



An improved ensemble particle swarm optimizer using niching behavior and covariance matrix adapted retreat phase

Libin Hong^{a,*}, Xinmeng Yu^a, Ben Wang^a, John Woodward^b, Ender Özcan^c

^a Hangzhou Normal University, No. 2318 Yuhangtang Road, Hangzhou 311121, China

^b Queen Mary University of London, Mile End Road, London E1 2AD, UK

^c University of Nottingham, Wollaton Road, Nottingham NG8 1BB, UK

ARTICLE INFO

Keywords:

Particle swarm optimizer
Niching behavior
Covariance matrix adapted retreat
Ensemble strategy

ABSTRACT

Over the past two decades, to overcome the limitations of certain algorithms, ensemble strategies or self-adaptive mechanisms for evolutionary computation algorithms have been proposed. Regardless of how these strategies or mechanisms were designed, their objective was to control the balance between the global and local search capabilities during the evolutionary process. Inspired by this, a novel ensemble strategy with three groups to improve the performance of the ensemble particle swarm optimizer (EPSO) is proposed. The first group uses a covariance matrix adapted retreat phase (CMAR), the second group induces the niching behavior for inertia weight particle swarm optimization (PSO), and the third group maintains the characteristics of a large subpopulation of EPSO. Furthermore, a sample pool and replacement mechanism are proposed to perturb the three groups. This strategy also recommends a group of empirical allocation rates for subpopulations based on various proportion combination tests. The performance of the proposed strategy is evaluated using CEC2005 benchmark functions with 10, 30, and 50-dimensional tests and compared with those of the state-of-the-art PSO variants: EPSO, a modified PSO using adaptive strategy (MPSO), PSO variant for single-objective numerical optimization (PSO-sono), terminal crossover and steering-based PSO (TCS-PSO), self-adapting hybrid strategy PSO (SaHSPS), and pyramid PSO (PPSO). Experimental results demonstrate that the improved EPSO, using CMAR, niching behavior, and sample pool, performs best.

1. Introduction

The “No Free Lunch (NFL)” theorem [1] states that there is no single optimization algorithm that can solve every problem effectively and efficiently. To overcome the limitations of individual algorithms, investigations on ensemble strategies and self-adaptive mechanisms for evolutionary computation algorithms have gained attention in recent years [2–8]. The particle swarm optimization (PSO) algorithm was primarily proposed by Kennedy and Eberhart [9]; it is widely applied to real-world problems such as energy efficient cluster head selection in wireless sensor networks [10], high dimensional data classification [11], circular hotspots detection [12], and the sparse reconstruction problem [13], etc. In [14], the research directions of the PSO algorithm were divided into four aspects: parameter tuning [15–17], topology choices [18,19], learning strategy improvements [20–22], and integration with other algorithms [23–25].

In many PSO-based hybridized algorithms, the hybridization can be classified into two types: either between PSO-based algorithms and other algorithms, or between PSO-based algorithms and their

variants. Moradi et al. [26] proposed a hybridized PSO with a local search strategy (HPSO-LS) to select the less correlated and salient feature subset. Wang et al. [27] proposed a hybrid PSO algorithm that employs an adaptive learning strategy (ALPSO), which employs a self-learning based candidate generation strategy to ensure exploration ability and a competitive-learning-based prediction strategy to guarantee exploitation of the algorithm. Chen et al. [28] proposed a hybrid PSO with a spiral-shaped mechanism (HPSO-SSM) to select the optimal feature subset for classification via a wrapper-based approach. Recently, Chih [29] proposed a PSO variant using pseudo-random number assignment strategy, introducing a new approach to PSO research. The ensemble particle swarm optimizer (EPSO) [5] demonstrates impressive performance by hybridizing different PSO variants: inertia weight PSO [30], comprehensive-learning PSO (CLPSO) [31], gbest-guided CLPSO, fitness-distance-ratio-based PSO (FDR-PSO) [32], self-organizing hierarchical PSO with time-varying acceleration coefficient (HPSO-TVAC) [33], and distance-based locally informed PSO (LIPSO) [34]. These PSO variants employ features that complement

* Corresponding author.

E-mail address: libin.hong@hznu.edu.cn (L. Hong).

<https://doi.org/10.1016/j.swevo.2023.101278>

Received 15 July 2022; Received in revised form 24 December 2022; Accepted 9 February 2023

Available online 20 February 2023

2210-6502/© 2023 Elsevier B.V. All rights reserved.

each other: inertia weight PSO and HPSO-TVAC excel in solving unimodal problems; CLPSO, FDR-PSO, and LIPSO excel in solving multimodal problems; inertia weight PSO guarantees fast convergence speed; HPSO-TVAC prevents premature convergence; CLPSO maintains population diversity; gbest-guided CLPSO encourages the particles to converge to the global optimum while maintaining the population diversity; and FDR-PSO enhances local search capability.

Previous studies have made great progress on ensemble strategies and self-adaptive mechanisms; their main motivation was to incorporate the advantages of different optimization algorithms when solving complex optimization problems. Wu et al. [35] pointed out that incorporating ensemble strategies into population-based optimization algorithms (POAs) can benefit from both the availability of diverse approaches at various stages and ease the computationally frequent offline tuning, and the algorithmic components of various advantages can support one another during the evolutionary process. Thus, the ensemble of components can potentially result in a versatile POA. These designed strategies and mechanisms are expected to maximize the strengths of the optimization algorithms at different stages of the evolutionary process. Most of the work was designed to offer a delicate balance between the global and local search capabilities of the optimization algorithms.

The improved ensemble particle swarm optimizer (IEPSO) retains the complementary characteristics of EPSO and enhances the global and local search capabilities. The EPSO large subpopulation employs the inertia weight PSO, HPSO-TVAC, FDR-PSO, LIPSO and gbest-guided CLPSO algorithms. Niching methods have been widely used in evolutionary algorithms (EAs) [36–42], to effectively prevent the EAs from losing population diversity and converging to a solitary basin of attraction [43] in scenarios where the locations of multiple optima are targeted. Inertia weight PSO converges rapidly to the global optima and performs better than the original PSO; it performs well with unimodal functions and does not show outstanding performance on complicated multimodal functions [15]. The loss of diversity of particles is the major reason that inertia weight PSO shows such characteristics; this feature motivated us to employ niching-based technologies to help the inertia weight PSO maintain the diversity of particles during the evolutionary process. CLPSO is assigned to the small subpopulation in EPSO, therefore both small and large subpopulations of EPSO do not use dedicated local search methods. This design drove us to integrate a local search method into IEPSO; through comparison and testing, a covariance matrix adapted retreat (CMAR) phase is developed. Covariance matrix adaptation (CMA) was originally designed to perform well with small populations [44,45]. However, [46] presents a method for handling larger populations efficiently by increasing the adaptation speed of the CMA. In [47], a CMAR phase was proposed to improve the local search capability of the effective butterfly optimizer (EBO). IEPSO employs the CMA proposed in [46] to control both the covariance matrix and step size adaptively. Information sharing is an effective mechanism during the optimization process, thus it motivated us to design a sample pool that can perturb the subpopulations. The purpose of designing a sample pool is to share the information between particles. Inspired by the above observations, IEPSO, which integrates the above inspirations, is proposed in this paper. The major contributions of this study are as follows:

- A novel ensemble strategy, which combines niching-based inertia weight PSO, CMAR, and a major part of EPSO, is proposed. A niching-based method induces and maintains population diversity, especially in the early stages of the evolutionary process, thus enhancing the global search capability. The CMAR is employed throughout the optimization process to enhance the local search capability.
- A group of empirical allocation rates, based on the various proportion combination tests, is recommended for the novel ensemble strategy. The allocation rates for CMAR, niching-based inertia weight PSO and EPSO large subpopulations are 10%, 20%, and 70%, respectively.

- A sample pool and replacement mechanism are proposed and they perturb the subpopulations. The particles in the sample pool are from the niching-based inertia weight PSO, CMAR, and EPSO large subpopulations. The sample pool is used to guide the evolution of the three subpopulations. In essence, the sample pool shares information among subpopulations.
- A fitness sharing-based method is employed to induce niching behavior for the inertia weight PSO in the proposed ensemble strategy. In this method, a novel mechanism is proposed to replace the sharing function.

The remainder of this paper is organized as follows. Section 2 describes the PSO and EPSO. Section 3 describes the strategy of the proposed IEPSO. The EPSO large subpopulation, niching-based inertia weight PSO subpopulation, CMAR subpopulation, sample pool, and perturbing mechanism are presented in detail. Section 4 presents the parameter settings of the tested algorithms and the experimental results. Comparisons with state-of-the-art optimization algorithms, analysis, and further discussions are also presented in this section. Section 5 summarizes and concludes the paper.

2. Particle swarm optimization and ensemble particle swarm optimizer

2.1. Particle swarm optimization

In classical PSO, proposed by James Kennedy and Russell C. Eberhart [48], updating the velocity and position of each particle is the basic operation. The velocity and position of the i th particle at generation t are recorded as $V_i(t)$ and $X_i(t)$, respectively. The velocity and position update formulas of particle i are presented below.

$$V_i^d(t+1) = V_i^d(t) + c_1 \cdot r1_i^d \cdot (Pbest_i^d(t) - X_i^d(t)) + c_2 \cdot r2_i^d \cdot (Gbest^d(t) - X_i^d(t)) \quad (1)$$

$$X_i^d(t+1) = X_i^d(t) + V_i^d(t+1) \quad (2)$$

$r1_i^d$ and $r2_i^d$ are random numbers uniformly distributed in $[0,1]$, $d = 1, 2, \dots, D$, where D represents the dimension. $Pbest_i^d(t)$ is the i th particle's previous best experience and $Gbest^d(t)$ is the whole swarm's best experience, and are as defined below.

$$Pbest_i^d(t) = \min\{fit(X_i^d(1)), fit(X_i^d(2)), \dots, fit(X_i^d(t))\} \quad (3)$$

$$Gbest^d(t) = \min\{fit(Pbest_1^d(t)), fit(Pbest_2^d(t)), \dots, fit(Pbest_N^d(t))\} \quad (4)$$

In inertia weight PSO, an inertia weight w is introduced to Eq. (5), forming an equation to update the velocity of the particles, which is given below; the inertia weight w plays the role of balancing the local and global search [15].

$$V_i^d(t+1) = w \cdot V_i^d(t) + c_1 \cdot r1_i^d \cdot (Pbest_i^d(t) - X_i^d(t)) + c_2 \cdot r2_i^d \cdot (Gbest^d(t) - X_i^d(t)) \quad (5)$$

CLPSO, gbest-guided CLPSO, FDR-PSO, HPSO-TVAC, and LIPSO use the velocity Eqs. (6), (7), (8), (9), and (10), respectively [34]. In Eq. (6), the $Pbest_{fi}^d$ can use the information derived from the historical best positions of different particles to generate new positions in the search space; therefore, it can increase the diversity of the swarm to yield improved performance when optimizing multimodal functions [31].

$$V_i^d(t+1) = w \cdot V_i^d(t) + c_1 \cdot r1_i^d \cdot (Pbest_{fi(d)}^d(t) - X_i^d(t)) \quad (6)$$

In Eq. (7), $Gbest^d$ is introduced with the overall best experience of the swarm; this rule encourages the particles to converge to the global optimum while maintaining the diversity of the swarm [5].

$$V_i^d(t+1) = w \cdot V_i^d(t) + c_1 \cdot r1_i^d \cdot (Pbest_{fi(d)}^d(t) - X_i^d(t)) + c_2 \cdot r2_i^d \cdot (Gbest^d(t) - X_i^d(t)) \quad (7)$$

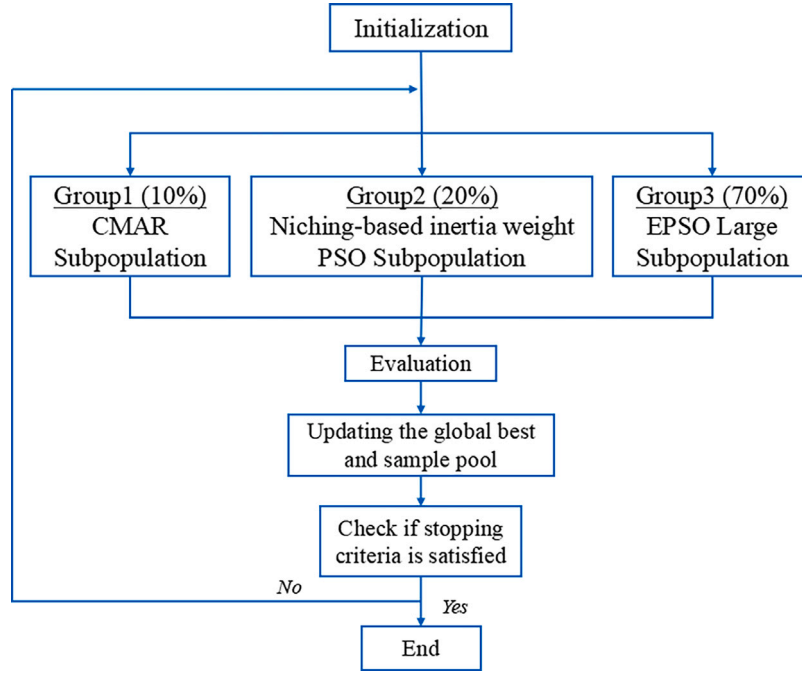


Fig. 1. Main workflow of IEP SO.

In Eq. (8), $nbest_j^d$ lets each particle further learn from the experience of its neighboring particles.

$$V_i^d(t+1) = w \cdot V_i^d(t) + c_1 \cdot r1_i^d \cdot (Pbest_i^d(t) - X_i^d(t)) + c_2 \cdot r2_i^d \cdot (Gbest^d(t) - X_i^d(t)) + c3 \cdot (nbest_j^d(t) - X_i^d(t)) \quad (8)$$

HPSO-TVAC removes $w \cdot V_i^d$ from Eq. (9) and time-varying acceleration coefficients are introduced to reduce premature convergence in coefficients.

$$V_i^d(t+1) = c_1 \cdot r1_i^d \cdot (Pbest_i^d(t) - X_i^d(t)) + c_2 \cdot r2_i^d \cdot (Gbest^d(t) - X_i^d(t)) \quad (9)$$

In Eq. (10), LIPSO utilizes multiple local best particles to guide the search of each particle; it can manipulate the settled niching behavior using the information offered by the neighborhoods.

$$V_i^d(t+1) = w \cdot (V_i^d(t) + \psi(P_i^d(t)) + X_i^d(t)) \quad (10)$$

where,

$$P_i = \frac{\sum_{j=1}^{nsize} (\psi_j \cdot nbest_j) / nsize}{\psi}, \psi_j \sim U(0, \frac{4.1}{nsize}), \psi = \sum_{j=1}^{nsize} \psi_j \quad (11)$$

Here, $w = 0.7298$, U represents uniform distribution, $nbest_j$ is the j th nearest neighborhood to i th particle's $Pbest$, and $nsize$ is the size of neighborhood.

2.2. Ensemble particle swarm optimizer

Lynn et al. [5] proposed an EPSO that employs complementary PSO variants using the following step-by-step self-adaptive selection strategy. The CLPSO is assigned to the small subpopulation, the inertia weight PSO, FDR-PSO, HPSO-TVAC, LIPSO and gbest-guided CLPSO are employed in the large subpopulation.

- Step01: Set $p_k = \frac{1}{k}$ for each PSO variant, where k is the number of PSO variants in the EPSO variant pool.
- Step02: The stochastic universal selection method is used to select the candidate PSO variant for the particles.
- Step03: If the selected PSO variant produces improved solutions at generation G , the selected PSO variant is increased by recording it as

a success memory $ns_{k,G}$. Otherwise, the PSO variant is decreased by recording it as a failure memory $nf_{k,G}$.

- Step04: The success and failure memories are updated for a fixed number of generations, called the learning period (LP). If the memories overflow after the LP, the earliest record stored in the memories will be removed for the numbers obtained in the current generation to be stored in the memory.
- Step05: For subsequent generations after the LP, the probability of selecting PSO variants from the pool are updated as follows:

$$P_{k,G} = \frac{S_{k,G}}{\sum_{k=1}^k S_{k,G}} \quad (12)$$

where,

$$S_{k,G} = \frac{\sum_{g=G-LP}^{G-1} ns_{k,g}}{\sum_{g=G-LP}^{G-1} ns_{k,g} + \sum_{g=G-LP}^{G-1} nf_{k,g}} + \epsilon \quad (13)$$

$G > LP$ and $S_{k,G}$ represent the success rates of the solutions generated by the k th PSO variant. $\epsilon = 0.01$ is used to avoid the possible null success rate. The PSO variant with a greater success rate has a higher probability to be selected to guide the particle in the current generation.

3. Improved ensemble particle swarm optimizer

A novel ensemble strategy is proposed for IEP SO. The particles are divided into three subpopulations: EPSO large subpopulation, niching-based inertia weight PSO subpopulation, and CMAR subpopulation. In EPSO, the population is composed of small and large subpopulations. In IEP SO, the small subpopulation is replaced by a niching-based inertia weight PSO subpopulation and CMAR subpopulation. The allocation rates of the different subpopulations are as follows. 70% of the particles are evolved by EPSO with a large subpopulation, 20% are evolved by niching-based inertia weight PSO, and 10% are evolved by CMAR. The proportion allocation is based on the various proportion combination tests; therefore, it is an empirical value. To enhance the population diversity, a mechanism which can perturb the three subpopulations is designed. This mechanism helps the particles avoid falling into local optima ahead of time. Fig. 1 is the main workflow chart of IEP SO. The pseudocode of IEP SO is given in Algorithm 1, and the niching-based

inertia weight PSO is described in Algorithm 2 which is invoked at line 9 of Algorithm 1.

Algorithm 1 Pseudocode of IEPSO

Input: Solution space $[X_{min}^D, X_{max}^D]$, velocity bound $[V_{min}^D, V_{max}^D]$, maximum number of function evaluation nfe_{max} , objective function $f(X)$
Output: Optimal location X_{gbest} , best fitness value $f(X_{gbest})$.
 population size = PS, $t = 0$.

- 1: **for** $i = 1 : PS$ **do**
- 2: Initialize the i^{th} particle, $V_{i,t}$, $X_{i,t}$.
- 3: Calculate the fitness value $f(X_{i,t})$.
- 4: **end for**
- 5: Label $X_{gbest,t}$, $f(X_{gbest,t})$.
- 6: **while** termination condition is not satisfied **do**
- 7: Separate the population into three subpopulations according to the ratio 10% -20% -70% .
- 8: Update particles in the CMAR subpopulation (10%).
- 9: Update particles in the niching-based inertia weight PSO subpopulation (20%).
- 10: Update particles in the EPSO Large subpopulation (70%).
- 11: Update $X_{gbest,t}$, $f(X_{gbest,t})$.
- 12: **for** $i = 1 : PS$ **do**
- 13: Calculate the probability $Prob_i$ of being selected into the sample pool using Eq. (24).
- 14: **if** $\text{rand}(0,1) < Prob_i$ **then**
- 15: Add the i^{th} particle into sample pool.
- 16: **end if**
- 17: **end for**
- 18: Randomly select two particles from sample pool to replace one particle in the CMAR subpopulation and another in niching-based PSO subpopulation.
- 19: $t = t + 1$.
- 20: **end while**

Algorithm 2 Pseudocode of niching-based inertia weight PSO

Define PS as the population size of niching-based inertia weight PSO subpopulation, t indicates the current generation, $Gbest_t$ is the best particle among all the three subpopulations at t^{th} generation and $f(Gbest_t)$ is the fitness value of $Gbest_t$.

- 1: **for** $i = 1 : PS$ **do**
- 2: Initialize the i^{th} particle, $V_{i,t}$, $X_{i,t}$.
- 3: Calculate the fitness value $f(X_{i,t})$.
- 4: **end for**
- 5: **while** termination condition is not satisfied **do**
- 6: Label the best particle $X_{group_best,t}$, $f(X_{group_best,t})$.
- 7: **for** $i = 1 : PS$ **do**
- 8: Calculate the Euclidean distance between $X_{i,t}$ and $X_{group_best,t}$ as $Distance_i$.
- 9: **end for**
- 10: Take the mean of $Distance$ as the niche radius in the current generation, denoted as r_t .
- 11: **for** $i = 1 : PS$ **do**
- 12: **if** $\text{rand}_i(0, 1) > \frac{t}{\text{max_generation}}$ **then**
- 13: Take $X_{i,t}$ as the center of the circle and r_t as the radius.
- 14: The particle within this circle which has the smallest fitness value is marked as $Nbest_{i,t}$.
- 15: **else**
- 16: $Nbest_{i,t} = Gbest_t$.
- 17: **end if**
- 18: Update $X_{i,t}$, $V_{i,t}$ using Eqs. (2) and (16), respectively.
- 19: **end for**
- 20: Update $Gbest_t$, $f(Gbest_t)$.
- 21: $t = t + 1$.
- 22: **end while**

3.1. Inducing niching behavior for inertia weight PSO

Fitness sharing is one of the common niching techniques. The basic concept of fitness sharing is to divide the population into different subgroups according to the similarity of the individuals. An individual, which is called a particle in PSO, shares its information with other individuals within the same niche. The shared fitness $f(i)'$ of i^{th} individual with fitness $f(i)$ can be represented as follows:

$$f(i)' = \frac{f(i)}{\sum_{j=1}^N sh(d_{ij})} \quad (14)$$

where the sharing function is represented as:

$$sh(d_{ij}) = \begin{cases} 1 - (d_{ij}/\sigma_s)^\alpha, & \text{if } d_{ij} < \sigma_s \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

where d_{ij} is the distance between individuals i and j , σ_s is the sharing radius, N is the population size, and α is a constant called sharing level.

In IEPSO, the sharing function is replaced by Eq. (17). The particles evolve in their own niche and $niching_best$ is used, and the formula for calculating the radius r is re-designed. In this way, the niching behavior of inertia weight PSO is induced by the following equations:

$$V_i^d(t+1) = w \cdot V_i^d(t) + c_1 \cdot r1_i^d \cdot (Pbest_i^d(t) - X_i^d(t)) + c_2 \cdot r2_i^d \cdot (Nbest^d(t) - X_i^d(t)) \quad (16)$$

where,

$$Nbest^d = \begin{cases} niching_best_i^d, & \text{if } rand_i > \frac{\text{current_generation}}{\text{max_generation}} \\ Gbest^d, & \text{otherwise.} \end{cases} \quad (17)$$

where $Gbest^d$ is the best particle among all the three subpopulations and $niching_best_i^d$ is the best particle in the i^{th} particle's niche. If we consider each particle as the center of a circle, the radius r is calculated as:

$$r = \frac{1}{niching_PS - 1} \sum_{i=1}^{niching_PS} |niching_X_i^d - niching_X_{group_best}^d| \quad (18)$$

where $niching_X_{group_best}^d$ denotes the best particle of the niching-based inertia weight PSO subpopulation in the current generation. $niching_PS$ is the number of particles in the niching-based inertia weight PSO subpopulation. $|niching_X_i^d - niching_X_{group_best}^d|$ denotes the Euclidean distance between $niching_X_i^d$ and $niching_X_{group_best}^d$.

3.2. CMAR phase employed in IEPSO

The incorporation of the CMAR phase includes the following steps: [46,47,49]:

- Step01: Two multivariate normal distributions, $M1(m, C)$ and $M2(m, C)$, are used to generate new particles, with a probability of 50%. Both the distributions are individually controlled by its mean $m \in \mathbb{R}^n$ and a symmetric and positive definite covariance matrix $C \in \mathbb{R}^{n \times n}$. The distributions $M1$ and $M2$ can be represented as: $M1(m, C) \sim m + M1(0, C)$ and $M2(m, C) \sim m + M2(0, C)$, where “ $\sim$ ” denotes equality in distribution.
- Step02: The new particles are produced by the following equation:

$$CMAR_X_{i+1}^t = \begin{cases} m^t + \sigma^t \cdot M1(0, C^t), & \text{if } rand_i \leq 0.5 \\ m^t + \sigma^t \cdot M2(0, C^t), & \text{otherwise.} \end{cases} \quad (19)$$

where σ is a step size to control the size of the distribution and $CMAR_X_i$ represents a particle of the CMAR subpopulation.

- Step03: The mean m is updated by the weighted average of half of the best particles of the CMAR subpopulation ($\{CMAR_X_1, CMAR_X_2, CMAR_X_3, \dots, CMAR_X_{CMAR_PS}\}$), as shown in Eq. (20).

$$m^t = \sum_{i=1}^{CMAR_PS} \frac{1}{2} w_i^t \cdot CMAR_X_i^t \quad (20)$$

Here, $CMAR_PS$ is the population size of the CMAR subpopulation and the weights w_i ($\sum_{i=1}^{CMAR_PS} w_i = 1$ and $w_1 \geq w_2 \geq \dots \geq w_{CMAR_PS} \geq 0$) are calculated using Eq. (21).

$$w_i^t = 1 - \frac{fitness(CMAR_X_i^t)}{\sum_{z=1}^{CMAR_PS} fitness(CMAR_X_z^t)} \quad (21)$$

- **Step04:** The covariance matrix C^t and global step size σ^t adaption are calculated using Eqs. (22) and (23), respectively, according to [46, 49].

$$C^{t+1} = (1 - c_1 - c_\mu) \cdot C^t + c_1 \cdot p_c^{t+1} \cdot (p_c^{t+1})^T + c_\mu \cdot \sum_{i=1}^{CMAR_PS} w_i \cdot y_i^{t+1} \cdot (y_i^{t+1})^T \quad (22)$$

where $c_1 = \frac{2}{(n+1.3)^2 + \mu_{eff}}$, $c_\mu = \min(1 - c_1, 2 \cdot \frac{\mu_{eff} - 2 + \frac{1}{\mu_{eff}}}{(n+2)^2 + \mu_{eff}})$, μ_{eff} is the variance effective selection mass for the mean, n is the dimension size, p_c^{t+1} is the evolution path, and $y_i^{t+1} = \frac{CMAR_X_i^{t+1} - m^t}{\sigma^t}$ [46, 49].

$$\sigma^{t+1} = \sigma^t \cdot \exp\left(\frac{c_\sigma}{d_\sigma} \cdot \left(\frac{\|p_\sigma^{t+1}\|}{E\|N(0, 1)\|} - 1\right)\right) \quad (23)$$

where $E\|N(0, 1)\|$ is the expected length of a $(0, I)$ -normally distributed random vector, $c_\sigma = \frac{\mu_{eff} + 2}{n + \mu_{eff} + 3}$, and $d_\sigma \approx 1$ is the damping parameter [46].

3.3. Sample pool and perturbing subpopulations

The framework in Fig. 2 shows how the sample pool perturbs the niching-based inertia weight PSO, CMAR, and EPSO large subpopulations. A probability P_i , where $i = 1, 2, 3, \dots, popsize$ for each particle at generation t is defined as:

$$P_i(t) = 1 - \frac{fitness_i(t) - Gbest(t)}{Gworst(t) - Gbest(t)} \quad (24)$$

If $rand_i(t) < P_i(t)$, the i th particle will be selected into the sample pool. Then, two particles will be selected randomly from the sample pool to replace a particle in the CMAR subpopulation and another in the niching-based inertia weight PSO subpopulation, randomly.

4. Performance evaluation

To evaluate the performance of IEP SO, we selected the state-of-the-art PSO algorithms and performed tests based on the CEC2005 benchmark functions. EPSO [5], a modified PSO using adaptive strategy (MP SO) [50], PSO variant for single-objective numerical optimization (PSO-sono) [51], terminal crossover and steering-based PSO (TCSP SO) [52], self-adapting hybrid strategy PSO (SaHSPS) [53], and pyramid PSO (PPSO) [54] algorithms were considered. For fair comparison, the function evaluations (FEs) for each algorithm were set to identical values with the same dimension size.

4.1. Parameter settings

FEs for all tested algorithms were set to 100,000, 300,000, and 500,000 for 10D, 30D, and 50D, respectively. In IEP SO, the population sizes for 10D, 30D, and 50D were set to 100, 140, and 150, respectively. For the subpopulation sizes and parameter settings of IEP SO, please refer to Table 1. The parameter settings for the compared algorithms are given in Table 2, and they follow the original investigators' recommendations, which usually have the best experimental effect on the optimization problems. The shifted and rotated CEC2005 benchmark functions are given in Table 3, $f_1 - f_5$ are unimodal functions, $f_6 - f_{12}$ are multimodal functions, $f_{13} - f_{14}$ are expanded functions, and $f_{15} - f_{25}$ are hybrid composition functions.

Table 1

Parameter settings for IEP SO.

Subpopulation in IEP SO	Default parameter settings
CMAR subpopulation	Subpopulation size for 10D/30D/50D=10/14/15, default settings of CMAR.
Niching-based inertia weight PSO subpopulation	Subpopulation size for 10D/30D/50D=20/28/30, $w=0.5$, $c_1=c_2=2$.
EPSO large subpopulation	Subpopulation size for 10D/30D/50D=70/98/105, default settings of its components.

Table 2

Parameter settings for the state-of-the-art PSO algorithms.

Algorithm	Default parameter settings
EPSO	Population size for 10D=20 (8/12), population size for 30D and 50D=40 (15/25), default settings of its components.
MP SO	Population size=50 (for 10D, 30D, and 50D), $w=0.9 \sim 0.4$, $cw=4 \times r \times (1-r)$, r is a random value and $r \in (0, 1)$.
PSO-sono	Population size=100 (for 10D, 30D, and 50D), $w=0.9 \sim 0.4$, $c = \frac{D}{PS} \times 0.01$, $r=0.5$.
TCSP SO	Population size=50 (for 10D, 30D, and 50D), $w=0.9 \sim 0.4$, $c_1=c_2=2$.
SaHSPS	Population size=50 (for 10D, 30D, and 50D), default settings of its components.
PPSO	Population size=64 (for 10D, 30D, and 50D), ρ is set to 0.008, 0.02 and 0.04 for 10D, 30D, and 50D, respectively.

4.2. Experimental results

The proposed IEP SO algorithm was compared with recent state-of-the-art PSO variants: EPSO, MP SO, PSO-sono, TCSP SO, SaHSPS, and PPSO. The experimental results for the 10, 30, and 50-dimensional tests are presented in Tables 6, 7 and 8. These tables summarize the results in terms of the mean, minimum, and median values in the last generation of the runs and their standard deviations over 30 runs for each function. The rankings for the mean, minimum, and median values are also listed in the tables. The best mean, minimum, and median values, among all variants, are in boldface. The top 1 rankings of the mean, minimum, and median values are in boldface as well. For ease of observing reliability and comparison, more decimal places are retained to illustrate robust performance of algorithms, instead of using scientific notation, for some of the values.

The Wilcoxon signed-rank test (WST) was performed for IEP SO versus EPSO, MP SO, PSO-sono, TCSP SO, SaHSPS, and PPSO. Tables 6, 7 and 8 also show the results of the WST at the 5% significance level, when comparing IEP SO with EPSO, MP SO, PSO-sono, TCSP SO, SaHSPS, and PPSO. In the tables, ' $\geq$ ' and ' $\leq$ ' indicate that IEP SO performs better or worse on f_n than EPSO, MP SO, PSO-sono, TCSP SO, SaHSPS, and PPSO. ' $>$ ' and ' $<$ ' indicate that IEP SO performs significantly better or worse than the compared algorithm, and ' $=$ ' means that there is no difference in the performances of the compared algorithms. In the majority of the cases, IEP SO outperforms EPSO, MP SO, PSO-sono, TCSP SO, SaHSPS, and PPSO.

To better illustrate the effectiveness of the tuned allocation rates and sample pool mechanism, the validation of different allocation rates was carried out. The allocation rates 0%–30%–70% refer to IEP SO without using CMAR subpopulation and 30%–0%–70% refer to IEP SO without using niching-based inertia weight PSO subpopulation. In addition, the IEP SO without using sample pool is tested. The experimental results are presented in Table 10.

To better understand the results, the statistics of rankings for IEP SO, EPSO, MP SO, PSO-sono, TCSP SO, SaHSPS, and PPSO are presented in

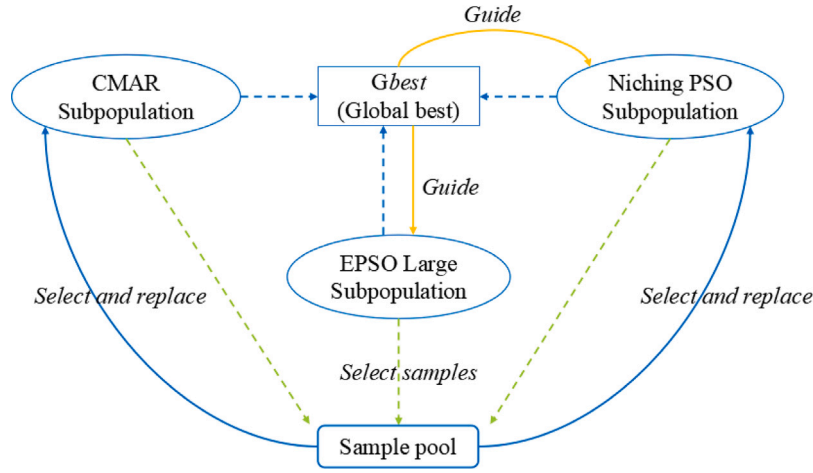


Fig. 2. Overview of sample pool and CMAR, niching-based inertia weight PSO, and EPSO large subpopulations.

Table 3
CEC2005 benchmark functions.

Benchmark functions	Initial range	Search range	$F(x^*)f_{bias}$
Unimodal functions			
f_1 :Shifted Rotated Sphere Function	$[-100, 100]^D$	$[-100, 100]^D$	-450
f_2 :Shifted Schwefel's Problem 1.2	$[-100, 100]^D$	$[-100, 100]^D$	-450
f_3 :Shifted Rotated High Conditioned Elliptic Function	$[-100, 100]^D$	$[-100, 100]^D$	-450
f_4 :Shifted Schwefel's Problem 1.2 with Noise in Fitness	$[-100, 100]^D$	$[-100, 100]^D$	-450
f_5 :Schwefel's Problem 2.6 with Global Optimum on Bounds	$[-100, 100]^D$	$[-100, 100]^D$	-310
Multimodal functions			
f_6 :Shifted Rosenbrock's Function	$[-100, 100]^D$	$[-100, 100]^D$	390
f_7 :Shifted Rotated Griewank's Function without Bounds	$[0, 600]^D$	$[-600, 600]^D$	-180
f_8 :Shifted Rotated Ackley's Function with Global Optimum on Bounds	$[-32, 32]^D$	$[-32, 32]^D$	-140
f_9 :Shifted Rastrigin's Function	$[-5, 5]^D$	$[-5, 5]^D$	-330
f_{10} :Shifted Rotated Rastrigin's Function	$[-5, 5]^D$	$[-5, 5]^D$	-330
f_{11} :Shifted Rotated Weierstrass Function	$[-0.5, 0.5]^D$	$[-0.5, 0.5]^D$	90
f_{12} :Schwefel's Problem 2.13	$[-100, 100]^D$	$[-100, 100]^D$	-460
Expanded functions			
f_{13} :Expanded Extended Griewank's plus Rosenbrock's Function (F8F2)	$[-3, 1]^D$	$[-3, 1]^D$	-130
f_{14} :Shifted Rotated Expanded Scaffer's F6	$[-100, 100]^D$	$[-100, 100]^D$	-300
Hybrid composition functions			
f_{15} :Hybrid Composition Function	$[-5, 5]^D$	$[-5, 5]^D$	120
f_{16} :Rotated Hybrid Composition Function	$[-5, 5]^D$	$[-5, 5]^D$	120
f_{17} :Rotated Hybrid Composition Function with Noise in Fitness	$[-5, 5]^D$	$[-5, 5]^D$	120
f_{18} :Rotated Hybrid Composition Function	$[-5, 5]^D$	$[-5, 5]^D$	10
f_{19} :Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum	$[-5, 5]^D$	$[-5, 5]^D$	10
f_{20} :Rotated Hybrid Composition Function with the Global Optimum on the Bounds	$[-5, 5]^D$	$[-5, 5]^D$	10
f_{21} :Rotated Hybrid Composition Function	$[-5, 5]^D$	$[-5, 5]^D$	360
f_{22} :Rotated Hybrid Composition Function with High Condition Number Matrix	$[-5, 5]^D$	$[-5, 5]^D$	360
f_{23} :Non-Continuous Rotated Hybrid Composition Function	$[-5, 5]^D$	$[-5, 5]^D$	360
f_{24} :Rotated Hybrid Composition Function	$[-5, 5]^D$	$[-5, 5]^D$	260
f_{25} :Rotated Hybrid Composition Function without Bounds	$[-2, 5]^D$	$[-5, 5]^D$	260

Table 9. The 1st, 2nd, and 3rd rankings of the mean, minimum and median values are counted for the 10, 30, and 50-dimensional tests for all compared algorithms. The largest number of rankings of 1st, 2nd, and 3rd are in boldface. The convergence characteristics graphs of IEPHO and the compared PSO variants are presented in Fig. 6. Owing to space limitations, only some of the evolutionary processes of functions over 30 runs of the 50-dimensional test are presented here.

4.3. Comparison and analysis

In the 10-dimensional tests, for the simple unimodal functions, all algorithms perform well on the simple unimodal function f_1 in terms of mean values except for the SaHPSO algorithm, while all algorithms

Table 4
Computational complexity of IEPHO algorithm.

	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2 - T_1)/T_0$
10D	0.0097	190.6024	178.9847	-1197.3282
30D	0.0097	191.4185	183.4455	-821.7034
50D	0.0097	193.3385	186.9754	-655.7959

have equivalent performance in terms of minimum and median values on f_1 . IEPHO has the best mean, minimum, and median values on f_2 - f_4 . For the multimodal functions, IEPHO illustrates the best mean values only on f_8 , whereas PSO-sono and SaHSPS perform best on f_6 and f_7 , respectively. Furthermore, EPSO performs best on f_9 and f_{12}

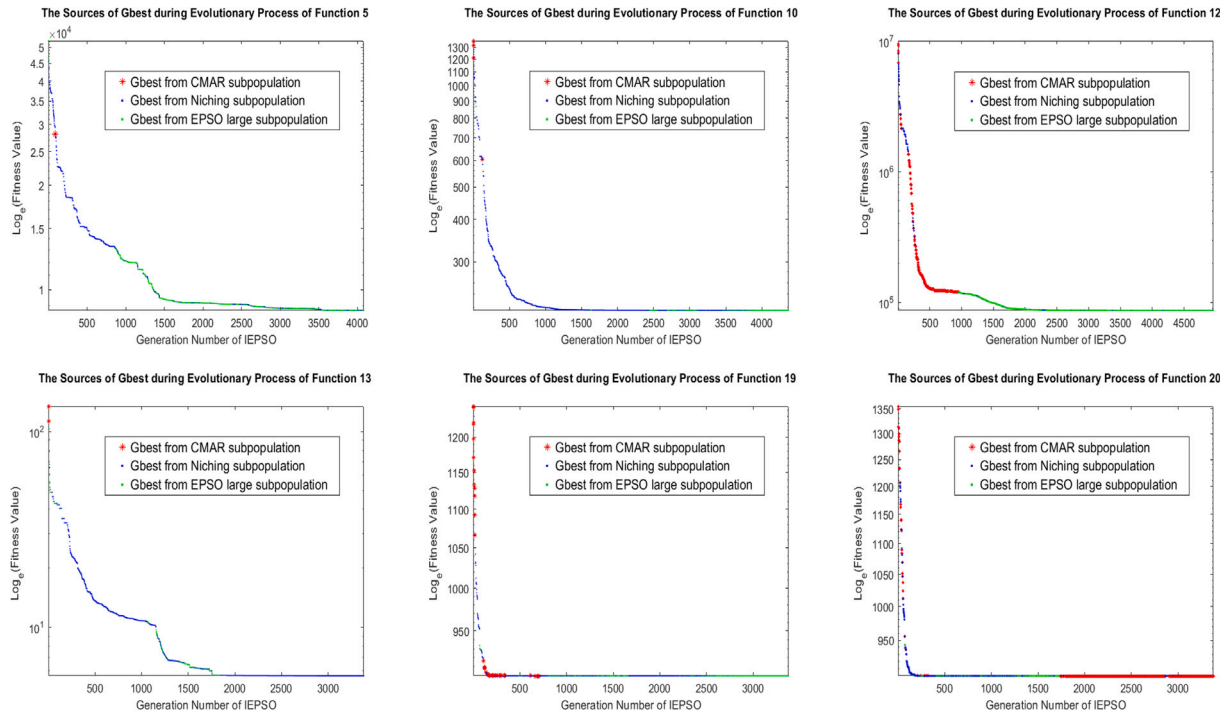


Fig. 3. Sources of Gbest during evolutionary process of single run for 50-dimensional test.

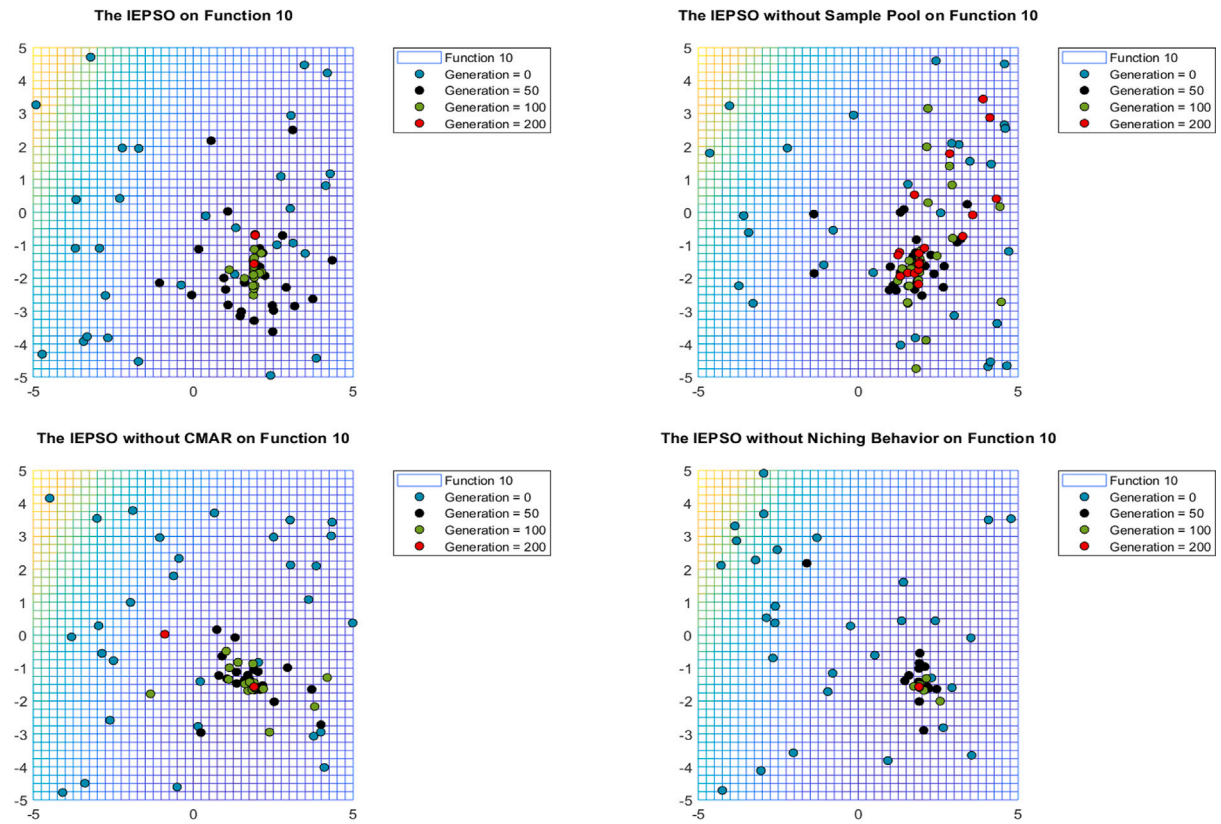


Fig. 4. Plot of 2-dimensional individuals in various generations (0, 50, 100, 200) on function 10.

and PPSO performs best on f_{10} and f_{11} ; however, IEPSO has the best minimum value on f_7 – f_9 , and f_{12} . For expanded functions, SaHSPS and EPSSO have the best mean values on f_{13} and f_{14} , respectively, and only IEPSO has the best minimum value on f_{12} . For hybrid composition functions, IEPSO has the best mean values on f_{19} , but has the best

minimum values on f_{15} and f_{18} – f_{25} . IEPSO has the best median values on f_{19} and f_{23} – f_{25} . In general, IEPSO performs better on unimodal and hybrid composition functions in low-dimensional tests.

In the 30-dimensional tests, for the simple unimodal functions, IEPSO has the best mean values on f_3 – f_5 , the best minimum values

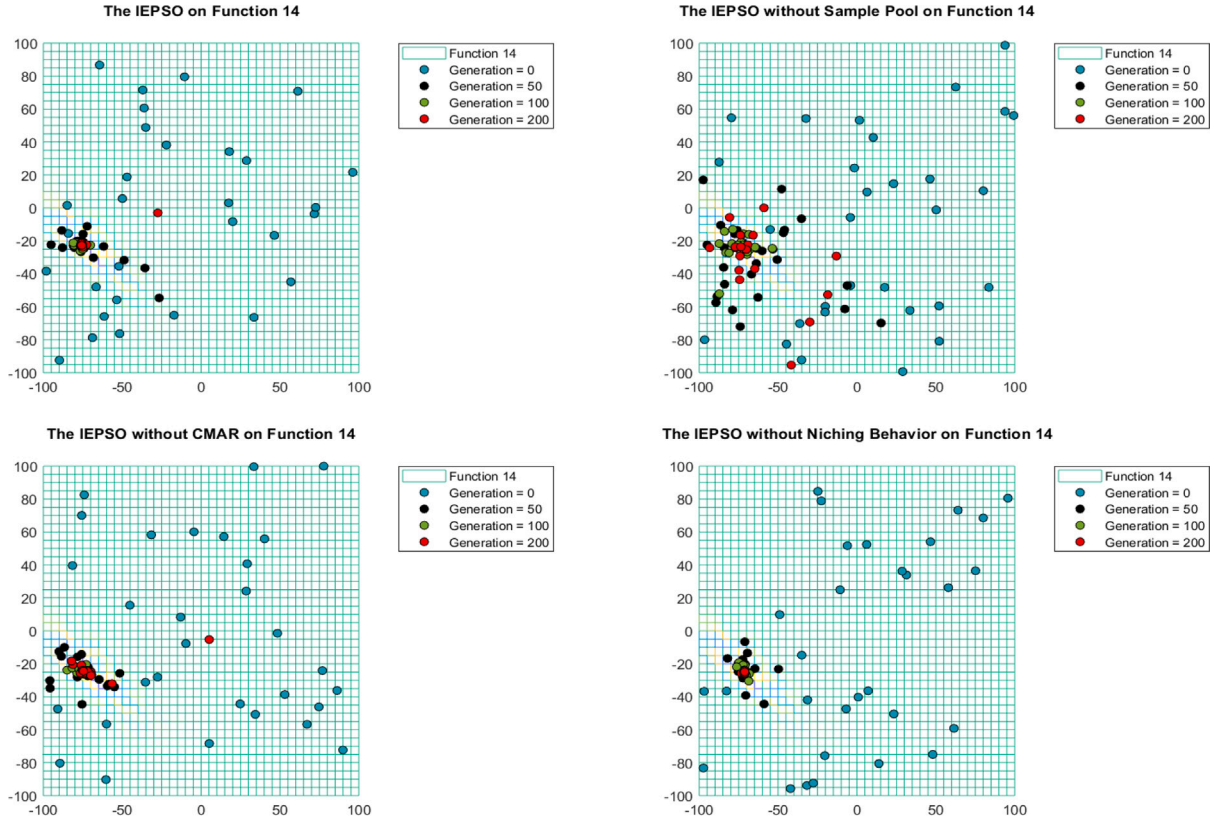


Fig. 5. Plot of 2-dimensional individuals in various generations (0, 50, 100, 200) on function 14.

Table 5

Computational complexity of IEPSO and compared PSO variants on 30 dimensions.

30D	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2 - T_1)/T_0$
IEPSO	0.0097	191.4185	183.4455	-821.7034
EPSO	0.0097	191.4185	219.9203	2938.3258
PSO-sono	0.0097	191.4185	7.6395	-18946.2911
MPSO	0.0097	191.4185	200.4882	935.0158
TCSPSO	0.0097	191.4185	429.6933	24564.4119
SaHSPS	0.0097	191.4185	190.5124	-93.4144
PPSO	0.0097	191.4185	4.6526	-19254.2187

on f_1 - f_4 , and the best median values on f_3 - f_4 ; whereas PSO-sono and PPSO have equivalent best mean values on f_1 and EPSO has the best mean value on f_2 . For the multimodal functions, IEPSO has the best mean values on f_7 and f_{11} - f_{12} , the best minimum values on f_8 and f_{11} , and the best median values on f_7 and f_{11} - f_{12} ; however, EPSO also has the best mean, minimum, and median values on f_6 and f_9 . PPSO has the best mean and median values on f_8 and f_{10} . For expanded functions, EPSO and IEPSO have the best mean, minimum, and median values on f_{13} and f_{14} , respectively. For hybrid composition functions, IEPSO has the best minimum values on f_{21} and f_{23} - f_{25} , the best median values on f_{18} - f_{21} and f_{23} ; EPSO has the best mean values on f_{15} - f_{17} and f_{23} ; PSO-sono has the best mean values on f_{21} and f_{25} ; TCSPSO has the best mean values on f_{18} - f_{20} ; SaHSPS has the best mean values on f_{22} ; and PPSO has the best mean values on f_{24} . Compared with 10-dimensional tests, the performance of IEPSO is significantly improved on multimodal and expanded functions.

In the 50-dimensional tests, for the simple unimodal functions, IEPSO has the best mean, minimum, and median values on f_4 - f_5 and the best minimum values on f_1 and f_3 ; EPSO has the best mean values on f_2 - f_3 ; and PPSO has the best mean values on f_1 . For the multimodal functions, IEPSO has the best mean values on f_7 and f_{12} and the best minimum values on f_7 and f_{11} - f_{12} ; EPSO has the best mean,

minimum, and median values on f_6 and f_9 ; and PPSO and PPSO-sono have the best mean values on f_8 and f_{10} , respectively. For the expanded functions, EPSO and IEPSO have the best mean, minimum, and median values on f_{13} and f_{14} , respectively. For hybrid composition functions, IEPSO has the best mean values on f_{19} - f_{20} and f_{25} , the best minimum values on f_{18} , f_{21} and f_{24} - f_{25} , and the best median values on f_{18} - f_{20} ; EPSO has the best mean values on f_{15} and f_{23} ; PSO-sono has the best mean values on f_{16} - f_{17} , f_{21} , and f_{24} ; TCSPSO has the best mean values on f_{18} - f_{20} ; and SaHSPS has the best mean values on f_{22} . Overall, IEPSO performs better in high-dimensional tests.

In Table 6, IEPSO has better mean values than EPSO for f_2 - f_5 , f_7 - f_8 , and f_{17} - f_{19} ; the mean values for f_1 are equivalent. In Table 7, IEPSO has better mean values than EPSO for f_1 , f_3 - f_5 , f_7 - f_8 , f_{10} - f_{12} , f_{14} , f_{18} - f_{20} , f_{22} , and f_{24} - f_{25} . In Table 8, IEPSO has better mean values than EPSO for f_1 , f_4 - f_5 , f_7 - f_8 , f_{10} , f_{12} , f_{14} , f_{16} - f_{20} , f_{22} , and f_{24} - f_{25} . Overall, if we consider the mean value as the measurement metric, IEPSO outperforms EPSO in high-dimensional tests.

In Table 9, IEPSO has the greatest number of 1st rankings (8 times) for mean values in the 50-dimensional test, and the second greatest number of 1st rankings (6 and 7 times) for mean values in the 10 and 30-dimensional tests. It also has the greatest number of 1st rankings (18, 11, and 12 times) for minimum values in the 10, 30, and 50-dimensional tests, and the greatest number of 1st rankings (9 and 11 times) for median values in the 10 and 30-dimensional tests; it has the second greatest number of 1st rankings (7 times) for median values in the 50-dimensional test. The total number of 1st rankings for the mean, minimum, and median values is 33, 29, and 27 times for the 10, 30, and 50-dimensional tests, respectively, which are the best counted times among all algorithms. These values demonstrate that IEPSO outperforms all tested algorithms. It also has the best performance in terms of the 2nd rankings in the 30 and 50-dimensional tests (22 and 25 times, respectively). The total number of 1st, 2nd, and 3rd rankings of IEPSO is 56 times in the 10-dimensional test, the best for EPSO is 58 times. The total number of 1st, 2nd, and 3rd rankings of IEPSO are 65

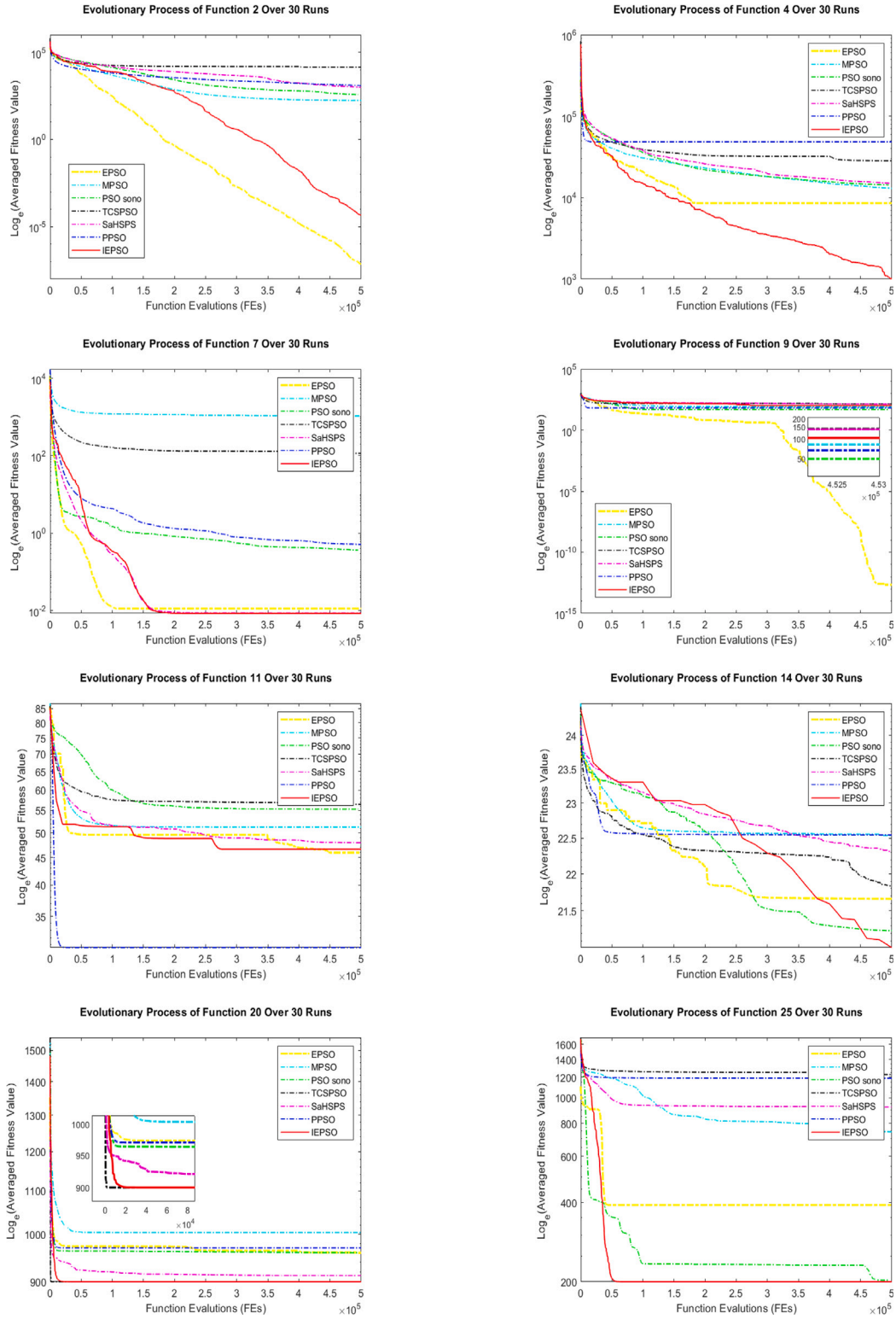


Fig. 6. Evolutionary process of 50D for IEPso, EPSO, MPSO, PSO-sono, TCSPSO, SaHSPS, and PPSO, averaged over 30 runs for f_2 , f_4 , f_7 , f_9 , f_{11} , f_{14} , f_{20} , and f_{25} . On the X-axis are the function evaluations (FEs) and on the Y-axis are the fitness values of the PSO variants.

and 64 times in the 30 and 50-dimensional tests, respectively, which are the best among those of all PSO variants.

In Table 10, IEPso with sample pool performs better than without sample pool on f_1 – f_8 , f_{10} – f_{16} , f_{22} , and f_{24} – f_{25} . The number of best mean values is 18, whereas IEPso without sample pool only has 7. IEPso has better performance than the IEPso without CMAR subpopulation (0%–30%–70%) on f_1 , f_3 , f_5 , f_7 – f_{13} , f_{15} – f_{20} , and f_{23} – f_{24} ; Here,

f_{25} is equivalent, the total number is 19 versus 7, and both include the equivalent function. IEPso outperforms IEPso without niching-based inertia weight PSO subpopulation on f_1 – f_2 , f_4 , f_7 , f_9 , f_{11} , f_{13} – f_{17} , and f_{22} – f_{25} ; the total number is 15 versus 10. The effectiveness of sample pool mechanism is evident from these comparisons. The recommended allocation rates perform better on more benchmark functions and also cover all types of functions.

Table 6

Comparison of experimental results of PSO variants, averaged over 30 runs, for 10-dimensional CEC2005 benchmark functions.

f_n	Criteria	IEPSO	EPSO	MPSO	PSO-sono	TCSPSO	SaHSPS	PPSO
f_1	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.79E-15	0.00E+00
	Rank WST	1	1 =	1 =	1 =	1 =	7 >	1 =
	Min	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Rank	1	1	1	1	1	1	1
	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Rank	1	1	1	1	1	1	1
	(Std Dev)	(0.00E+00)	(0.00E+00)	(0.00E+00)	(0.00E+00)	(0.00E+00)	(1.44E-14)	(0.00E+00)
f_2	Mean	3.79E-15	8.15E-14	1.31E-02	3.79E-15	3.42E-11	5.34E-05	4.30E-11
	Rank WST	1	3 >	7 >	1 =	4 >	6 >	5 >
	Min	0.00E+00	0.00E+00	8.13E-12	0.00E+00	0.00E+00	5.68E-14	1.71E-13
	Rank	1	1	7	1	1	5	6
	Median	0.00E+00	5.68E-14	5.54E-10	0.00E+00	5.68E-14	4.37E-08	2.81E-12
	Rank	1	3	6	1	3	7	5
	(Std Dev)	(1.44E-14)	(4.65E-14)	(7.15E-02)	(1.44E-14)	(1.18E-10)	(1.30E-04)	(8.83E-11)
f_3	Mean	4.97E+02	5.36E+04	3.72E+04	8.93E+04	4.30E+05	2.12E+05	7.72E+04
	Rank WST	1	3 >	2 >	5 >	7 >	6 >	4 >
	Min	5.04E+00	3.07E+03	1.63E+03	5.76E+01	3.10E+03	5.73E+03	1.84E+04
	Rank	1	4	3	2	5	6	7
	Median	3.16E+02	4.57E+04	2.93E+04	1.08E+04	1.97E+05	1.41E+05	6.20E+04
	Rank	1	4	3	2	7	6	5
	(Std Dev)	(7.96E+02)	(4.11E+04)	(3.29E+04)	(2.02E+05)	(5.01E+05)	(2.28E+05)	(5.63E+04)
f_4	Mean	4.17E-14	3.77E-09	1.09E-05	2.26E-08	2.94E-03	4.67E+00	2.21E+02
	Rank WST	1	2 >	4 >	3 >	5 >	6 >	7 >
	Min	0.00E+00	1.14E-13	1.08E-09	0.00E+00	9.21E-12	1.14E-13	9.18E-01
	Rank	1	3	6	1	5	3	7
	Median	5.68E-14	6.14E-10	1.04E-06	6.53E-11	1.31E-08	1.32E-04	1.29E+02
	Rank	1	3	5	2	4	6	7
	(Std Dev)	(3.93E-14)	(7.62E-09)	(2.54E-05)	(6.83E-08)	(1.45E-02)	(2.55E+01)	(2.62E+02)
f_5	Mean	3.19E-11	1.42E-03	6.98E-10	9.01E+02	2.18E+03	6.67E-13	1.78E-01
	Rank WST	2	4 >	3 >	6 >	7 >	1 <	5 ≥
	Min	7.28E-12	2.10E-09	0.00E+00	6.50E+02	7.89E+02	0.00E+00	0.00E+00
	Rank	4	5	1	6	7	1	1
	Median	2.09E-11	7.51E-06	0.00E+00	8.81E+02	1.91E+03	0.00E+00	0.00E+00
	Rank	4	5	1	6	7	1	1
	(Std Dev)	(2.56E-11)	(3.31E-03)	(2.58E-09)	(1.85E+02)	(1.17E+03)	(2.55E-12)	(9.12E-01)
f_6	Mean	1.84E+00	1.04E+00	2.55E+00	1.34E-01	1.65E+01	5.61E+01	1.69E+01
	Rank WST	3	2 <	4 >	1 <	5 >	7 >	6 >
	Min	5.24E-05	1.67E-05	3.32E-02	0.00E+00	2.43E-01	2.20E-01	3.31E-01
	Rank	3	2	4	1	6	5	7
	Median	1.18E+00	9.50E-02	1.93E+00	9.32E-12	3.55E+00	4.09E+00	4.94E+00
	Rank	3	2	4	1	5	6	7
	(Std Dev)	(2.23E+00)	(2.67E+00)	(2.62E+00)	(7.28E-01)	(4.01E+01)	(2.07E+02)	(6.15E+01)
f_7	Mean	1.59E-01	2.21E-01	3.53E-01	9.48E-02	5.33E-01	6.80E-02	1.31E-01
	Rank WST	4	5 >	6 >	2 <	7 >	1 <	3 ≤
	Min	0.00E+00	7.38E-02	5.18E-02	9.86E-03	4.92E-02	1.65E-02	9.86E-03
	Rank	1	7	6	2	5	4	3
	Median	1.07E-01	2.12E-01	2.86E-01	5.17E-02	4.29E-01	5.70E-02	8.00E-02
	Rank	4	5	6	1	7	2	3
	(Std Dev)	(1.34E-01)	(1.08E-01)	(3.11E-01)	(9.00E-02)	(3.95E-01)	(5.17E-02)	(1.09E-01)
f_8	Mean	2.00E+01	2.03E+01	2.03E+01	2.03E+01	2.03E+01	2.03E+01	2.02E+01
	Rank WST	1	7 >	6 >	5 >	4 >	3 >	2 >
	Min	2.00E+01	2.02E+01	2.01E+01	2.01E+01	2.01E+01	2.01E+01	2.00E+01
	Rank	1	7	6	3	4	5	2
	Median	2.00E+01	2.03E+01	2.03E+01	2.03E+01	2.03E+01	2.03E+01	2.02E+01
	Rank	1	7	6	5	4	3	2
	(Std Dev)	(0.00E+00)	(6.87E-02)	(7.52E-02)	(8.63E-02)	(7.50E-02)	(7.62E-02)	(1.12E-01)
f_9	Mean	2.06E+00	0.00E+00	6.17E+00	5.07E+00	4.22E+00	3.30E+00	3.55E+00
	Rank WST	2	1 <	7 >	6 >	5 >	3 >	4 >
	Min	0.00E+00	0.00E+00	0.00E+00	9.95E-01	9.95E-01	9.95E-01	9.95E-01
	Rank	1	1	1	4	4	7	4
	Median	1.99E+00	0.00E+00	5.47E+00	4.97E+00	2.99E+00	2.98E+00	2.98E+00
	Rank	2	1	7	6	5	4	3
	(Std Dev)	(1.73E+00)	(0.00E+00)	(3.18E+00)	(2.27E+00)	(2.66E+00)	(1.55E+00)	(1.22E+00)
f_{10}	Mean	1.03E+01	7.66E+00	1.26E+01	7.06E+00	2.14E+01	1.08E+01	3.08E+00
	Rank WST	4	3 <	6 >	2 <	7 >	5 >	1 <
	Min	4.97E+00	2.98E+00	1.99E+00	1.99E+00	9.95E+00	3.98E+00	9.95E-01
	Rank	6	4	3	2	7	5	1
	Median	9.95E+00	7.10E+00	1.24E+01	6.96E+00	2.19E+01	9.95E+00	2.98E+00
	Rank	4	3	6	2	7	5	1
	(Std Dev)	(4.27E+00)	(3.12E+00)	(6.06E+00)	(3.20E+00)	(7.65E+00)	(4.59E+00)	(1.37E+00)

(continued on next page)

The results show that the proposed strategy performs exceptionally in high-dimensional (30D and 50D) tests on CEC2005 benchmark functions, compared with most recent PSO variants. Although the total

number of top 1, 2, and 3 rankings of IEPSO is 56 times in 10-dimensional test, which is less than that of EPSO, the total top 1 rankings for the mean, minimum, and median values of IEPSO is 33

Table 6 (continued).

f_n	Criteria	IEPSO	EPSO	MPSO	PSO-sono	TCSPSO	SaHSPS	PPSO
f_{11}	Mean	4.01E+00	3.31E+00	4.04E+00	4.25E+00	4.89E+00	4.99E+00	2.49E+00
	Rank WST	3	2 $\leq$	4 $\geq$	5 $\geq$	6 $>$	7 $>$	1 $<$
	Min	1.52E+00	1.63E+00	1.48E+00	2.00E+00	8.73E-01	1.21E+00	2.95E-01
	Rank	5	6	4	7	2	3	1
	Median	3.92E+00	3.31E+00	3.56E+00	4.50E+00	5.24E+00	4.90E+00	2.39E+00
	Rank	4	2	3	5	7	6	1
	(Std Dev)	(1.41E+00)	(1.00E+00)	(1.57E+00)	(1.26E+00)	(1.73E+00)	(1.65E+00)	(9.38E-01)
f_{12}	Mean	1.07E+02	2.48E+01	1.04E+03	4.48E+01	2.01E+02	3.88E+02	6.06E+02
	Rank WST	3	1 $\leq$	7 $>$	2 $\leq$	4 $>$	5 $>$	6 $>$
	Min	0.00E+00	3.74E-04	3.19E-02	0.00E+00	2.53E-01	1.41E+00	3.11E-02
	Rank	1	3	5	1	6	7	4
	Median	1.00E+01	1.00E+01	1.67E+01	9.52E+00	1.97E+01	3.55E+01	1.60E+01
	Rank	2	3	5	1	6	7	4
	(Std Dev)	(3.11E+02)	(9.06E+01)	(1.68E+03)	(8.48E+01)	(4.24E+02)	(6.30E+02)	(1.29E+03)
f_{13}	Mean	7.99E-01	5.41E-01	7.91E-01	7.86E-01	6.50E-01	4.36E-01	1.21E+00
	Rank WST	6	2 $<$	5 $\leq$	4 $\leq$	3 $\leq$	1 $<$	7 $>$
	Min	3.66E-01	2.76E-01	2.89E-01	3.10E-01	9.88E-03	6.21E-02	4.91E-01
	Rank	6	3	4	5	1	2	7
	Median	6.41E-01	5.35E-01	8.14E-01	7.41E-01	6.38E-01	4.65E-01	1.24E+00
	Rank	4	2	6	5	3	1	7
	(Std Dev)	(4.12E-01)	(1.33E-01)	(3.14E-01)	(3.34E-01)	(2.53E-01)	(2.25E-01)	(3.77E-01)
f_{14}	Mean	2.46E+00	2.37E+00	3.29E+00	2.56E+00	2.95E+00	3.17E+00	3.49E+00
	Rank WST	2	1 $\leq$	6 $>$	3 $>$	4 $>$	5 $>$	7 $>$
	Min	1.03E+00	1.18E+00	2.52E+00	1.80E+00	2.20E+00	2.36E+00	2.72E+00
	Rank	1	2	6	3	4	5	7
	Median	2.53E+00	2.33E+00	3.15E+00	2.64E+00	3.01E+00	3.21E+00	3.52E+00
	Rank	2	1	5	3	4	6	7
	(Std Dev)	(5.43E-01)	(5.03E-01)	(3.70E-01)	(4.39E-01)	(3.48E-01)	(3.55E-01)	(3.26E-01)
f_{15}	Mean	2.13E+02	3.10E+01	3.09E+02	3.42E+02	2.69E+02	3.30E+02	3.07E+02
	Rank WST	2	1 $<$	5 $>$	7 $>$	3 $>$	6 $>$	4 $>$
	Min	0.00E+00	0.00E+00	1.247E-01	7.00E+01	4.15E+01	4.15E+01	0.00E+00
	Rank	1	1	4	7	5	6	1
	Median	1.02E+02	0.00E+00	4.06E+02	4.00E+02	2.50E+02	4.06E+02	4.00E+02
	Rank	2	1	6	4	3	7	4
	(Std Dev)	(1.68E+02)	(8.20E+01)	(1.60E+02)	(1.19E+02)	(1.85E+02)	(1.42E+02)	(1.31E+02)
f_{16}	Mean	1.17E+02	1.11E+02	1.24E+02	1.10E+02	1.42E+02	1.22E+02	9.90E+01
	Rank WST	4	3 $<$	6 $>$	2 $<$	7 $>$	5 $>$	1 $<$
	Min	1.03E+02	9.43E+01	1.02E+02	8.96E+01	1.02E+02	9.74E+01	9.10E+01
	Rank	7	3	5	1	6	4	2
	Median	1.13E+02	1.11E+02	1.21E+02	1.07E+02	1.41E+02	1.14E+02	9.80E+01
	Rank	4	3	6	2	7	5	1
	(Std Dev)	(1.06E+01)	(9.39E+00)	(1.70E+01)	(1.03E+01)	(2.80E+01)	(1.91E+01)	(5.65E+00)
f_{17}	Mean	1.12E+02	1.12E+02	1.25E+02	1.11E+02	1.43E+02	1.22E+02	9.97E+01
	Rank WST	3	4 $>$	6 $>$	2 $\leq$	7 $>$	5 $>$	1 $<$
	Min	7.68E+01	6.18E+01	9.71E+01	9.22E+01	1.07E+02	9.21E+01	9.18E+01
	Rank	2	1	6	5	7	4	3
	Median	1.12E+02	1.13E+02	1.26E+02	1.12E+02	1.37E+02	1.21E+02	9.93E+01
	Rank	2	4	6	3	7	5	1
	(Std Dev)	(1.24E+01)	(1.37E+01)	(1.43E+01)	(1.05E+01)	(2.82E+01)	(1.65E+01)	(4.09E+00)
f_{18}	Mean	5.38E+02	6.14E+02	9.07E+02	5.19E+02	8.94E+02	6.19E+02	9.26E+02
	Rank WST	2	3 $>$	6 $>$	1 $\leq$	5 $>$	4 $>$	7 $>$
	Min	3.00E+02	3.00E+02	3.00E+02	3.00E+02	8.00E+02	3.00E+02	8.00E+02
	Rank	1	1	1	1	7	1	6
	Median	3.87E+02	5.95E+02	9.83E+02	3.00E+02	9.00E+02	7.06E+02	9.53E+02
	Rank	2	3	7	1	5	4	6
	(Std Dev)	(2.49E+02)	(1.92E+02)	(1.94E+02)	(2.97E+02)	(2.21E+01)	(2.96E+02)	(6.80E+01)
f_{19}	Mean	547.386484719	662.626291322	933.226485047	707.479077324	900.000000000	708.386684871	959.707483614
	Rank WST	1	2 $>$	6 $>$	3 $\leq$	5 $>$	4 $>$	7 $>$
	Min	300.000000000	347.537614346	404.242818873	300.000000000	900.000000000	300.000000000	800.000014336
	Rank	1	4	5	1	7	1	6
	Median	403.849937393	800.000000003	976.757727894	800.000000000	900.000000000	800.008477432	990.032633731
	Rank	1	3	6	2	5	4	7
	(Std Dev)	(2.50E+02)	(1.81E+02)	(1.43E+02)	(2.75E+02)	(0.00E+00)	(2.98E+02)	(7.04E+01)
f_{20}	Mean	653.124840961	636.255824427	934.037269068	527.389711654	900.000000000	604.318138272	930.588433735
	Rank WST	4	3 $\leq$	7 $>$	1 $\leq$	5 $>$	2 $\leq$	6 $>$
	Min	300.000000000	300.000000000	622.337979195	300.000000000	900.000000000	300.000000000	752.716571757
	Rank	1	1	5	1	7	1	6
	Median	800.000000000	800.000000003	982.865408546	300.000000000	900.000000000	474.732994115	969.045928962
	Rank	3	4	7	1	5	2	6
	(Std Dev)	(2.21E+02)	(1.96E+02)	(1.10E+02)	(2.78E+02)	(0.00E+00)	(3.00E+02)	(8.74E+01)

(continued on next page)

times, which is the highest. The proposed strategy exhibits significant improvements over EPSO, and the best test performance in most cases.

We performed the 2-dimensional tests for IEPSO and IEPSO without individual mechanisms. The 2-dimensional individuals in various generations (0, 50, 100, 200) on f_{10} and f_{14} are plotted in Figs. 4 and

5. From the figures, it can be observed that the individuals are more scattered during the evolutionary process when IEPSO does not use the sample pool; information sharing mechanism makes the particles smarter and more targeted. The IEPSO without niching behavior leads to more gathering of individuals due to loss of population diversity,

Table 6 (continued).

f_n	Criteria	IEPSO	EPSO	MPSO	PSO-sono	TCSPSO	SaHSPS	PPSO
f_{21}	Mean	7.01E+02	4.32E+02	1.03E+03	6.34E+02	7.64E+02	5.17E+02	1.05E+03
	Rank WST	4	1 <	6 >	3 ≤	5 ≥	2 <	7 >
	Min	3.00E+02	3.00E+02	5.00E+02	5.00E+02	3.00E+02	3.00E+02	5.00E+02
	Rank	1	1	5	5	1	1	5
	Median	6.50E+02	3.00E+02	1.08E+03	5.00E+02	7.44E+02	5.00E+02	1.14E+03
	Rank	4	1	6	2	5	2	7
	(Std Dev)	(2.90E+02)	(1.86E+02)	(1.84E+02)	(2.27E+02)	(3.86E+02)	(2.16E+02)	(2.09E+02)
f_{22}	Mean	672.791483663637	664.302759603500	776.016771584684	781.357987190608	768.928571053127	701.517953711591	783.269908262144
	Rank WST	2	1 ≤	5 >	6 >	4 >	3 >	7 >
	Min	300.000000000000	300.000000000002	732.977627933257	730.408328969641	300.000000000002	300.000000000000	724.494658916934
	Rank	1	3	7	6	3	1	5
	Median	760.849372206458	753.816852089834	773.007516355743	757.992872991054	771.017337359072	756.462103194073	752.834288279931
	Rank	5	2	7	4	6	3	1
	(Std Dev)	(1.90E+02)	(1.85E+02)	(3.30E+01)	(4.09E+01)	(1.02E+02)	(1.86E+02)	(5.85E+01)
f_{23}	Mean	701.659559849291	622.018278048935	1143.95675777379	824.341970895486	880.434164547717	619.220207766860	1054.62849248215
	Rank WST	3	2 <	7 >	4 >	5 >	1 <	6 >
	Min	547.625456262461	547.662737240993	721.222275493949	559.468310791340	559.468310791336	559.468310791324	559.468310791339
	Rank	1	2	7	6	4	3	5
	Median	559.468310791336	559.468310791336	1194.39713977751	764.985707098758	818.170061470840	559.468310791336	1123.10964566369
	Rank	1	1	7	4	5	1	6
	(Std Dev)	(1.81E+02)	(1.08E+02)	(1.30E+02)	(2.82E+02)	(2.58E+02)	(1.16E+02)	(2.45E+02)
f_{24}	Mean	250.1465793193	200.0000000002	362.3410179098	200.0000000000	326.6666666920	341.5393444070	379.9267342138
	Rank WST	3	2 ≤	6 >	1 ≤	4 >	5 >	7 >
	Min	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02
	Rank	1	1	1	1	1	1	1
	Median	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02
	(Std Dev)	(1.14E+02)	(1.14E+02)	(2.34E+02)	(0.00E+00)	(2.05E+02)	(2.39E+02)	(2.93E+02)
f_{25}	Mean	2.30E+02	2.00E+02	3.32E+02	2.20E+02	2.93E+02	3.14E+02	4.89E+02
	Rank WST	3	1 ≤	6 ≥	2 ≤	4 ≥	5 ≥	7 >
	Min	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02
	Rank	1	1	1	1	1	1	1
	Median	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02
	Rank	1	1	1	1	1	1	1
	(Std Dev)	(9.15E+01)	(0.00E+00)	(2.76E+02)	(7.61E+01)	(2.12E+02)	(2.16E+02)	(3.94E+02)

Table 7

Comparison of experimental results of PSO variants, averaged over 30 runs, for 30-dimensional CEC2005 benchmark functions.

f_n	Criteria	IEPSO	EPSO	MPSO	PSO-sono	TCSPSO	SaHSPS	PPSO
f_1	Mean	4.55E-14	6.06E-14	3.22E-14	0.00E+00	5.68E-14	5.31E-14	0.00E+00
	Rank WST	4	7 >	3 <	1 <	6 >	5 >	1 <
	Min	0.00E+00	5.68E-14	0.00E+00	0.00E+00	5.68E-14	0.00E+00	0.00E+00
	Rank	1	6	1	1	6	1	1
	Median	5.68E-14	5.68E-14	5.68E-14	0.00E+00	5.68E-14	5.68E-14	0.00E+00
	Rank	3	3	3	1	3	3	1
	(Std Dev)	(2.31E-14)	(1.44E-14)	(2.86E-14)	(0.00E+00)	(6.42E-30)	(1.44E-14)	(0.00E+00)
f_2	Mean	2.54E-10	2.76E-12	4.72E-01	7.21E-02	1.67E+03	1.63E+01	1.02E+01
	Rank WST	2	1 <	4 >	3 >	7 >	6 >	5 >
	Min	5.12E-13	6.82E-13	5.49E-02	1.96E-05	2.40E+01	1.96E-04	4.05E-01
	Rank	1	2	5	3	7	4	6
	Median	1.10E-11	2.30E-12	2.73E-01	5.02E-03	1.27E+03	7.39E-01	4.88E+00
	Rank	2	1	4	3	7	5	6
	(Std Dev)	(8.06E-10)	(1.86E-12)	(6.45E-01)	(1.82E-01)	(1.70E+03)	(4.04E+01)	(1.59E+01)
f_3	Mean	2.87E+05	3.18E+05	4.89E+06	8.21E+06	2.33E+07	1.23E+06	1.09E+06
	Rank WST	1	2 >	5 >	6 >	7 >	4 >	3 >
	Min	6.07E+03	7.67E+04	4.06E+05	1.57E+05	3.89E+06	3.38E+05	3.75E+05
	Rank	1	2	6	3	7	4	5
	Median	1.98E+04	2.90E+05	1.99E+06	4.26E+06	1.67E+07	7.00E+05	1.03E+06
	Rank	1	2	5	6	7	3	4
	(Std Dev)	(5.52E+05)	(1.56E+05)	(6.73E+06)	(8.32E+06)	(1.98E+07)	(1.17E+06)	(4.36E+05)
f_4	Mean	1.75E+00	2.57E+02	6.22E+02	1.12E+03	4.17E+03	6.62E+02	2.06E+04
	Rank WST	1	2 >	3 >	5 >	6 >	4 >	7 >
	Min	3.99E-02	6.91E+01	1.15E+02	1.14E+02	1.23E+03	4.41E+01	1.18E+04
	Rank	1	3	5	4	6	2	7
	Median	5.04E-01	2.00E+02	4.90E+02	7.36E+02	3.85E+03	3.06E+02	1.93E+04
	Rank	1	2	4	5	6	3	7
	(Std Dev)	(3.75E+00)	(1.84E+02)	(4.20E+02)	(1.06E+03)	(2.09E+03)	(8.64E+02)	(5.91E+03)
f_5	Mean	3.529E+03	4.602E+03	3.844E+03	4.728E+03	9.220E+03	3.942E+03	4.734E+03
	Rank WST	1	4 >	2 >	5 >	7 >	3 >	6 >
	Min	1.716E+03	3.078E+03	2.406E+03	3.499E+03	4.851E+03	6.223E+02	2.695E+03
	Rank	2	5	3	6	7	1	4
	Median	3.637E+03	4.307E+03	3.680E+03	4.504E+03	8.959E+03	3.431E+03	4.973E+03
	Rank	2	4	3	5	7	1	6
	(Std Dev)	(1.01E+03)	(9.86E+02)	(8.32E+02)	(8.68E+02)	(1.98E+03)	(3.00E+03)	(1.10E+03)

(continued on next page)

Table 7 (continued).

f_n	Criteria	IEPSO	EPSO	MPSO	PSO-sono	TCSPSO	SaHSPS	PPSO
f_6	Mean	4.37E+01	3.83E+00	2.93E+06	1.44E+01	6.65E+01	5.99E+02	1.85E+02
	Rank WST	3	1 <	7 >	2 <	4 >	6 >	5 >
	Min	2.76E-01	1.31E-04	1.80E+02	1.05E-01	1.74E+00	4.83E+00	4.57E+00
	Rank	3	1	7	2	4	6	5
	Median	2.16E+01	3.13E+00	1.42E+05	1.26E+01	3.50E+01	3.79E+01	5.30E+01
	Rank	3	1	7	2	4	5	6
	(Std Dev)	(5.86E+01)	(6.18E+00)	(6.09E+06)	(1.22E+01)	(6.71E+01)	(2.85E+03)	(3.12E+02)
f_7	Mean	1.70E-02	2.11E-02	7.66E-01	1.71E+00	4.48E+01	2.03E-02	6.42E-02
	Rank WST	1	3 ≥	5 >	6 >	7 >	2 ≥	4 >
	Min	2.27E-13	2.84E-13	1.42E-03	5.68E-14	1.73E+00	3.56E-08	5.43E-08
	Rank	2	3	6	1	7	4	5
	Median	1.11E-02	1.48E-02	2.62E-02	1.23E+00	3.76E+01	1.73E-02	1.23E-02
	Rank	1	3	5	6	7	4	2
	(Std Dev)	(1.40E-02)	(2.13E-02)	(3.69E+00)	(2.20E+00)	(3.55E+01)	(1.60E-02)	(2.04E-01)
f_8	Mean	2.049E+01	2.093E+01	2.089E+01	2.091E+01	2.088E+01	2.084E+01	2.018E+01
	Rank WST	2	7 >	5 >	6 >	4 >	3 >	1 <
	Min	2.000E+01	2.080E+01	2.066E+01	2.078E+01	2.071E+01	2.072E+01	2.004E+01
	Rank	1	7	3	6	4	5	2
	Median	2.047E+01	2.094E+01	2.089E+01	2.092E+01	2.088E+01	2.083E+01	2.018E+01
	Rank	2	7	5	6	4	3	1
	(Std Dev)	(4.45E-01)	(5.27E-02)	(9.50E-02)	(6.41E-02)	(6.20E-02)	(6.64E-02)	(7.10E-02)
f_9	Mean	2.02E+01	1.10E-13	4.34E+01	2.23E+01	6.05E+01	5.24E+01	2.57E+01
	Rank WST	2	1 <	5 >	3 >	7 >	6 >	4 >
	Min	5.97E+00	5.68E-14	9.01E+00	7.96E+00	2.12E+01	1.99E+01	1.39E+01
	Rank	2	1	4	3	7	6	5
	Median	1.84E+01	1.14E-13	4.25E+01	2.04E+01	5.97E+01	5.20E+01	2.64E+01
	Rank	2	1	5	3	7	6	4
	(Std Dev)	(8.61E+00)	(3.93E-14)	(1.49E+01)	(8.97E+00)	(1.89E+01)	(1.74E+01)	(5.33E+00)
f_{10}	Mean	5.49E+01	5.85E+01	8.60E+01	3.66E+01	1.35E+02	8.88E+01	2.74E+01
	Rank WST	3	4 >	5 >	2 ≤	7 >	6 >	1 <
	Min	1.98E+01	3.28E+01	4.08E+01	1.39E+01	7.46E+01	4.48E+01	1.49E+01
	Rank	3	4	5	1	7	6	2
	Median	3.73E+01	5.48E+01	8.41E+01	3.58E+01	1.30E+02	8.82E+01	2.64E+01
	Rank	3	4	5	2	7	6	1
	(Std Dev)	(4.49E+01)	(1.77E+01)	(2.73E+01)	(1.28E+01)	(3.75E+01)	(2.42E+01)	(8.47E+00)
f_{11}	Mean	1.31E+01	2.26E+01	2.47E+01	2.80E+01	2.77E+01	2.34E+01	1.42E+01
	Rank WST	1	3 >	5 >	7 >	6 >	4 >	2 >
	Min	5.59E+00	1.52E+01	1.35E+01	1.98E+01	1.99E+01	1.55E+01	1.05E+01
	Rank	1	4	3	6	7	5	2
	Median	1.29E+01	2.28E+01	2.40E+01	2.78E+01	2.80E+01	2.42E+01	1.38E+01
	Rank	1	3	4	6	7	5	2
	(Std Dev)	(4.53E+00)	(2.87E+00)	(4.31E+00)	(2.94E+00)	(3.39E+00)	(4.14E+00)	(2.14E+00)
f_{12}	Mean	3.98E+03	4.24E+03	5.35E+04	1.32E+04	2.57E+04	4.44E+04	7.97E+03
	Rank WST	1	2 ≥	7 >	4 >	5 >	6 >	3 >
	Min	2.17E+02	4.86E+01	6.40E+03	1.47E+02	4.93E+03	1.41E+03	7.91E+01
	Rank	4	1	7	3	6	5	2
	Median	2.75E+03	2.98E+03	4.93E+04	8.83E+03	1.79E+04	2.06E+04	7.51E+03
	Rank	1	2	7	4	5	6	3
	(Std Dev)	(4.06E+03)	(3.95E+03)	(2.87E+04)	(1.35E+04)	(2.64E+04)	(5.11E+04)	(5.63E+03)
f_{13}	Mean	2.54E+00	1.88E+00	2.73E+00	3.38E+00	3.31E+00	3.53E+00	4.51E+00
	Rank WST	2	1 <	3 >	5 >	4 >	6 >	7 >
	Min	1.44E+00	1.20E+00	1.32E+00	1.59E+00	2.08E+00	2.03E+00	2.78E+00
	Rank	3	1	2	4	6	5	7
	Median	2.61E+00	1.89E+00	2.71E+00	3.34E+00	3.14E+00	3.47E+00	4.51E+00
	Rank	2	1	3	5	4	6	7
	(Std Dev)	(5.40E-01)	(3.65E-01)	(6.84E-01)	(1.13E+00)	(7.71E-01)	(9.45E-01)	(9.67E-01)
f_{14}	Mean	1.18E+01	1.21E+01	1.28E+01	1.20E+01	1.22E+01	1.27E+01	1.32E+01
	Rank WST	1	3 >	6 >	2 ≥	4 >	5 >	7 >
	Min	1.01E+01	1.09E+01	1.17E+01	1.03E+01	1.11E+01	1.14E+01	1.18E+01
	Rank	1	3	6	2	4	5	7
	Median	1.20E+01	1.22E+01	1.29E+01	1.21E+01	1.23E+01	1.28E+01	1.32E+01
	Rank	1	3	6	2	4	5	7
	(Std Dev)	(6.99E-01)	(4.88E-01)	(3.87E-01)	(5.64E-01)	(4.33E-01)	(4.10E-01)	(3.77E-01)
f_{15}	Mean	3.478E+02	5.662E+01	3.174E+02	3.067E+02	3.677E+02	3.929E+02	3.464E+02
	Rank WST	5	1 <	3 ≤	2 <	6 >	7 >	4 ≤
	Min	1.151E+02	0.000E+00	4.305E+01	2.000E+02	2.179E+02	2.214E+02	3.000E+02
	Rank	3	1	2	4	5	6	7
	Median	3.226E+02	4.679E+01	3.025E+02	3.003E+02	3.238E+02	4.000E+02	3.235E+02
	Rank	4	1	3	2	6	7	5
	(Std Dev)	(1.01E+02)	(6.81E+01)	(1.28E+02)	(8.10E+01)	(8.70E+01)	(7.60E+01)	(5.72E+01)

(continued on next page)

while using niching behavior can maintain the population diversity even in the later generations of the run. The individuals are more scattered in the later generations for IEPSO without CMAR as a local search

method. The figures show strong evidence that the proposed mechanisms are effective. Thus, the exploration and exploitation abilities of the proposed IEPSO have been greatly improved.

Table 7 (continued).

f_n	Criteria	IEPSO	EPSO	MPSO	PSO-sono	TCSPSO	SaHSPS	PPSO
f_{16}	Mean	1.54E+02	1.16E+02	2.35E+02	1.41E+02	2.98E+02	2.07E+02	2.16E+02
	Rank WST	3	1 ≤	6 >	2 ≤	7 >	4 >	5 >
	Min	5.13E+01	4.35E+01	6.54E+01	4.44E+01	1.01E+02	6.21E+01	3.95E+01
	Rank	4	2	4	3	7	5	1
	Median	9.43E+01	1.21E+02	1.68E+02	7.91E+01	2.75E+02	1.30E+02	1.13E+02
	Rank	2	4	6	1	7	5	3
	(Std Dev)	(1.27E+02)	(4.65E+01)	(1.55E+02)	(1.33E+02)	(1.26E+02)	(1.48E+02)	(1.90E+02)
f_{17}	Mean	1.532E+02	1.252E+02	3.098E+02	1.769E+02	3.067E+02	2.419E+02	3.104E+02
	Rank WST	2	1 ≤	6 >	3 ≥	5 >	4 >	7 >
	Min	5.373E+01	5.244E+01	7.147E+01	4.386E+01	1.194E+02	8.046E+01	5.638E+01
	Rank	3	2	5	1	7	6	4
	Median	9.466E+01	1.181E+02	3.105E+02	7.931E+01	2.529E+02	1.487E+02	3.351E+02
	Rank	2	3	6	1	5	4	7
	(Std Dev)	(1.24E+02)	(4.72E+01)	(1.99E+02)	(1.71E+02)	(1.54E+02)	(1.61E+02)	(2.14E+02)
f_{18}	Mean	9.009172E+02	9.085728E+02	9.390933E+02	9.271146E+02	9.000000E+02	9.112815E+02	9.183807E+02
	Rank WST	2	3 >	7 >	6 >	1 =	4 >	5 >
	Min	9.000000E+02	8.000000E+02	9.000001E+02	9.197070E+02	9.000000E+02	9.000002E+02	9.080982E+02
	Rank	2	1	4	7	2	5	6
	Median	9.000000E+02	9.156903E+02	9.299443E+02	9.264305E+02	9.000000E+02	9.079635E+02	9.173780E+02
	Rank	1	4	7	6	1	3	5
	(Std Dev)	(3.51E+00)	(2.97E+01)	(3.23E+01)	(5.05E+00)	(0.00E+00)	(1.49E+01)	(6.61E+00)
f_{19}	Mean	9.01E+02	9.10E+02	9.29E+02	9.27E+02	9.00E+02	9.08E+02	9.19E+02
	Rank WST	2	4 >	7 >	6 >	1 =	3 >	5 >
	Min	9.00E+02	8.00E+02	8.00E+02	9.22E+02	9.00E+02	9.05E+02	9.11E+02
	Rank	3	1	1	7	3	5	6
	Median	9.00E+02	9.17E+02	9.32E+02	9.26E+02	9.00E+02	9.07E+02	9.18E+02
	Rank	1	4	7	6	1	3	5
	(Std Dev)	(3.03E+00)	(3.00E+01)	(4.26E+01)	(3.85E+00)	(0.00E+00)	(1.85E+00)	(6.51E+00)
f_{20}	Mean	9.01E+02	9.02E+02	9.34E+02	9.29E+02	9.00E+02	9.11E+02	9.19E+02
	Rank WST	2	3 >	7 >	6 >	1 =	4 >	5 >
	Min	9.00E+02	8.00E+02	8.00E+02	9.22E+02	9.00E+02	9.05E+02	9.07E+02
	Rank	3	1	1	7	3	5	6
	Median	9.00E+02	9.16E+02	9.36E+02	9.28E+02	9.00E+02	9.08E+02	9.18E+02
	Rank	1	4	7	6	1	3	5
	(Std Dev)	(3.96E+00)	(4.08E+01)	(4.52E+01)	(6.30E+00)	(0.00E+00)	(1.54E+01)	(6.81E+00)
f_{21}	Mean	661.233261089813	500.000046798861	981.482147070620	5.00000046798860	854.266458999325	809.235429268918	587.181287002300
	Rank WST	4	2 <	7 >	1 <	6 >	5 >	3 ≤
	Min	409.344479480473	500.000046798860	500.000046798861	500.000046798860	500.000046798860	409.344495851186	500.000046798860
	Rank	1	3	7	3	3	2	3
	Median	500.000046798860	500.000046798861	1171.24076429419	500.000046798860	850.000000000001	946.327108305680	500.000046798860
	Rank	1	4	7	1	5	6	1
	(Std Dev)	(1.96E+02)	(4.64E−13)	(3.21E+02)	(3.05E−13)	(3.26E+02)	(3.06E+02)	(2.26E+02)
f_{22}	Mean	9.07E+02	9.24E+02	9.20E+02	9.34E+02	1.04E+03	8.76E+02	9.25E+02
	Rank WST	2	4 >	3 >	6 >	7 >	1 <	5 >
	Min	8.56E+02	8.95E+02	8.55E+02	9.01E+02	9.66E+02	8.33E+02	8.62E+02
	Rank	3	5	2	6	7	1	4
	Median	9.05E+02	9.22E+02	9.19E+02	9.34E+02	1.05E+03	8.69E+02	9.20E+02
	Rank	2	5	3	6	7	1	4
	(Std Dev)	(2.56E+01)	(1.74E+01)	(3.73E+01)	(2.11E+01)	(4.29E+01)	(3.03E+01)	(2.85E+01)
f_{23}	Mean	649.294669458429	534.164548670978	997.235721285348	534.165875308246	789.051832758851	699.548473485693	591.808178466945
	Rank WST	4	1 <	2 =	7 >	6 >	5 >	3 ≤
	Min	534.164077735861	534.164077735866	534.164077735866	534.164077735867	534.164077735868	534.164077735870	534.164867375170
	Rank	1	2	2	4	5	6	7
	Median	534.164560483313	534.164577704798	1173.82694166518	534.164580179308	534.166529204030	534.164937587441	539.930746340269
	Rank	1	2	7	3	5	4	6
	(Std Dev)	(1.66E+02)	(4.14E−04)	(3.09E+02)	(7.42E−03)	(3.03E+02)	(2.43E+02)	(1.20E+02)
f_{24}	Mean	200.000000000002	2.00000000000003	200.000000000001	200.000000000001	267.393762055215	595.622074225127	200.000000000000
	Rank WST	4	5 =	2 =	2 =	6 >	7 >	1 =
	Min	200.000000000000	200.000000000003	200.000000000000	200.000000000000	200.000000000000	200.000000000003	200.000000000000
	Rank	1	6	1	1	1	6	1
	Median	200.000000000003	200.000000000003	200.000000000002	200.000000000000	200.000000000003	932.301577163733	200.000000000000
	Rank	4	4	3	1	4	7	1
	(Std Dev)	(1.40E−12)	(9.12E−13)	(1.53E−12)	(1.36E−12)	(2.57E+02)	(3.76E+02)	(0.00E+00)
f_{25}	Mean	200.000000000002	200.000000000003	200.000000000002	200.000000000000	273.345204779768	695.333073329963	286.984563408777
	Rank WST	2	4 =	2 =	1 =	5 >	7 >	6 ≥
	Min	200.000000000000	200.000000000003	200.000000000000	200.000000000000	200.000000000003	200.000000000000	200.000000000000
	Rank	1	6	1	1	6	1	1
	Median	200.000000000003	200.000000000003	200.000000000002	200.000000000000	200.000000000003	940.871036580070	200.000000000000
	Rank	4	4	3	1	4	7	1
	(Std Dev)	(1.14E−12)	(7.58E−13)	(2.64E−12)	(0.00E+00)	(2.60E+02)	(3.56E+02)	(2.65E+02)

In Fig. 6, the representative evolutionary process of functions are presented. IEPSO maintains the convergence rates in the earlier and later generations of the evolutionary process for f_2 , f_4 , and f_{14} ; the convergence rates for f_{20} and f_{25} are very sharp in the very early stage of the optimization process and keep flat subsequently. The current proposed strategy of IEPSO does not fit well compared to EPSO and

PPSO, for f_9 and f_{11} , receptively; therefore, there is still room for improvement. It is worth investigating the facts that EPSO maintains impressive convergence rates for f_9 and PPSO exhibits outstanding performance for f_{11} , on average, with the exception of the functions. From the figures, the search capability of IEPSO in neither effective nor efficient in both earlier and later generations, for f_9 and f_{11} . Either the

Table 8

Comparison of experimental results of PSO variants, averaged over 30 runs, for 50-dimensional CEC2005 benchmark functions.

f_n	Criteria	IEPSO	EPSO	MPSO	PSO-sono	TCSPSO	SaHSPS	PPSO
f_1	Mean	5.684E-14	1.213E-13	1.130E+03	5.684E-15	1.118E-13	7.769E-14	0.00E+00
	Rank WST	3	6 >	7 ≥	2 <	5 >	4 ≥	1 <
	Min	0.000E+00	5.684E-14	5.684E-14	0.000E+00	5.684E-14	5.684E-14	0.000E+00
	Rank	1	4	4	1	4	4	1
	Median	5.684E-14	1.137E-13	5.684E-14	0.000E+00	1.137E-13	5.684E-14	0.000E+00
	Rank	3	6	3	1	6	3	1
	(Std Dev)	(2.59E-14)	(2.88E-14)	(2.70E+03)	(2.29E-14)	(2.35E-14)	(2.79E-14)	(0.00E+00)
f_2	Mean	8.42E-05	7.31E-08	1.73E+02	3.76E+02	1.40E+04	9.88E+02	1.24E+03
	Rank WST	2	1 <	3 >	4 >	7 >	5 >	6 >
	Min	1.06E-05	8.42E-09	6.76E+00	6.21E-01	5.81E+03	2.63E-01	2.30E+02
	Rank	2	1	5	4	7	3	6
	Median	6.31E-05	6.65E-08	1.80E+01	6.12E+01	1.24E+04	4.01E+02	1.13E+03
	Rank	2	1	3	4	7	5	6
	(Std Dev)	(8.57E-05)	(4.91E-08)	(7.26E+02)	(8.77E+02)	(6.32E+03)	(1.67E+03)	(7.40E+02)
f_3	Mean	2.14E+06	4.26E+05	4.59E+07	4.67E+07	1.16E+08	1.99E+07	2.43E+06
	Rank WST	2	1 <	5 >	6 >	7 >	4 >	3 >
	Min	2.83E+04	2.32E+05	1.02E+06	6.04E+06	3.20E+07	7.15E+05	1.08E+06
	Rank	1	2	4	6	7	3	5
	Median	1.91E+06	4.22E+05	2.08E+07	2.65E+07	9.86E+07	3.15E+06	2.25E+06
	Rank	2	1	5	6	7	4	3
	(Std Dev)	(9.50E+05)	(1.12E+05)	(5.19E+07)	(4.65E+07)	(6.62E+07)	(5.05E+07)	(8.86E+05)
f_4	Mean	1.01E+03	8.56E+03	1.30E+04	1.43E+04	2.84E+04	1.49E+04	4.86E+04
	Rank WST	1	2 >	3 >	6 >	5 >	7 >	7 >
	Min	1.57E+02	3.15E+03	6.09E+03	4.32E+03	1.65E+04	2.89E+03	3.09E+04
	Rank	1	3	5	4	6	2	7
	Median	7.91E+02	7.81E+03	1.25E+04	1.35E+04	2.64E+04	1.29E+04	4.83E+04
	Rank	1	2	3	5	6	4	7
	(Std Dev)	(7.41E+02)	(3.66E+03)	(4.36E+03)	(7.16E+03)	(6.29E+03)	(9.67E+03)	(8.95E+03)
f_5	Mean	5.6323E+03	1.1244E+04	8.6198E+03	1.1240E+04	1.3960E+04	9.0133E+03	1.0325E+04
	Rank WST	1	6 >	2 >	5 >	7 >	3 >	4 >
	Min	3.3525E+03	8.1204E+03	5.7268E+03	8.4395E+03	1.0586E+04	5.8051E+03	8.5219E+03
	Rank	1	4	2	5	7	3	6
	Median	4.9115E+03	1.0819E+04	8.4699E+03	1.0906E+04	1.3500E+04	8.5331E+03	1.0246E+04
	Rank	1	5	2	6	7	3	4
	(Std Dev)	(1.82E+03)	(2.16E+03)	(1.69E+03)	(1.79E+03)	(1.93E+03)	(2.64E+03)	(1.20E+03)
f_6	Mean	6.36E+01	2.28E+01	5.60E+07	6.41E+01	1.17E+02	6.49E+02	2.17E+02
	Rank WST	2	1 <	7 >	3 >	4 >	6 >	5 >
	Min	2.35E+01	9.52E-04	5.15E+02	2.59E+01	5.58E+00	7.37E+00	3.88E+01
	Rank	4	1	7	5	2	3	6
	Median	3.88E+01	1.22E+01	2.27E+07	4.36E+01	8.81E+01	9.11E+01	9.83E+01
	Rank	2	1	7	3	4	5	6
	(Std Dev)	(5.59E+01)	(2.87E+01)	(6.97E+07)	(3.58E+01)	(1.05E+02)	(2.88E+03)	(2.01E+02)
f_7	Mean	8.45E-03	1.12E-02	1.06E+03	3.61E-01	1.15E+02	8.62E-03	5.11E-01
	Rank WST	1	3 ≥	7 >	4 >	6 >	2 ≥	5 >
	Min	2.84E-14	1.56E-12	1.73E-02	5.01E-08	1.48E+00	1.93E-12	1.31E-05
	Rank	1	2	6	4	7	3	5
	Median	1.09E-12	8.63E-03	1.19E+03	2.84E-05	1.07E+02	7.40E-03	4.41E-02
	Rank	1	4	7	2	6	3	5
	(Std Dev)	(9.84E-03)	(1.47E-02)	(7.25E+02)	(1.23E+00)	(9.12E+01)	(1.02E-02)	(1.44E+00)
f_8	Mean	2.1103E+01	2.1105E+01	2.1096E+01	2.1098E+01	2.1109E+01	2.1038E+01	2.0176E+01
	Rank WST	5	6 ≥	3 ≤	4 ≤	7 >	2 <	1 <
	Min	2.1001E+01	2.1009E+01	2.0845E+01	2.1016E+01	2.1030E+01	2.0918E+01	2.0058E+01
	Rank	4	5	2	6	7	3	1
	Median	2.1101E+01	2.1105E+01	2.1102E+01	2.1100E+01	2.1112E+01	2.1056E+01	2.0174E+01
	Rank	4	6	5	3	7	2	1
	(Std Dev)	(3.01E-02)	(4.02E-02)	(6.72E-02)	(4.52E-02)	(3.65E-02)	(5.75E-02)	(5.63E-02)
f_9	Mean	1.016182E+02	2.160050E-13	8.018712E+01	4.871979E+01	1.399141E+02	1.343084E+02	6.589938E+01
	Rank WST	5	1 <	4 ≤	2 <	7 >	6 >	3 ≤
	Min	3.382856E+01	1.136868E-13	5.472267E+01	2.288406E+01	6.884752E+01	5.472271E+01	3.880339E+01
	Rank	3	1	5	2	7	6	4
	Median	6.566723E+01	2.273737E-13	7.484116E+01	4.974791E+01	1.345391E+02	1.364891E+02	6.516974E+01
	Rank	4	1	5	2	6	7	3
	(Std Dev)	(6.70E+01)	(4.81E-14)	(2.16E+01)	(1.11E+01)	(3.68E+01)	(3.82E+01)	(1.28E+01)
f_{10}	Mean	9.90E+01	1.41E+02	1.70E+02	6.28E+01	3.25E+02	1.99E+02	7.31E+01
	Rank WST	3	4 >	5 >	1 <	7 >	6 >	2 ≤
	Min	4.28E+01	8.16E+01	1.02E+02	3.88E+01	1.92E+02	1.24E+02	4.78E+01
	Rank	2	4	5	1	7	6	3
	Median	7.21E+01	1.31E+02	1.70E+02	6.02E+01	2.97E+02	1.88E+02	7.36E+01
	Rank	2	4	5	1	7	6	3
	(Std Dev)	(6.18E+01)	(4.30E+01)	(4.76E+01)	(1.45E+01)	(1.09E+02)	(4.35E+01)	(1.35E+01)

(continued on next page)

global or local search capability is insufficient in the current search strategy of IEPSO, the search behavior needs to be further improved in the future.

4.4. Search behavior of IEPSO

In Fig. 3, the best fitness values with their sources are plotted using different colors. The search modes are as follows:

Table 8 (continued).

f_n	Criteria	IEPSO	EPSO	MPSO	PSO-sono	TCSPSO	SaHSPS	PPSO
f_{11}	Mean	4.66E+01	4.60E+01	5.13E+01	5.53E+01	5.65E+01	4.79E+01	3.06E+01
	Rank WST	3	2 $\leq$	5 $>$	6 $>$	7 $>$	4 $>$	1 $<$
	Min	1.51E+01	3.88E+01	3.99E+01	3.97E+01	4.89E+01	3.44E+01	2.39E+01
	Rank	1	4	6	5	7	3	2
	Median	4.74E+01	4.58E+01	5.15E+01	5.69E+01	5.65E+01	4.77E+01	3.02E+01
	Rank	3	2	5	7	6	4	1
	(Std Dev)	(7.63E+00)	(3.85E+00)	(4.41E+00)	(5.80E+00)	(4.79E+00)	(6.91E+00)	(3.68E+00)
f_{12}	Mean	2.23E+04	2.51E+04	3.58E+05	8.63E+04	1.69E+05	1.78E+05	3.53E+04
	Rank WST	1	2 $\geq$	7 $>$	4 $>$	5 $>$	6 $>$	3 $>$
	Min	4.99E+02	1.37E+03	1.40E+05	9.65E+03	1.69E+04	4.43E+03	6.65E+03
	Rank	1	2	7	5	6	3	4
	Median	1.55E+04	1.43E+04	3.54E+05	7.62E+04	1.35E+05	4.41E+04	2.81E+04
	Rank	2	1	7	5	6	4	3
	(Std Dev)	(1.69E+04)	(2.51E+04)	(1.34E+05)	(6.28E+04)	(1.10E+05)	(2.66E+05)	(2.46E+04)
f_{13}	Mean	5.52E+00	3.61E+00	6.03E+00	6.79E+00	9.03E+00	9.53E+00	6.72E+00
	Rank WST	2	1 $<$	3 $>$	5 $>$	6 $>$	7 $>$	4 $>$
	Min	3.07E+00	2.80E+00	3.25E+00	4.00E+00	6.01E+00	6.12E+00	4.33E+00
	Rank	2	1	3	4	6	7	5
	Median	5.61E+00	3.57E+00	5.97E+00	6.83E+00	8.90E+00	9.08E+00	6.45E+00
	Rank	2	1	3	5	6	7	4
	(Std Dev)	(1.54E+00)	(4.60E-01)	(1.54E+00)	(1.40E+00)	(1.87E+00)	(2.39E+00)	(1.15E+00)
f_{14}	Mean	2.101E+01	2.166E+01	2.254E+01	2.123E+01	2.184E+01	2.231E+01	2.255E+01
	Rank WST	1	3 $>$	6 $>$	2 $>$	4 $>$	5 $>$	7 $>$
	Min	1.945E+01	1.958E+01	2.128E+01	1.979E+01	2.055E+01	2.141E+01	2.134E+01
	Rank	1	2	5	3	4	7	6
	Median	2.120E+01	2.181E+01	2.262E+01	2.126E+01	2.180E+01	2.238E+01	2.252E+01
	Rank	1	4	7	2	3	5	6
	(Std Dev)	(7.52E-01)	(6.40E-01)	(4.69E-01)	(6.47E-01)	(4.71E-01)	(4.44E-01)	(5.10E-01)
f_{15}	Mean	3.623E+02	6.855E+01	2.853E+02	2.829E+02	3.306E+02	4.313E+02	3.846E+02
	Rank WST	5	1 $<$	3 $<$	2 $<$	4 $\leq$	7 $>$	6 $>$
	Min	5.881E+01	8.431E-04	6.892E+01	2.002E+02	1.723E+02	2.000E+02	2.684E+02
	Rank	2	1	3	6	4	5	7
	Median	4.000E+02	4.000E+01	2.770E+02	2.962E+02	3.321E+02	4.122E+02	4.000E+02
	Rank	5	1	2	3	4	7	5
	(Std Dev)	(8.54E+01)	(7.33E+01)	(1.29E+02)	(6.45E+01)	(7.81E+01)	(1.36E+02)	(4.31E+01)
f_{16}	Mean	1.37E+02	1.67E+02	2.12E+02	9.90E+01	2.63E+02	1.79E+02	1.55E+02
	Rank WST	2	4 $>$	6 $>$	1 $\leq$	7 $>$	5 $>$	3 $\geq$
	Min	5.44E+01	7.38E+01	7.45E+01	4.70E+01	1.42E+02	9.23E+01	5.52E+01
	Rank	2	4	5	1	7	6	3
	Median	9.23E+01	1.77E+02	1.50E+02	5.86E+01	2.37E+02	1.41E+02	7.73E+01
	Rank	3	6	5	1	7	4	2
	(Std Dev)	(1.09E+02)	(7.44E+01)	(1.21E+02)	(9.15E+01)	(9.33E+01)	(9.73E+01)	(1.40E+02)
f_{17}	Mean	1.165E+02	1.514E+02	2.234E+02	1.129E+02	3.202E+02	2.238E+02	2.237E+02
	Rank WST	2	3 $>$	4 $>$	1 $\leq$	7 $>$	6 $>$	5 $>$
	Min	5.110E+01	5.649E+01	7.822E+01	4.579E+01	1.305E+02	1.202E+02	7.014E+01
	Rank	2	3	5	1	7	6	4
	Median	8.964E+01	1.088E+02	1.338E+02	7.665E+01	3.215E+02	1.874E+02	1.444E+02
	Rank	2	3	4	1	7	6	5
	(Std Dev)	(8.90E+01)	(9.71E+01)	(1.56E+02)	(9.63E+01)	(9.38E+01)	(1.14E+02)	(1.52E+02)
f_{18}	Mean	9.0231310E+02	9.6074288E+02	1.0028306E+03	9.6190231E+02	9.0000000E+02	9.0924346E+02	9.7128563E+02
	Rank WST	2	4 $>$	7 $>$	5 $>$	1 $=$	3 $>$	6 $>$
	Min	9.0000000E+02	9.4113556E+02	9.0000005E+02	9.4800779E+02	9.0000000E+02	9.0000000E+02	9.4122726E+02
	Rank	1	5	4	7	1	1	6
	Median	9.0000000E+02	9.5938286E+02	1.0063086E+03	9.5879223E+02	9.0000000E+02	9.0000000E+02	9.7031995E+02
	Rank	1	5	7	4	1	1	6
	(Std Dev)	(1.27E+01)	(1.07E+01)	(8.61E+01)	(1.02E+01)	(0.00E+00)	(1.57E+01)	(1.53E+01)
f_{19}	Mean	900.000000	954.226143	988.445961	962.580695	900.000000	906.874815	964.444373
	Rank WST	1	4 $>$	7 $>$	5 $>$	1 $=$	3 $>$	6 $>$
	Min	900.000000	800.000000	900.000004	946.852999	900.000000	900.000000	800.000000
	Rank	3	1	6	7	3	3	1
	Median	900.000000	961.086234	987.377347	961.267157	900.000000	900.000000	967.762907
	Rank	1	4	7	5	1	1	6
	(Std Dev)	(1.37E-12)	(3.12E+01)	(8.69E+01)	(9.61E+00)	(0.00E+00)	(1.54E+01)	(3.28E+01)
f_{20}	Mean	9.0000000E+02	9.5955784E+02	1.0038841E+03	9.5993597E+02	9.0000000E+02	9.1248152E+02	9.7006094E+02
	Rank WST	1	4 $>$	7 $>$	5 $>$	1 $=$	3 $>$	6 $>$
	Min	9.0000000E+02	8.0000000E+02	9.0000001E+02	9.4367366E+02	9.0000000E+02	9.0000000E+02	9.4372761E+02
	Rank	2	1	5	6	2	2	7
	Median	9.0000000E+02	9.6487990E+02	9.9635336E+02	9.5851442E+02	9.0000000E+02	9.0000000E+02	9.7092971E+02
	Rank	1	5	7	4	1	1	6
	(Std Dev)	(0.00E+00)	(3.18E+01)	(8.62E+01)	(9.44E+00)	(0.00E+00)	(1.96E+01)	(1.15E+01)

(continued on next page)

• Niching behavior conducts an effective search in the earlier generations and EPSO with a large subpopulation conducts an effective search in the later generations.

• Niching behavior and CMAR conduct effective searches in the earlier generations and EPSO with a large subpopulation conducts an effective search in the later generations.

Table 8 (continued).

f_n	Criteria	IEPSO	EPSO	MPSO	PSO-sono	TCSPSO	SaHSPS	PPSO
f_{21}	Mean	613.437569433645	500.000008461117	1161.46110536247	500.000008461113	883.676576815860	819.491033205878	636.414818897335
	Rank WST	3	2 $\leq$	7 $>$	1 $\leq$	6 $>$	5 $>$	4 $\geq$
	Min	500.000008461113	500.000008461115	500.000008461117	500.000008461113	500.000008461114	500.000008461114	500.000008461115
	Rank	1	5	7	1	3	5	5
	Median	500.000008461115	500.000008461117	1231.66864340018	500.000008461113	1203.09107654092	1013.55388182718	500.000008461115
	Rank	2	4	7	1	6	5	2
	(Std Dev)	(2.60E+02)	(1.15E-12)	(2.24E+02)	(6.13E-13)	(3.65E+02)	(2.67E+02)	(2.78E+02)
f_{22}	Mean	9.52E+02	9.74E+02	9.53E+02	9.80E+02	1.09E+03	9.16E+02	9.50E+02
	Rank WST	3	5 $>$	4 $\geq$	6 $>$	7 $>$	1 $<$	2 $\leq$
	Min	9.29E+02	9.53E+02	9.01E+02	9.50E+02	1.05E+03	8.77E+02	9.17E+02
	Rank	4	6	2	5	7	1	3
	Median	9.53E+02	9.73E+02	9.49E+02	9.81E+02	1.09E+03	9.06E+02	9.47E+02
	Rank	4	5	3	6	7	1	2
	(Std Dev)	(1.47E+01)	(1.33E+01)	(1.96E+01)	(1.31E+01)	(2.68E+01)	(3.30E+01)	(2.05E+01)
f_{23}	Mean	742.609853	539.122883	1017.11242	540.367686	972.254549	777.585020	735.404921
	Rank WST	4	1 $<$	7 $>$	2 $<$	6 $>$	5 $\geq$	3 $\leq$
	Min	539.122187	539.122186	539.123391	539.122612	539.122245	539.122175	539.178692
	Rank	3	2	6	5	4	1	7
	Median	541.596572	539.122830	121.637167	539.156626	1204.19165	726.098769	655.216820
	Rank	3	1	7	2	6	5	4
	(Std Dev)	(2.84E+02)	(5.21E-04)	(3.19E+02)	(3.72E+00)	(3.32E+02)	(2.44E+02)	(2.18E+02)
f_{24}	Mean	200.000000000003	264.811167746673	633.822239303899	200.000000000000	1220.45492957378	976.410356096796	1123.19942383807
	Rank WST	2	3 $\geq$	4 $>$	1 $=$	7 $>$	5 $>$	6 $>$
	Min	200.000000000000	200.000000000000	200.000000000003	200.000000000000	1157.74590818420	964.082692224894	200.000000000007
	Rank	1	1	4	1	7	6	5
	Median	200.000000000003	200.000000000010	237.179562382916	200.000000000000	1227.07809760976	976.440099157082	1156.02491605009
	Rank	2	3	4	1	7	5	6
	(Std Dev)	(1.60E-12)	(2.47E+02)	(4.81E+02)	(9.19E-13)	(3.40E+01)	(7.56E+00)	(1.76E+02)
f_{25}	Mean	200.000000000004	391.86465255472	743.149305184622	202.832607769106	1227.36369568358	923.488674988717	1188.66670948000
	Rank WST	1	3 $>$	4 $>$	2 $\geq$	7 $>$	5 $>$	6 $>$
	Min	200.000000000000	200.000000000007	200.000000000003	200.000000000000	1165.90755879837	200.000000000003	1148.16101167538
	Rank	1	5	3	1	7	3	6
	Median	200.000000000003	200.000000000010	949.974656562677	200.000000000000	1230.6315344879	974.215549242883	1190.06322381066
	Rank	2	3	4	1	7	5	6
	(Std Dev)	(1.85E-12)	(3.90E+02)	(4.65E+02)	(1.55E+01)	(3.03E+01)	(1.97E+02)	(1.45E+01)

Table 9

Statistics of rankings for IEPSO, EPSO, MPSO, PSO-sono, TCSPSO, SaHSPS and PPSO.

Dim	Criteria	IEPSO	EPSO	MPSO	PSO-sono	TCSPSO	SaHSPS	PPSO
10D	Mean (Rankings 1st 2nd 3rd)	6 6 7	8 7 6	1 1 1	6 6 4	1 0 2	4 2 3	5 1 1
	Min (Rankings 1st 2nd 3rd)	18 1 1	10 3 5	6 0 2	11 3 2	6 1 1	9 1 3	7 2 2
	Median (Rankings 1st 2nd 3rd)	9 6 2	8 4 7	4 0 2	9 6 2	3 0 3	6 3 2	9 1 2
	SUM (Rankings 1st 2nd 3rd)	33 13 10	26 14 18	11 1 5	26 15 8	10 1 6	19 6 8	21 4 5
30D	Mean (Rankings 1st 2nd 3rd)	7 10 3	8 4 5	0 3 5	3 7 3	3 0 0	1 1 3	4 1 4
	Min (Rankings 1st 2nd 3rd)	11 4 8	8 5 4	5 4 3	6 2 6	1 1 3	4 2 0	4 4 1
	Median (Rankings 1st 2nd 3rd)	11 8 3	5 4 5	0 0 7	6 4 3	3 0 1	2 0 7	6 2 2
	SUM (Rankings 1st 2nd 3rd)	29 22 14	21 13 14	5 7 15	15 13 12	7 1 4	7 3 10	14 7 7
50D	Mean (Rankings 1st 2nd 3rd)	8 8 5	7 4 5	0 1 5	5 6 1	3 0 0	1 2 4	3 2 5
	Min (Rankings 1st 2nd 3rd)	12 7 3	8 5 2	0 3 3	7 1 1	1 2 2	3 2 11	3 1 3
	Median (Rankings 1st 2nd 3rd)	7 10 4	8 2 3	0 2 5	7 4 3	3 0 1	4 1 3	3 3 4
	SUM (Rankings 1st 2nd 3rd)	27 25 12	23 11 10	0 6 13	19 11 5	7 2 3	8 5 18	9 6 12

- *Niching behavior and CMAR conduct effective searches in the earlier generations, EPSO with a large subpopulation conducts an effective search in the middle of the evolutionary process, and CMAR conducts an effective search in the later generations.*
- *Niching behavior, EPSO with a large subpopulation, and CMAR conduct effective searches in the earlier generations, irregularly. Then, EPSO with a large subpopulation takes over in the middle generations, and the niching behavior conducts an effective search for the rest of the evolutionary process.*

In different evolutionary stages, the landscapes near the particles keep changing. An ideal strategy can guide the evolutionary process to fit well within different landscapes. In general, the proposed IEPSO strategy adjusts its search behavior according to the complex optimization problems. The strategy is more effective and efficient, in most cases, compared with the state-of-the-art PSO algorithms.

In Fig. 3, f_5 is a unimodal function, f_{10} and f_{12} are multimodal functions, f_{13} is an expanded function, and f_{19} and f_{20} are hybrid

composition functions. It can be observed from the search behavior profiles that IEPSO employs the characteristics of EPSO large subpopulation, niching subpopulation, and CMAR subpopulation. The different subpopulations guide IEPSO to find better solutions according to the problems. The CMAR subpopulation performs the major searching role in f_{12} , whereas in f_{20} , it performs a minor role. The appropriate subpopulation takes over according to the needs of the optimization process.

4.5. Algorithmic computational complexity

The computational complexity of the proposed IEPSO algorithm was determined as defined in the CEC2005 benchmark competition [55]. The algorithm complexities were calculated on 10, 30, and 50 dimensions. All the experiments were conducted on the following system:

- OS: Windows 10
- CPU: Intel Core i7-1165G7 @ 2.80 GHz 1.69 GHz CPU
- RAM: 16 GB

Table 10

Mean best values (averaged over 30 runs) and standard deviations (in parentheses) of IEP SO with different allocation rates, tested on 30-dimensional CEC2005 benchmark functions.

f_n	IEPSO (10%-20%-70%)		IEPSO with sample pool		IEPSO with sample pool	
	with sample pool vs. without sample pool		10%-20%-70% vs. 0%-30%-70%		10%-20%-70% vs. 30%-0%-70%	
f_1	4.55E-14 (2.31E-14)	2.30E-07 (9.18E-07)	4.55E-14 (2.31E-14)	5.31E-14 (1.44E-14)	4.55E-14 (2.31E-14)	4.93E-14 (1.97E-14)
f_2	2.54E-10 (8.06E-10)	8.57E-03 (1.01E-02)	2.54E-10 (8.06E-10)	1.21E-11 (1.35E-11)	2.54E-10 (8.06E-10)	1.55E-08 (3.38E-08)
f_3	2.87E+05 (5.52E+05)	1.22E+06 (7.61E+05)	2.87E+05 (5.52E+05)	1.12E+06 (5.38E+05)	2.87E+05 (5.52E+05)	3.16E+02 (1.37E+02)
f_4	1.75E+00 (3.75E+00)	1.64E+01 (1.96E+01)	1.75E+00 (3.75E+00)	2.46E-01 (3.91E-01)	1.75E+00 (3.75E+00)	1.79E+02 (1.59E+02)
f_5	3.53E+03 (1.01E+03)	3.91E+03 (9.78E+02)	3.53E+03 (1.01E+03)	3.81E+03 (7.87E+02)	3.53E+03 (1.01E+03)	5.72E+02 (8.86E+02)
f_6	4.37E+01 (5.86E+01)	9.73E+01 (1.32E+02)	4.37E+01 (5.86E+01)	2.83E+01 (3.99E+01)	4.37E+01 (5.86E+01)	7.35E+00 (5.27E+00)
f_7	1.70E-02 (1.40E-02)	2.43E-02 (2.12E-02)	1.70E-02 (1.40E-02)	1.76E-02 (1.82E-02)	1.70E-02 (1.40E-02)	2.17E-02 (1.85E-02)
f_8	2.049E+01 (4.45E-01)	2.054E+01 (4.33E-01)	2.049E+01 (4.45E-01)	2.095E+01 (3.95E-02)	2.049E+01 (4.45E-01)	2.000E+01 (0.00E+00)
f_9	2.02E+01 (8.61E+00)	7.96E+00 (3.24E+00)	2.02E+01 (8.61E+00)	3.01E+01 (1.20E+01)	2.02E+01 (8.61E+00)	4.87E+01 (4.12E+01)
f_{10}	5.49E+01 (4.49E+01)	7.91E+01 (4.72E+01)	5.49E+01 (4.49E+01)	7.93E+01 (5.30E+01)	5.49E+01 (4.49E+01)	5.14E+01 (1.31E+01)
f_{11}	1.31E+01 (4.53E+00)	2.40E+01 (3.25E+00)	1.31E+01 (4.53E+00)	2.29E+01 (2.81E+00)	1.31E+01 (4.53E+00)	1.35E+01 (3.11E+00)
f_{12}	3.98E+03 (4.06E+03)	8.65E+03 (8.77E+03)	3.98E+03 (4.06E+03)	1.68E+04 (1.58E+04)	3.98E+03 (4.06E+03)	2.60E+03 (3.98E+03)
f_{13}	2.54E+00 (5.40E-01)	2.64E+00 (6.46E-01)	2.54E+00 (5.40E-01)	2.91E+00 (6.69E-01)	2.54E+00 (5.40E-01)	2.87E+00 (6.76E-01)
f_{14}	1.18E+01 (6.99E-01)	1.22E+01 (5.25E-01)	1.18E+01 (6.99E-01)	1.14E+01 (5.87E-01)	1.18E+01 (6.99E-01)	1.20E+01 (6.40E-01)
f_{15}	3.48E+02 (1.01E+02)	3.59E+02 (1.20E+02)	3.48E+02 (1.01E+02)	3.49E+02 (8.76E+01)	3.48E+02 (1.01E+02)	3.53E+02 (9.65E+01)
f_{16}	1.54E+02 (1.27E+02)	1.75E+02 (1.51E+02)	1.54E+02 (1.27E+02)	2.80E+02 (1.73E+02)	1.54E+02 (1.27E+02)	1.89E+02 (1.52E+02)
f_{17}	1.53E+02 (1.24E+02)	1.08E+02 (6.88E+01)	1.53E+02 (1.24E+02)	2.36E+02 (1.87E+02)	1.53E+02 (1.24E+02)	1.74E+02 (1.45E+02)
f_{18}	9.01E+02 (3.51E+00)	9.00E+02 (9.31E-13)	9.01E+02 (3.51E+00)	9.16E+02 (4.13E+00)	9.01E+02 (3.51E+00)	9.00E+02 (8.85E-13)
f_{19}	9.008E+02 (3.03E+00)	9.007E+02 (3.75E+00)	9.01E+02 (3.03E+00)	9.18E+02 (3.24E+01)	9.01E+02 (3.03E+00)	9.00E+02 (8.85E-13)
f_{20}	9.01E+02 (3.96E+00)	9.00E+02 (2.20E+00)	9.01E+02 (3.96E+00)	9.18E+02 (4.08E+00)	9.01E+02 (3.96E+00)	9.00E+02 (8.48E-13)
f_{21}	6.61E+02 (1.96E+02)	5.53E+02 (1.71E+02)	6.61E+02 (1.96E+02)	6.06E+02 (2.37E+02)	6.61E+02 (1.96E+02)	5.32E+02 (1.30E+02)
f_{22}	9.07E+02 (2.56E+01)	9.18E+02 (2.06E+01)	9.07E+02 (2.56E+01)	9.01E+02 (1.63E+01)	9.07E+02 (2.56E+01)	9.65E+02 (3.63E+01)
f_{23}	6.49E+02 (1.66E+02)	5.61E+02 (1.01E+02)	6.49E+02 (1.66E+02)	6.95E+02 (2.57E+02)	6.49E+02 (1.66E+02)	6.66E+02 (2.68E+02)
f_{24}	200.000000000002 (1.40E-12)	200.000000092634 (1.59E-07)	200.000000000002 (1.40E-12)	200.000000000003 (7.65E-13)	200.000000000002 (1.40E-12)	200.000000000003 (7.65E-13)
f_{25}	200.000000000002 (1.14E-12)	200.000000025191 (6.97E-08)	200.000000000002 (1.14E-12)	200.000000000002 (1.23E-12)	200.000000000002 (1.14E-12)	200.000000000003 (9.63E-13)
Total	18	7	19	7	15	10

• Language: Matlab 2018b

The computation complexity of IEP SO algorithm on 10, 30, and 50 dimensions follows [55], and are presented in Table 4. In Table 4, T_0 is the computational time for executing the following statements:

for $i = 1 : 1000000$

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

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

end

T_1 is the time required to execute the benchmark function f_3 independently for 200,000 evaluations with D dimensions. T_2 is the

execution time of IEP SO algorithm on function f_3 with 200,000 evaluations in D dimensions. The time ($\hat{T}_2$) is the mean value, averaged over 5 runs, of T_2 . As shown in Table 4, T_1 , $\hat{T}_2$, and $((\hat{T}_2 - T_1)/T_0)$ scaled linearly with the number of dimensions. Comparison of the computational complexity of the IEP SO algorithm with the compared PSO variants is presented in Table 5 for 30 dimensions. The same procedure and system are used to calculate the computational complexity of each PSO variant.

In Table 5, although the computational complexity of IEP SO ranks 3rd, behind PSO-sono and PPSO, it is much better than EPSO. Thus, IEP SO is much more efficient and performs better than EPSO. The computational cost of IEP SO is reasonable and affordable compared with EPSO.

5. Conclusions

This paper proposed a novel ensemble strategy, which uses the niching behavior and CMAR phase, to enhance both the global and local search capabilities of EPSO. A perturbing mechanism and sample pool, sharing information between subpopulations, were employed to replace the particles of the subpopulations. The particles were divided into EPSO large, niching-based inertia weight PSO, and CMAR subpopulations. Sophisticated allocation rates were recommended and they performed best when combined with the novel ensemble strategy; validation with different allocation rates demonstrated that the proposed strategy is reasonable and effective. The EPSO large subpopulation maintained the major characteristics of EPSO, the niching-based inertia weight PSO subpopulation enhanced the population diversity, and the CMAR subpopulation enhanced the local search capability. The validation of IEP SO with/without sample pool illustrates the effectiveness of the sample pool mechanism. Experimental results showed the outstanding performance and superior scalability of the proposed ensemble strategy compared to the most recent PSO algorithms. However, the proposed IEP SO algorithm has some limitations: The fixed allocation rates may limit the balance control between local and global search capabilities and the selection of the current local search method is not dynamic. Future research directions for IEP SO include applying to real-world problems, dynamic adjustment of the particle number to perturb subpopulations instead of replacing just one particle, tuning of the subpopulation size during the evolutionary process, and designing adaptive selection mechanism to choose a method according to the landscape of particles such as sequential quadratic programming [56], interior-point method [57,58], and CMAR, multiple local search methods.

CRedit authorship contribution statement

Libin Hong: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Formal analysis, Editing & revision. **Xinmeng Yu:** Software, Investigation, Writing – review & editing. **Ben Wang:** Review & editing. **John Woodward:** Review & editing. **Ender Özcan:** Review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Libin HONG reports administrative support was provided by Hangzhou Normal University.

Data availability

Data will be made available on request.

References

- [1] D. Wolpert, W. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.* 1 (1) (1997) 67–82.
- [2] O.S. Ajani, R. Mallipeddi, Adaptive evolution strategy with ensemble of mutations for reinforcement learning, *Knowl.-Based Syst.* (ISSN: 0950-7051) 245 (2022) 108624.
- [3] L. Hong, C. Liu, J. Cui, F. Liu, Mutation strategy based on step size and survival rate for evolutionary programming, *Discrete Dyn. Nat. Soc.* (ISSN: 1026-0226) 2021 (2021).
- [4] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, P. Suganthan, Ensemble of differential evolution variants, *Inform. Sci.* (ISSN: 0020-0255) 423 (2018) 172–186.
- [5] N. Lynn, P.N. Suganthan, Ensemble particle swarm optimizer, *Appl. Soft Comput.* (ISSN: 1568-4946) 55 (2017) 533–548.
- [6] P. Karthikeyan, S. Baskar, Genetic algorithm with ensemble of immigrant strategies for multicast routing in ad hoc networks, *Soft Comput.* (ISSN: 1026-0226) 19 (2015) 489–498.
- [7] Y. Chen, M.-L. Wong, H. Li, Applying ant colony optimization to configuring stacking ensembles for data mining, *Expert Syst. Appl.* (ISSN: 0957-4174) 41 (6) (2014) 2688–2702.
- [8] Y. Wang, B. Li, T. Weise, Two-stage ensemble memetic algorithm: Function optimization and digital IIR filter design, *Inform. Sci.* (ISSN: 0020-0255) 220 (2013) 408–424, *Online Fuzzy Machine Learning and Data Mining*.
- [9] R. Eberhart, J. Kennedy, Particle swarm optimization, in: *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, Citeseer, 1995, pp. 1942–1948.
- [10] T. Shankar, S. Shanmugavel, A. Rajesh, Hybrid HSA and PSO algorithm for energy efficient cluster head selection in wireless sensor networks, *Swarm Evol. Comput.* 30 (2016) 1–10.
- [11] A.A. Yahya, A. Osman, M.S. El-Bashir, Rocchio algorithm-based particle initialization mechanism for effective PSO classification of high dimensional data, *Swarm Evol. Comput.* 34 (2017) 18–32.
- [12] A. Wadhwa, M.K. Thakur, Most significant hotspot detection using improved particle swarm optimizers, *Swarm Evol. Comput.* (2022) 101177.
- [13] W. Han, H. Li, M. Gong, J. Li, Y. Liu, Z. Wang, Multi-swarm particle swarm optimization based on CUDA for sparse reconstruction, *Swarm Evol. Comput.* 75 (2022) 101153.
- [14] W. Li, X. Meng, Y. Huang, Z.-H. Fu, Multipopulation cooperative particle swarm optimization with a mixed mutation strategy, *Inform. Sci.* 529 (2020) 179–196.
- [15] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360)*, IEEE, 1998, pp. 69–73.
- [16] Z.-H. Zhan, J. Zhang, Y. Li, H.S.-H. Chung, Adaptive particle swarm optimization, *IEEE Trans. Syst. Man Cybern. B* 39 (6) (2009) 1362–1381.
- [17] K. Chen, F. Zhou, A. Liu, Chaotic dynamic weight particle swarm optimization for numerical function optimization, *Knowl.-Based Syst.* 139 (2018) 23–40.
- [18] J. Kennedy, R. Mendes, Population structure and particle swarm performance, in: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 2, IEEE, 2002, pp. 1671–1676.
- [19] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, *IEEE Trans. Evol. Comput.* 8 (3) (2004) 204–210.
- [20] J.-J. Liang, P.N. Suganthan, Dynamic multi-swarm particle swarm optimizer, in: *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, IEEE, 2005, pp. 124–129.
- [21] L. Cao, L. Xu, E.D. Goodman, A neighbor-based learning particle swarm optimizer with short-term and long-term memory for dynamic optimization problems, *Inform. Sci.* 453 (2018) 463–485.
- [22] H.-R. Liu, J.-C. Cui, Z.-D. Lu, D.-Y. Liu, Y.-J. Deng, A hierarchical simple particle swarm optimization with mean dimensional information, *Appl. Soft Comput.* 76 (2019) 712–725.
- [23] Y.-J. Gong, J.-J. Li, Y. Zhou, Y. Li, H.S.-H. Chung, Y.-H. Shi, J. Zhang, Genetic learning particle swarm optimization, *IEEE Trans. Cybern.* 46 (10) (2015) 2277–2290.
- [24] A. Meng, Z. Li, H. Yin, S. Chen, Z. Guo, Accelerating particle swarm optimization using crisscross search, *Inform. Sci.* 329 (2016) 52–72.
- [25] X. Zhang, X. Wang, Q. Kang, J. Cheng, Differential mutation and novel social learning particle swarm optimization algorithm, *Inform. Sci.* 480 (2019) 109–129.
- [26] P. Moradi, M. Gholampour, A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy, *Appl. Soft Comput.* 43 (2016) 117–130.
- [27] F. Wang, H. Zhang, K. Li, Z. Lin, J. Yang, X.-L. Shen, A hybrid particle swarm optimization algorithm using adaptive learning strategy, *Inform. Sci.* 436 (2018) 162–177.
- [28] K. Chen, F.-Y. Zhou, X.-F. Yuan, Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection, *Expert Syst. Appl.* 128 (2019) 140–156.
- [29] M. Chih, Stochastic stability analysis of particle swarm optimization with pseudo random number assignment strategy, *European J. Oper. Res.* (ISSN: 0377-2217) (2022).
- [30] Y. Shi, R. Eberhart, Empirical study of particle swarm optimization, in: *Proceedings of the 1999 Congress on Evolutionary Computation*, vol. 3, 1999, pp. 1945–1950.
- [31] 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.* (ISSN: 1089-778X) 10 (3) (2006) 281–295.
- [32] T. Peram, K. Veeramachaneni, C. Mohan, Fitness-distance-ratio based particle swarm optimization, in: *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, 2003, pp. 174–181.
- [33] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, *IEEE Trans. Evol. Comput.* (ISSN: 1089-778X) 8 (3) (2004) 240–255.
- [34] B.Y. Qu, P.N. Suganthan, S. Das, A distance-based locally informed particle swarm model for multimodal optimization, *IEEE Trans. Evol. Comput.* (ISSN: 1089-778X) 17 (3) (2013) 387–402.
- [35] G. Wu, R. Mallipeddi, P.N. Suganthan, Ensemble strategies for population-based optimization algorithms—A survey, *Swarm Evol. Comput.* 44 (2019) 695–711.

- [36] Z. Wang, Z. Chen, Z. Wang, J. Wei, X. Chen, Q. Li, Y. Zheng, W. Sheng, Adaptive memetic differential evolution with multi-niche sampling and neighborhood crossover strategies for global optimization, *Inform. Sci. (ISSN: 0020-0255)* 583 (2022) 121–136.
- [37] X. Li, Niching without niching parameters: Particle swarm optimization using a ring topology, *IEEE Trans. Evol. Comput.* 14 (1) (2010) 150–169.
- [38] X. Li, Efficient differential evolution using speciation for multimodal function optimization, in: *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, New York, NY, USA, ISBN: 1595930108, 2005, pp. 873–880.
- [39] R. Thomsen, Multimodal optimization using crowding-based differential evolution, in: *Proceedings of the 2004 Congress on Evolutionary Computation*, vol. 2, 2004, pp. 1382–1389.
- [40] J.-P. Li, M.E. Balazs, G.T. Parks, P.J. Clarkson, A species conserving genetic algorithm for multimodal function optimization, *Evolut. Comput. (ISSN: 1063-6560)* 10 (3) (2002) 207–234.
- [41] A. Petrowski, A clearing procedure as a niching method for genetic algorithms, in: *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 798–803.
- [42] D.E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and their Application*, Lawrence Erlbaum Associates Inc, USA, ISBN: 0805801588, 1987, pp. 41–49.
- [43] O.M. Shir, Niching in evolutionary algorithms, in: *Handbook of Natural Computing*, Springer Berlin Heidelberg, ISBN: 978-3-540-92910-9, 2012, pp. 1035–1069.
- [44] N. Hansen, The CMA evolution strategy: a comparing review, *Towards New Evolut. Comput.* (2006) 75–102.
- [45] N. Hansen, A. Ostermeier, Convergence properties of evolution strategies with the derandomized covariance matrix adaptation: The $(\mu/\mu_I, \lambda)$ -CMA-ES, in: *Proceedings of 5th European Congress on Intelligent Techniques and Soft Computing*, 97, Citeseer Press, 1997, pp. 650–654.
- [46] N. Hansen, S.D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), *Evolut. Comput.* 11 (1) (2003) 1–18.
- [47] A. Kumar, R.K. Misra, D. Singh, Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase, in: *2017 IEEE Congress on Evolutionary Computation*, 2017, pp. 1835–1842.
- [48] J. Kennedy, R. Eberhart, Particle swarm optimization, in: *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, IEEE Press, 1995, pp. 1942–1948.
- [49] N. Hansen, The CMA evolution strategy: A tutorial, 2016, arXiv preprint arXiv: 1604.00772.
- [50] H. Liu, X.-W. Zhang, L.-P. Tu, A modified particle swarm optimization using adaptive strategy, *Expert Syst. Appl.* 152 (2020) 113353.
- [51] Z. Meng, Y. Zhong, G. Mao, Y. Liang, PSO-sono: A novel PSO variant for single-objective numerical optimization, *Inform. Sci.* 586 (2022) 176–191.
- [52] X. Zhang, H. Liu, T. Zhang, Q. Wang, Y. Wang, L. Tu, Terminal crossover and steering-based particle swarm optimization algorithm with disturbance, *Appl. Soft Comput.* 85 (2019) 105841.
- [53] C. Wang, Y. Liu, Y. Chen, Y. Wei, Self-adapting hybrid strategy particle swarm optimization algorithm, *Soft Comput.* 20 (12) (2016) 4933–4963.
- [54] T. Li, J. Shi, W. Deng, Z. Hu, Pyramid particle swarm optimization with novel strategies of competition and cooperation, *Appl. Soft Comput.* 121 (2022) 108731.
- [55] 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, *KanGAL Report* (2005).
- [56] P.E. Gill, E. Wong, Sequential quadratic programming methods, in: *Mixed Integer Nonlinear Programming*, Springer, 2012, pp. 147–224.
- [57] N. Karmarkar, A new polynomial-time algorithm for linear programming, *Combinatorica* 4 (5) (1984) 373–395.
- [58] A. Forsgren, P.E. Gill, M.H. Wright, Interior methods for nonlinear optimization, *SIAM Rev.* 44 (4) (2002) 525–597.