sizes of EPSO algorithm are tuned using the first 14 CEC 2005 30-dimensional test functions. The results are ranked based on the mean error values. Based on the results, the parameter setting ($g_1 = 15$, $g_2 = 25$) provides the satisfactory and consistent performance on the 14 problems. Based on these experimental results, the small/large subpopulation group sizes (8/12 for population size 20 and 15/25 for population size 40) were determined for the 10 and 30 dimensional problems, respectively.

The experiment of 10 dimensional problems is conducted using the number of function evaluations (FES) 100000 and population size 20. As mentioned above, group sizes 8/12 are used for small/large subpopulation group sizes. The experimental results are illustrated in Table 3. For the unimodal problems, all algorithms perform well on the simple unimodal functions F_1 and F_2, except LIPS, CLPSO and sHPSO algorithms. For the remaining three umimodal problems, the FDR-PSO algorithm yields the best results on functions F_3 and F_5 and *inertia weight* PSO performs the best on function F_4. The proposed EPSO algorithm consistently performs well and ranked within top 3 for the unimdoal problems. For the multimodal problems (F_6–F_{12}), the proposed EPSO algorithm yields the best results on functions F_7, F_{10} and F_{12} and the second-best results on functions F_6, F_9 and F_{11}. The sHPSO algorithm performs the best on function F_8. For the two expanded functions, EPSO offers the best result on the function F_{14} while CLPSO performs the best on function F_{13}. For hybrid composition functions (F_{15}–F_{25}), EPSO algorithm obtains the best results on functions F_{16}–F_{18}, F_{21}, F_{24}, F_{25} and surpasses its individual PSO strategies as well as sHPSO algorithm. For the remaining hybrid composition functions, CLPSO and LIPS offers the best performance, followed by EPSO algorithm.

In *inertia weight* PSO and FDR-PSO algorithms, the particles are guided by the whole swarm best experience, *gbest*. Besides, using the linearly decreasing inertia weight to adjust the balance between the global and local search, the algorithms promote the global search at the early stage of the search and dramatically changes to the local search at the end of the search. Therefore, *inertia weight* PSO and FDR-PSO algorithms perform well on unimodal problems. However, *inertia weight* PSO performs poorly on multimodal problems. FDR-PSO algorithm continues its good performance on multimodal problems as the local search ability in FDR-PSO is enhanced by the guidance of the neighborhood particles. In CLPSO, the particles are guided by different particles for different dimensions. Therefore, CLPSO fails to perform well on unimdoal problems. However, CLPSO performs well on multimodal problems. HPSO-TVAC performs moderately on unimodal as well as multimodal problems. Compared to other PSO variants, LIPS and sHPSO algorithms perform poorly on both unimodal and multimodal problems. Due to the ensemble approach, EPSO per-

[1] Matlab source codes of the EPSO will be released online.

Table 4
Comparison of experimental results of PSO algorithms for 30 dimensional CEC2005 test functions.

Functions	Criteria	EPSO	PSO	LIPS	CLPSO	HPSO-TVAC	FDR-PSO	sHPSO	OLPSO
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**
	h		=	=	=	=	=	=	=
F_2	mean	**0**	0.37	83.52	1.14E+03	3.79E-06	0	1.44E-02	13.79
	std	1.88E-12	3.21E-01	326.6323	2.53E+02	2.82E-06	1.01E-11	7.10E-02	8.33
	rank	**1**	5	7	8	3	1	4	6
	h		+	+	+	+	+	+	+
F_3	mean	**3.62E+05**	6.53E+06	1.00E+07	1.22E+07	7.72E+05	4.14E+05	8.75E+05	1.60E+07
	std	1.35E+05	4.17E+06	1.35E+07	3.34E+06	2.96E+05	1.65E+05	5.34E+05	7.04E+06
	rank	**1**	5	6	7	3	2	4	8
	h		+	+	+	+	+	+	+
F_4	mean	**2.38E+02**	3.81E+02	3.32E+04	8.77E+03	2.48E+04	4.20E+02	2.02E+04	2.18E+03
	std	1.51E+02	3.31E+02	8.20E+03	1852.49	5.71E+03	2.47E+02	9.94E+03	1.09E+03
	rank	**1**	2	8	5	7	3	6	4
	h		+	+	+	+	+	+	+
F_5	mean	4.63E+03	3.85E+03	9.35E+03	4.47E+03	9.20E+03	**3.20E+03**	6.94E+03	3.30E+03
	std	8.08E+02	8.00E+02	1.75E+03	426.30	1.81E+03	6.20E+02	1.43E+03	3.75E+02
	rank	5	3	8	4	7	**1**	6	2
	h		+	+	=	+	–	+	–
F_6	mean	3.36	70.16	106.33	**2.39**	50.41	3.55	115.27	20.68
	std	4.15	95.09	205.89	3.84	50.54	7.04	228.56	24.97
	rank	2	6	7	**1**	5	3	8	4
	h		+	+	=	+	=	+	+
F_7	mean	0.02	0.76	85.38	0.70	**0.01**	0.02	0.04	**0.01**
	std	0.01	1.41	45.37	0.15	0.01	0.01	0.04	0.01
	rank	3	7	8	6	**1**	3	5	**1**
	h		+	+	+	–	=	=	=
F_8	mean	20.91	20.90	20.92	20.92	20.71	20.88	**20.18**	20.96
	std	0.08	0.07	0.12	0.06	0.15	0.07	0.19	0.08
	rank	5	4	6	6	2	3	**1**	8
	h		=	=	=	–	=	–	+
F_9	mean	**0**	19.00	49.49	**0**	10.71	30.25	82.54	**0**
	std	0	5.37	12.40	0	4.96	8.63	24.35	0
	rank	**1**	5	7	1	4	6	8	**1**
	h		+	+	=	+	+	+	=
F_{10}	mean	**54.56**	100.32	67.16	99.78	275.73	55.63	242.83	109.23
	std	16.98	58.63	19.23	12.48	45.49	18.68	89.17	18.94
	rank	**1**	5	3	4	8	2	7	6
	h		+	+	+	+	=	+	+
F_{11}	mean	22.23	23.18	21.08	24.42	30.62	**16.92**	32.29	25.05
	std	2.58	2.81	2.68	1.78	2.58	3.72	3.84	3.14
	rank	3	4	2	5	7	**1**	8	6
	h		=	=	+	+	–	+	+
F_{12}	mean	**4.10E+03**	5.07E+04	8.01E+04	1.47E+04	4.47E+03	9.52E+03	2.43E+04	1.22E+04
	std	4.19E+03	3.50E+04	7.02E+04	3.45E+03	4.66E+03	1.53E+04	2.66E+04	5.42E+03
	rank	**1**	7	8	5	2	3	6	4
	h		+	+	+	=	–	+	+
F_{13}	mean	**1.80**	3.02	4.08	1.86	3.85	2.97	6.14	1.86
	std	0.43	0.69	1.20	0.18	1.11	0.64	2.31	0.28
	rank	**1**	5	7	2	6	4	8	2
	h		+	+	=	+	+	+	=
F_{14}	mean	11.86	12.65	12.35	12.64	12.31	**11.66**	13.07	13.13
	std	0.77	0.43	0.50	0.22	0.37	0.58	0.39	0.20
	rank	2	6	4	5	3	**1**	7	8
	h		+	+	+	+	=	+	+
F_{15}	mean	56.00	299.71	379.36	**40.10**	321.29	306.28	457.16	244.88
	std	62.90	107.62	88.12	37.90	107.12	150.98	138.94	84.09
	rank	2	4	7	**1**	6	5	8	3
	h		+	+	=	+	+	+	+
F_{16}	mean	**104.18**	235.84	164.23	163.61	336.21	293.38	376.70	135.54
	std	34.19	152.98	93.55	27.87	97.20	182.92	99.46	55.48
	rank	**1**	5	4	3	7	6	8	2
	h		+	+	+	+	+	+	+
F_{17}	mean	**106.28**	262.63	288.11	219.31	391.68	280.55	367.25	204.56
	std	46.50	153.27	92.59	40.09	97.34	192.62	133.86	46.67

Table 4 (*Continued*)

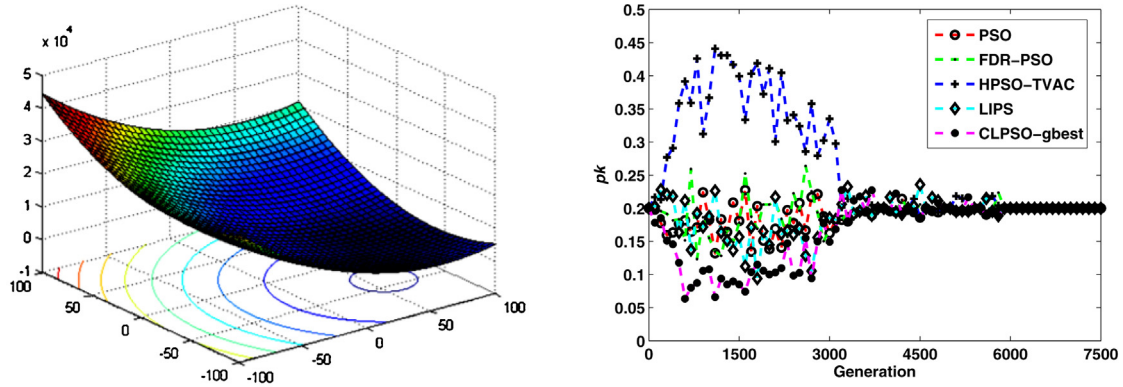
Functions	Criteria	EPSO	PSO	LIPS	CLPSO	HPSO-TVAC	FDR-PSO	sHPSO	OLPSO
	rank	**1**	4	6	3	8	5	7	2
	h		+	+	+	+	+	+	+
F_{18}	mean	**905.74**	927.77	969.31	907.76	908.24	918.16	997.99	906.63
	std	36.00	2.86	16.05	26.75	79.00	3.47	60.04	20.23
	rank	**1**	6	7	3	4	5	8	2
	h		+	+	=	=	=	+	–
F_{19}	mean	**901.87**	926.69	971.95	915.09	912.45	916.00	982.94	912.45
	std	40.82	2.49	16.13	1.36	79.78	4.72	78.69	79.78
	rank	**1**	6	7	4	2	5	8	2
	h		+	+	=	=	=	+	–
F_{20}	mean	**890.17**	927.06	973.73	911.24	905.92	913.53	974.13	905.92
	std	50.69	2.35	14.53	20.57	74.72	21.79	67.96	74.72
	rank	**1**	6	7	4	2	5	8	2
	h		+	+	=	=	=	+	=
F_{21}	mean	**496.98**	510.00	520.00	500.00	1173.50	539.42	996.01	500.00
	std	16.55	54.77	76.11	0.00	229.74	141.69	328.97	0
	rank	**1**	4	5	2	8	6	7	2
	h		=	+	=	+	=	+	+
F_{22}	mean	**923.34**	938.66	1009.91	974.32	1216.89	927.56	1123.41	941.70
	std	21.97	12.96	29.64	17.22	73.68	17.45	103.68	16.30
	rank	**1**	3	6	5	8	2	7	4
	h		+	+	+	+	=	+	+
F_{23}	mean	**534.16**	574.17	647.24	534.16	1258.30	559.82	996.79	**534.16**
	std	3.57E-04	122.10	155.06	2.33E-04	34.57	100.45	323.48	5.04E-04
	rank	**1**	5	6	1	8	4	7	1
	h		+	+	=	+	+	+	=
F_{24}	mean	**200.00**	**200.00**	357.67	**200.00**	1276.11	210.00	995.63	**200.00**
	std	0	0	323.80	0	205.10	54.77	409.92	0
	rank	**1**	**1**	6	**1**	8	5	7	**1**
	h		=	+	=	+	+	+	=
F_{25}	mean	**200.00**	1113.45	411.68	200.00	1331.03	200.00	1037.43	207.46
	std	8.18E-13	1.44E+01	3.73E+02	5.07E-11	2.60E+01	1.21E-12	423.59	7.09
	rank	**1**	7	5	**1**	8	**1**	6	4
	h		+	+	+	+	+	+	+
best/2nd best/worst ranking		18/3/0	1/1/0	1/0/4	7/2/1	2/3/7	6/3/0	2/0/9	5/7/3
+/=/-			20/5/0	22/3/0	12/13/0	18/5/2	12/11/2	22/2/1	15/7/3
Algorithms		EPSO	PSO	LIPS	CLPSO	HPSO-TVAC	FDR-PSO	sHPSO	OLPSO

forms consistently well on unimodal and multimodal problems and performs especially well on hybrid composition problems. In these hybrid problems, EPSO obtains outstandingly superior performance over other PSO algorithms. Overall, EPSO achieves the best performance on 13 functions and the second best on 8 functions out of 25 10D problems. In addition, EPSO performs significantly better than its individual PSO variants on all four types of 10D problems.

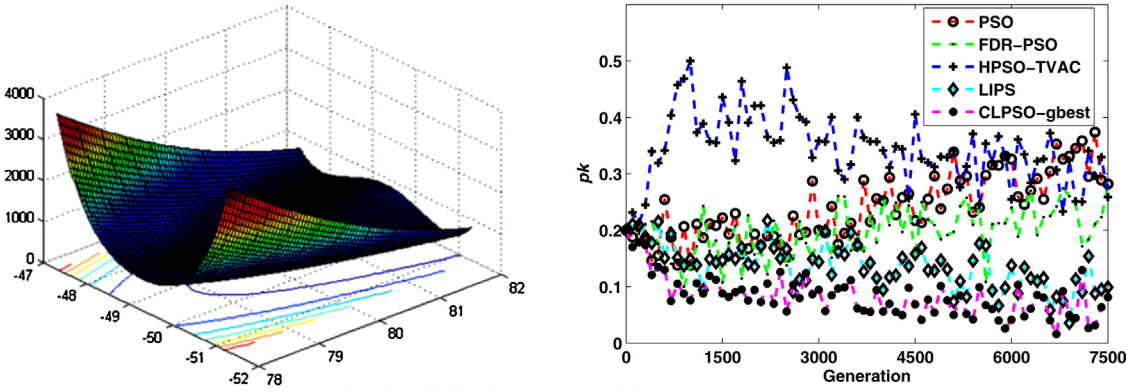
For 30 dimensional problems, the experiment is conducted using the number of FES 300000 and population size 40 and the results are illustrated in Table 4 together with the Wilcoxon signed rank test results. In EPSO, the group sizes 15/25 are used for the small/large subpopulation group sizes. For the five unimodal problems, EPSO gives the best results on the first 4 functions (F_1–F_4) and FDR-PSO yields the best results on function F_5. On seven multimodal functions, EPSO performs the best on functions F_9, F_{10} and F_{12}. For the remaining four multimodal problems, EPSO follows CLPSO on function F_6, OLPSO on function F_7, sHPSO on function F_8 and FDR-PSO on function F_{11} as the second best. EPSO algorithm yields the best on all expanded functions and FDR-PSO obtains the similar results on function F_{14}. For the hybrid composition problems, EPSO performs the best on 10 out of 11 hybrid composition functions.

From the experimental results, we can observe that *inertia weight* PSO fails to maintain its good performance on high dimensional problems and FDR-PSO algorithm robustly performs well on 30 dimensional unimodal and multimodal problems. CLPSO and OLPSO algorithm perform well on high dimensional multimodal problems as well as hybrid composition problems. However, the proposed EPSO algorithm performs the best on most of unimodal, multimodal and hybrid composition functions and ranked within the top 3, except for functions F_5 and F_8. The results are also statistically verified by non-parametric Wilcoxon signed rank test. Wilcoxon signed rank test results (+/=/–) are also presented in Table 4 and showed that overall, EPSO performs significantly better than other PSO algorithms. Besides, it can be observed from the number of (best/2nd best/worst ranking) shown in the last row of Tables 3 and 4, EPSO performs better on the high dimensional problems and achieved either the best or the second-best results on most of the test problems.

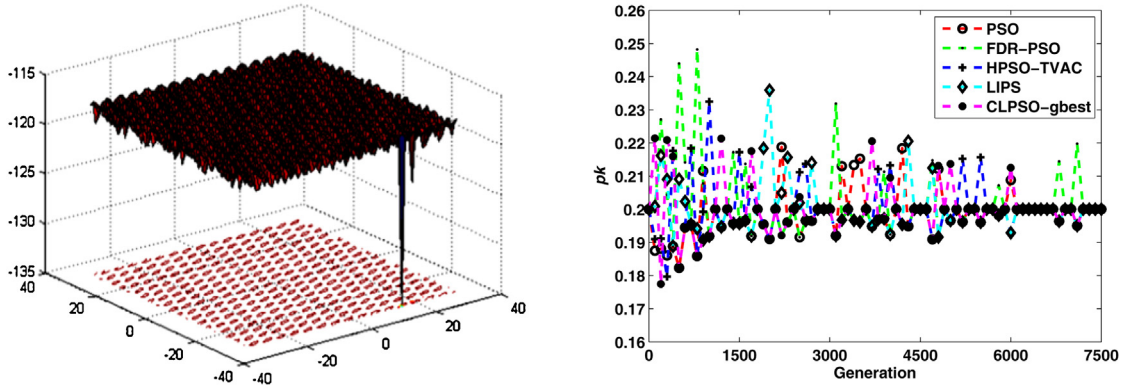
In addition, the outstanding performance of the proposed EPSO algorithm can also be seen in the convergence characteristic graphs. As shown in Figs. 3 and 4, the proposed EPSO algorithm performs either the best or second best on unimodal and multimodal problems and it obtains better convergence performance than other PSO algorithms on hybrid composition problems. Therefore, the



(a) F_1: Shifted Sphere Function



(b) F_6: Shifted Rosenbrock's Function



(c) F_8: Shifted Rotated Ackley's Function with Global Optimum on Bounds

Fig. 1. Search behavior plots for CEC2005 benchmark functions: (a) unimodal function F_1 and two multimodal functions: (b) F_6 and (c) F_8.

Table 5
Computational complexity of EPSO algorithm.

	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2 - T_1)/T_0$
10D	0.1431	320.4231	290.3601	−210.0836
30D	0.1431	320.4231	294.8366	−178.8013

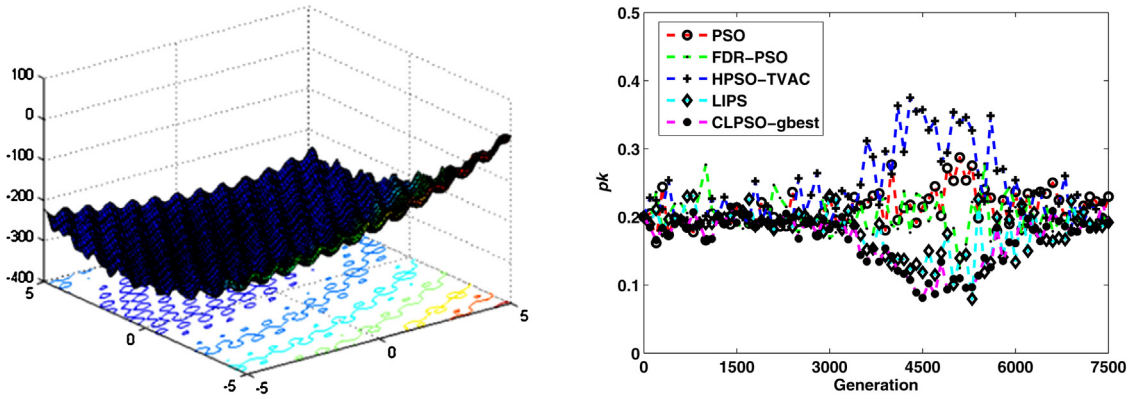
Table 6
Comparison of Computational Complexity of EPSO And IT's PSO variants on 30 D.

30D	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2 - T_1)/T_0$
EPSO	0.1431	320.4231	294.8366	−178.8013
PSO	0.1431	320.4231	319.0709	−9.4491
LIPS	0.1431	320.4231	329.6596	64.5460
FDR-PSO	0.1431	320.4231	378.1112	403.1316
CLPSO	0.1431	320.4231	361.4847	286.9436
HPSO-TVAC	0.1431	320.4231	293.6760	−186.9117

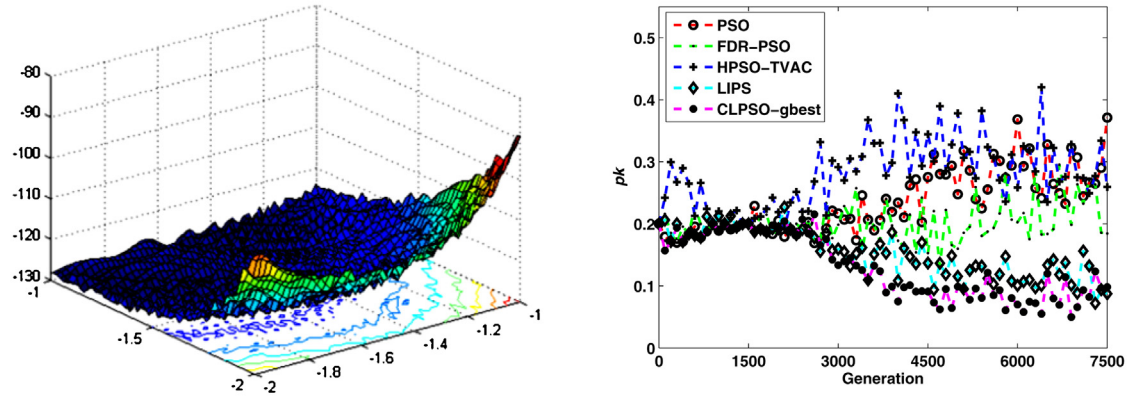
proposed EPSO performs well consistently and obtained superior performance regardless of dimensionality and problem type.

4.3. Algorithmic computational complexity

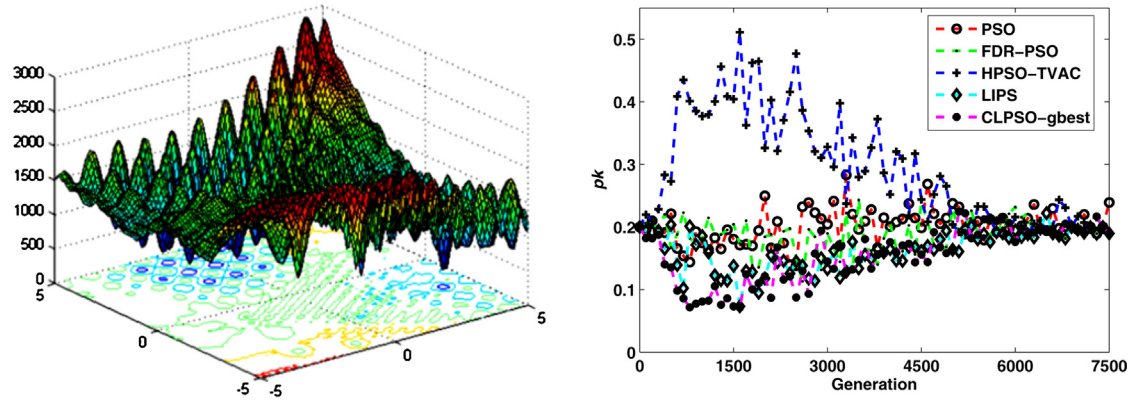
The computational complexity of the proposed EPSO algorithm is determined as defined in the CEC2005 benchmark competition [2] and presented in this section. In addition, the algorithm complexity of its individual PSO variants is also calculated and the performance of the proposed EPSO algorithm is evaluated based on the computational cost. The algorithm complexities are calculated on 10 and 30 dimensions. All the experiments are conducted on the following system:



(a) F_{10}: Shifted Rotated Rastrigin's Function



(b) F_{13}:Shifted Expanded Griewank's plus Rosenbrock's Function (F_8F_2)



(c) F_{15}: Hybrid Composition Function

Fig. 2. Search behavior plots for CEC2005 benchmark functions: (a) multimodal function F_{10}, (b) shifted expanded function F_{13} and (c) hybrid composition function F_{15}.

- OS: Windows 7
- CPU: Intel Core i7-2600 @ 3.40 GHz CPU
- RAM: 8GB
- Language: Matlab 2014a

Table 5 shows the computation complexity of EPSO algorithm on 10 and 30 dimensions. In Table 5, T_0 is the time computed by running the following codes from:

for $i = 1{:}1000000$

$x = (\text{double})\ 5.55;\ x = x + x;\ x = x./2;\ x = x^{*}x;$

$x = \text{sqrt}(x);\ x = \log(x);\ \mathbf{x} = \exp(x);\ y = x/x;$

end

T_1 is the time to execute 200,000 evaluations of benchmark function F_3 by itself with D dimensions. T_2 is the execution time of EPSO algorithm on the same function F_3 with the same number of function evaluation 200,000 in D dimensions. The time ($\hat{T}_2$) is the mean values of T_2 over five runs. As seen in Table 5, both T_1, $\hat{T}_2$ and $(\hat{T}_2 - T_1)/T_0$ scaled linearly with the number of dimensions. The computation complexity comparisons of the EPSO algorithm and its individual PSO variants are shown in Table 6 for 30D. The same procedure and the same system are used to compute the computation complexity of the individual PSO variants of the proposed EPSO algorithm. It can be observed from Table 6 that the computational cost of the EPSO algorithm is lower than other PSO algorithms, except for the HPSO-TVAC algorithm.

However, as shown in the performance comparison, Tables 3 and 4, the performance of EPSO is significantly bet-

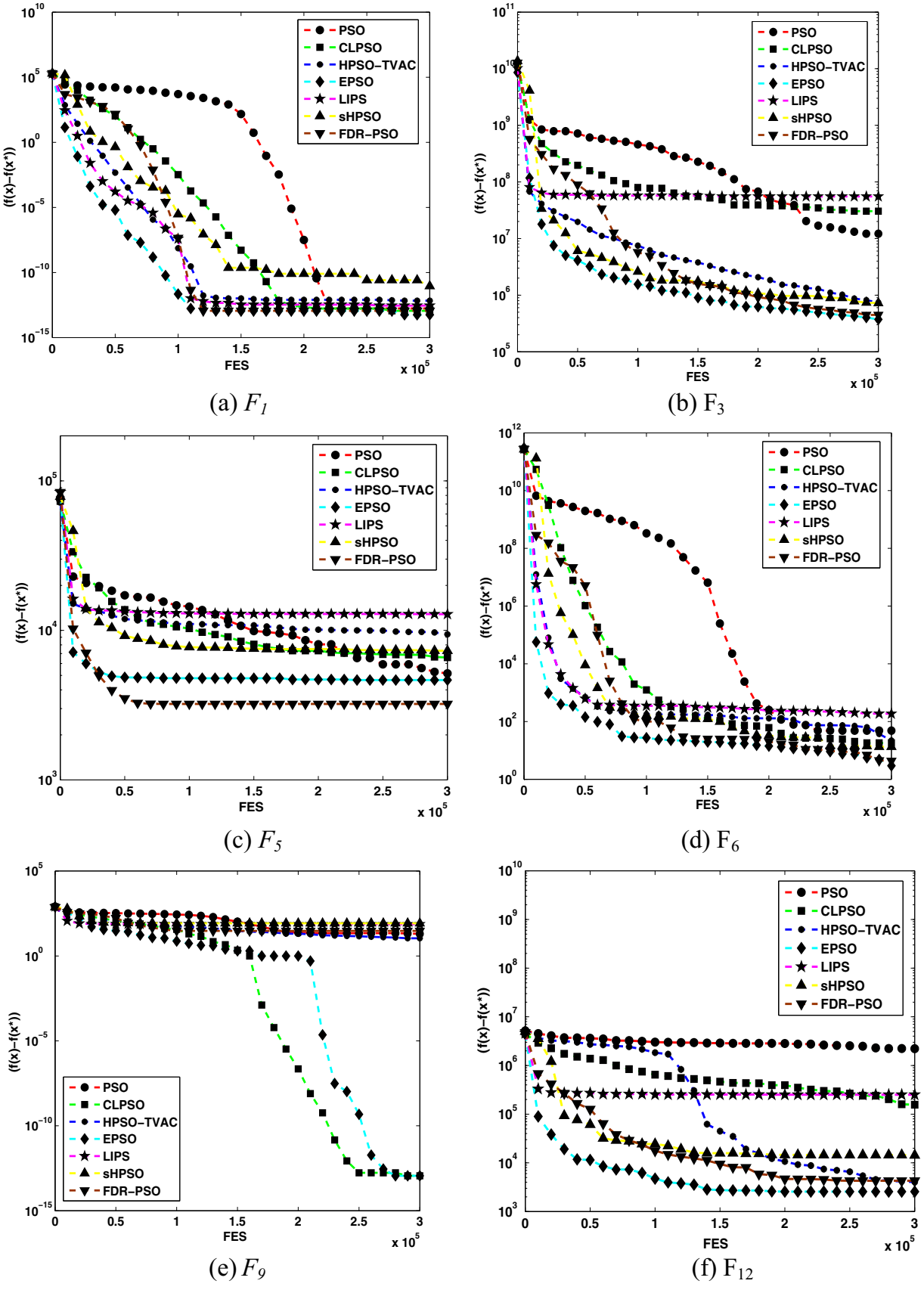


Fig. 3. Median convergence characteristics graphs of 30 dimensional CEC2005 benchmark functions: (a) F_1: Unimodal Shifted Sphere Function, (b) F_3: Unimodal Shifted Rotated High Conditioned Elliptic Function, (c) F_5: Unimodal Schwefel's Problem 2.6, (d) F_9: Multimodal Shifted Rosenbrock's Function (e) F_9: Multimodal Shifted Rastrigin's Function (f) F_{12}: Multimodal Schwefel's Problem 2.13.

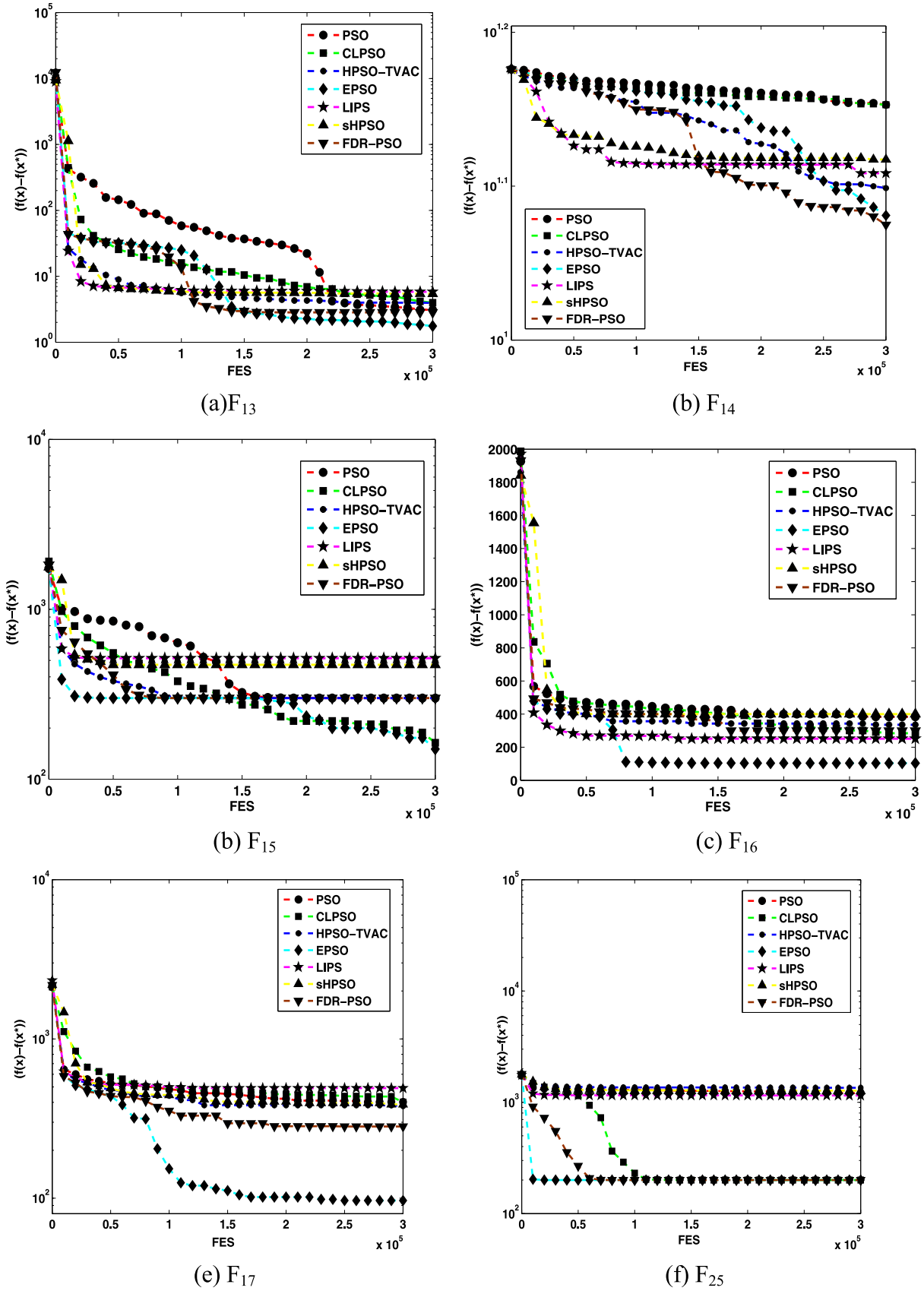


Fig. 4. Median convergence characteristics graphs of 30 dimensional CEC2005 benchmark functions: (a) F_{13}: Expanded Extended Griewank's plus Rosenbrock's Function (F8F2), (b) F_{14}: Shifted Rotated Expanded Scaffer's F_6, (c) F_{15}: Hybrid Composition Function, (d) F_{16}: Rotated Hybrid Composition Function, (e) F_{17}: Rotated Hybrid Composition Function with Noise in fitness, (f) F_{25}: Rotated Hybrid Composition Function without Bounds.

ter than HPSO-TVAC algorithm. Therefore, the computation cost of EPSO algorithm is reasonable and affordable compared to the computational cost of its individual PSO variants.

4.4. Search behavior of EPSO on benchmark problems

In this section, the different search behavior profiles of EPSO algorithm on different benchmark problems are discussed together with their 3D maps for 2D functions. The 3D maps of the benchmark problems are obtained from the CEC2005 benchmark competition [2]. The probability values (p_k) of the PSO strategies in EPSO are plotted against the number of generations to show the search behavior profiles of the proposed EPSO algorithm. The average behavior profiles of 30 trial runs for CEC2005 benchmark problems are shown in Figs. 1 and 2.

In the proposed EPSO algorithm, *inertia weight* PSO, HPSO-TVAC, FDR-PSO, LIPS and CLPSO strategies are employed through the self-adaptive evaluation strategy. *Inertia weight* PSO is used to ensure fast converge speed. HPSO-TVAC reinitializes the particles whenever they are stagnated in the search space and uses time-varying acceleration coefficients to enhance the global search in the early stage of the optimization phase and encourage local search at the end of the process. Thus, HPSO-TVAC in EPSO prevents premature convergence. In addition, FDR-PSO is used to enhance local search capability. Searching multiple local optima in multimodal optimization is achieved by using LIPS in the ensemble PSO (EPSO) algorithm. Furthermore, CLPSO-*gbest* is deployed to enhance the performance on solving multimodal optimization problems.

In Figs. 1 and 2, the search behavior profiles of EPSO algorithm on CEC2005 unimodal, multimodal, expanded and hybrid composition functions are shown alongside with their 3-D maps for 2D functions. The figures indicate that the behavior of EPSO is varying from one problem to another. In simple unimodal sphere function, as shown in Fig. 1(a), EPSO promotes HPSO-TVAC algorithm in the early stage of the optimization process. Towards the global optimum, LIPSO, FDR-PSO, and CLPSO-*gbest* variants are assigned alternately to find the better solutions. After the population is converged, all algorithms became equal probability.

In the multimodal Shifted Rosenbrock's function shown in Fig. 1(b), EPSO assigned HPSO-TVAC algorithm at the beginning of optimization process and switched to PSO at the end of the process to find the near-optimum solution/the optimum. On the other hand, in the multimodal Shifted Rotated Ackley's function with the multiple local optima, LIPS, FDR-PSO, and CLPSO-*gbest* algorithms took the lead roles in finding the optimal solution as shown in Fig. 1(c). Similarly, in the Shifted Rotated Rastrigin's problem with multiple local optima, FDR-PSO and LIPS have higher probability values in the early stage as shown in Fig. 2(a). After get-ting the good solution, HPSO-TVAC fine-tuned to find better solutions. Finally, PSO drove the particle to converge toward the near-optimum/optimum solution.

The search behavior profiles of EPSO on the Shifted Expanded Griewank's plus Rosenbrock's problem and hybrid composition problem with multiple local optima are illustrated in Fig. 2(b) and (c). The figures showed that the search behavior profiles are similar to that of unimodal as well as multimodal problems. It can be observed from the search behavior profiles that EPSO employs various PSO strategies throughout the optimization process. During the optimization process, EPSO promotes its PSO strategies based on their merits to take the leading role in finding better solutions. Different PSO strategies are promoted for different types of problems. Therefore, the proposed EPSO algorithm performs consistently on different types of benchmark problems from CEC2005 benchmark competition.

4.5. Discussions

The performance of the proposed ensemble PSO (EPSO) algorithm is investigated using different classes of problems such as unimodal, multimodal, expanded and hybrid composition problems. The experimental results of the EPSO algorithm are compared with the results of other PSO algorithm. The experimental results show consistently the EPSO performs better than its individual PSO strategies as well as other recent PSO algorithms regardless of problem types and dimensionality. It can be observed from the convergence graphs that EPSO maintains the population diversity throughout the evolutionary search process and enables to avoid premature convergence. In addition, the computational cost of EPSO algorithm is compared with its individual PSO strategy. The results show EPSO offers the reasonable and affordable computation cost compared to the computational cost of its individual PSO variants. Furthermore, it can be seen from the search behavior profiles that EPSO promotes different PSO strategies on different types of problems and obtains either the best or second best results on most of the test problems. This verified that the proposed ensemble PSO (EPSO) algorithm benefits from each individual PSO algorithm and enhances the universality and robust performance.

5. Conclusions

In this paper, ensemble particle swarm optimizer (EPSO) is proposed in which a pool of PSO strategies is constructed with *inertia weight* PSO, LIPS, HPSO-TVAC, FDR-PSO and CLPSO algorithms. Then the suitable PSO strategy is gradually selected through meritocracy approach to guide the particle's flying direction in the current stage of the search process. In addition, CLPSO is assigned to guide the particles in the small subpopulation to preserve the population diversity. In this way, the diversity is maintained throughout the search process and EPSO encourages the particles to converge to the global optimum without getting trap into a local optimum. The performance of EPSO algorithm is evaluated using four different types of 25 CEC2005 benchmark functions. The experimental results show that overall, the proposed EPSO algorithms performs well consistently on all four different types of problems and outperforms its individual PSO variants as well as recent state-of-the-art-PSO algorithms. The future research directions for EPSO include introducing dynamic population size and investigating different selection strategies using the same PSO algorithms.

References

[1] S. Das, P.N. Suganthan, Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems, Jadavpur University, Nanyang Technological University, Kolkata, 2010.
[2] P. Suganthan, N. Hansen, J. Liang, K. Deb, Y.-P. Chen, A. Auger, et al., Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization, presented at the IEEE congress on evolutionary computation 2005, 2005.
[3] J. Liang, T.P. Runarsson, M. Mezura-Montes, P. Clerc, C.A.C. Coello, et al., Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization, Nangyang Technological University, Singapore, 2006.
[4] R. Mallipeddi, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2010 Competition on Constrained Real-Parameter Optimization, Nanyang Technological University, Singapore, 2010.
[5] B. Liang, P. Suganthan, A.G. Hernández-Díaz, Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, 2013.
[6] J. Liang, B. Qu, P. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Zhengzhou University, China, Zhongyuan University of Technology, Zhengzhou, China and Nanyang Technological University, Singapore, 2013.
[7] B. Liang, P. Suganthan, Q. Chen, Problem Definitions and Evaluation Criteria for the CEC 2015 Competition on Learning-Based Real-Parameter Single

Objective Optimization, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 2014.
[8] M.Y. Cheng, D.H. Tran, Two-phase differential evolution for the multiobjective optimization of time–cost tradeoffs in resource-Constrained construction projects, IEEE Trans. Eng. Manage. 61 (2014) 450–461.
[9] M.R. Tanweer, S. Suresh, N. Sundararajan, Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems, Inf. Sci. 326 (2016) 1–24, 1/1/2016.
[10] Y. Xu, K. Li, J. Hu, K. Li, A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues, Inf. Sci. 270 (2014) 255–287, 6/20/2014.
[11] Y. Xu, K. Li, L. He, L. Zhang, K. Li, A hybrid chemical reaction optimization scheme for task scheduling on heterogeneous computing systems, IEEE Trans. Parallel Distrib. Syst. 26 (2015) 3208–3222.
[12] T.C. Carneiro, S.P. Melo, P.C.M. Carvalho, A.P.d.S. Braga, Particle swarm optimization method for estimation of Weibull parameters: a case study for the Brazilian northeast region, Renewable Energy 86 (2016) 751–759 (2//2016).
[13] A. Ouyang, Z. Tang, X. Zhou, Y. Xu, G. Pan, K. Li, Parallel hybrid PSO with CUDA for lD heat conduction equation, Comput. Fluids 110 (2015) 198–210, 3/30/2015.
[14] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82.
[15] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, Proceedings of the 1999 Congress on Evolutionary Computation, 1999 (1999).
[16] J.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 (2006) 281–295.
[17] T. Peram, K. Veeramachaneni, C.K. Mohan, Fitness-distance-ratio based particle swarm optimization, Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE (2003) 174–181.
[18] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (2004) 240–255.
[19] B.Y. Qu, P.N. Suganthan, S. Das, A distance-based locally informed particle swarm model for multimodal optimization, IEEE Trans. Evol. Comput. 17 (2013) 387–402.
[20] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2009) 398–417.
[21] J.E. Baker, Reducing bias and inefficiency in the selection algorithm, Proceedings of the Second International Conference on Genetic Algorithms (1987) 14–21.
[22] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, Proceedings of the Sixth International Symposium on Micro Machine and Human Science (1995) 39–43.
[23] J. Kennedy, R. Eberhart, Particle swarm optimization, IEEE International Conference on Neural Networks 1995 Proceedings vol 4 (1995) 1942–1948.
[24] K.E. Parsopoulos, Particle Swarm Optimization and Intelligence: Advances and Applications: Advances and Applications: Information Science Reference, 2010.
[25] X.D. Li, A.P. Engelbrecht, Particle swarm optimization: An introduction and its recent developments, presented at the Proceedings of the Genetic and Evolutionary Computation Conference, 2007.
[26] X. Hu, Y. Shi, R.C. Eberhart, Recent advances in particle swarm, IEEE Congress on Evolutionary Computation (2004) 90–97.
[27] A. Banks, J. Vincent, C. Anyakoha, A review of particle swarm optimization. Part I: background and development, Nat. Comput. 6 (2007) 467–484.
[28] A.P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, Wiley, New York, 2006, pp. 2006.
[29] Y. Shi, R. Eberhart, A modified particle swarm optimizer, IEEE International Conference on Evolutionary Computation Proceedings, 1998 (1998) 69–73.
[30] Z.H. Zhan, J. Zhang, Y. Li, Y.H. Shi, Orthogonal learning particle swarm optimization, IEEE Trans. Evol. Comput. 15 (2011) 832–847.
[31] W. Du, B. Li, Multi-strategy ensemble particle swarm optimization for dynamic optimization, Inf. Sci. 178 (2008) 3096–3109.
[32] A.P. Engelbrecht, Heterogeneous particle swarm optimization, in: Swarm Intelligence, Springer, 2010, pp. 191–202.
[33] M.A.M.d. Oca, J. Pena, T. Stutzle, C. Pinciroli, M. Dorigo, Heterogeneous particle swarm optimizers, IEEE Congress on Evolutionary Computation. CEC '09 (2009) 698–705.
[34] F.V. Nepomuceno, A.P. Engelbrecht, A self-adaptive heterogeneous pso for real-parameter optimization, IEEE Congress on Evolutionary Computation (CEC) (2013) 361–368.
[35] M.A. Potter, The Design and Analysis of a Computational Model of Cooperative Coevolution, Citeseer, 1997.
[36] O. Olorunda, A.P. Engelbrecht, An analysis of heterogeneous cooperative algorithms, IEEE Congress on Evolutionary Computation, 2009. CEC '09, 2009 (2009) 1562–1569.
[37] J.A. Vrugt, B.A. Robinson, J.M. Hyman, Self-adaptive multimethod search for global optimization in real-parameter spaces, IEEE Trans. Evol. Comput. 13 (2009) 243–259.
[38] J. Grobler, A.P. Engelbrecht, G. Kendall, V.S.S. Yadavalli, Alternative hyper-heuristic strategies for multi-method global optimization, IEEE Congress on Evolutionary Computation (CEC) (2010) 1–8.
[39] E.K. Burke, G. Kendall, E. Soubeiga, A tabu-search hyperheuristic for timetabling and rostering, J. Heuristics 9 (2003) 451–470.
[40] W.M. Spears, Adapting crossover in evolutionary algorithms, The 4th Annual Conference on Evolutionary Programming (1995) 367–384.
[41] H.-S. Yoon, B.-R. Moon, An empirical study on the synergy of multiple crossover operators, IEEE Trans. Evol. Comput. 6 (2002) 212–223.
[42] A.K. Qin, P.N. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, IEEE Congress on Evolutionary Computation, 2005 2005 (2005) 1785–1791.
[43] R. Mallipeddi, S. Mallipeddi, P.N. Suganthan, Ensemble strategies with adaptive evolutionary programming, Inf. Sci. 180 (2010) 1571–1581.
[44] H. Wang, Z. Wu, S. Rahnamayan, H. Sun, Y. Liu, J.-s. Pan, Multi-strategy ensemble artificial bee colony algorithm, Inf. Sci. 279 (2014) 587–603.
[45] H. Wang, W. Wang, H. Sun, Multi-strategy ensemble artificial bee colony algorithm for large-scale production scheduling problem, Int. J. Innovative Comput. Appl. 6 (2015) 128–136.
[46] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, IEEE Trans. Evol. Comput. 8 (2004) 204–210.
[47] J. Kennedy, The particle swarm: social adaptation of knowledge, IEEE International Conference on Evolutionary Computation (1997) 303–308.
[48] J. Kennedy, Bare bones particle swarms, Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE, 2003 (2017) 80–87.
[49] J.D. Gibbons, S. Chakraborti, Nonparametric statistical inference, in: M. Lovric (Ed.), International Encyclopedia of Statistical Science, Springer, Berlin, Heidelberg, 2011, pp. 977–979.