Highlights

In Search of Exploration and Exploitation Balance: E-AHPSO, A Novel PSO-based Ensemble Algorithm

Fevzi Tugrul Varna, Phil Husbands

- Comprehensive preliminary experiments were conducted to select the most promising ensemble combination for E-AHPSO.
- The performance of E-AHPSO was verified on CEC'13/14/17/20 test suites against 39 highly competitive benchmark algorithms at dimensions 10, 30, 50 and 100.
- E-AHPSO ranked 1^{st} on 100-dimensional CEC'14 and CEC'17 problems.
- E-AHPSO obtained the average final ranks (across all tested dimensions) of 4.5, 4, 3.2 and 5.5 for CEC'13, CEC'14, CEC'17 and CEC'20 test suites, respectively, against 39 benchmarks comprised of PSO and non-PSO algorithms.
- Analysis indicates highly consistent performance across different problem types (multimodal, hybrid, composition) and dimensions without the need for parameter tuning.
- E-AHPSO possesses lower time complexity than 64% (average across all dimensions) of the 39 benchmark algorithms.

In Search of Exploration and Exploitation Balance: E-AHPSO, A Novel PSO-based Ensemble Algorithm

Fevzi Tugrul Varna, Phil Husbands

AI Group, University of Sussex, United Kingdom

Abstract

This paper proposes a novel PSO-based ensemble algorithm, E-AHPSO, utilising several previously published simpler off-the-shelf PSO variants collaborating with the recently published bio-inspired Altruistic Heterogeneous Particle Swarm Optimisation algorithm (AHPSO). The collaborating algorithms in the proposed method were selected from a pool of candidate algorithms after extensive preliminary experiments to determine the optimal combination and ordering of the constituent algorithms in the ensemble to best complement AHPSO. E-AHPSO incorporates the powerful exploration capabilities of AHPSO with the exploitation capabilities of the simpler PSO variants. As a result, the collaboration provides a highly effective general-purpose optimisation algorithm capable of tackling diverse problem types, complexities and sizes without parameter tuning. E-AHPSO was comprehensively tested using the CEC'13, CEC'14, CEC'17 and CEC'20 problem suites on dimensions 10, 30, 50, and 100 against 39 highly competitive comparator algorithms, consisting of 15 well-known PSO variants, 14 recent PSO variants (published after 2020), and 10 non-PSO variants including Differential Evolution variants (IEEE CEC competition winner L-SHADE, CoDE, and MTDE). The proposed algorithm ranked 1^{st} on the 100-dimensional CEC'14 and CEC'17 problems against the 39 benchmarks. Moreover, E-AHPSO was ranked in the top 5 for all four test suites averaged across all dimensions. The novel algorithm performed consistently well across all problem types and dimensions. Analysis of E-AHPSO revealed that the convergence and population diversity patterns were comparable to or better than those of the top-performing comparator algorithms. Moreover, time complexity analysis showed that despite its effectiveness, E-AHPSO was more efficient than 64% of the benchmark algorithms on average (across all dimensions) and compared very favourably

with the other high-performing algorithms.[1]

Keywords: Swarm intelligence; Particle swarm optimisation; Ensemble algorithm; PSO-based ensemble algorithm; general-purpose optimisation algorithm

1. Introduction

The No Free Lunch Theorem [1] tells us that it is not possible to design an optimisation algorithm that, on average, performs better than all other algorithms in solving problems of all kinds. In other words, a completely general-purpose infallible search algorithm is impossible. However, the development of optimisation algorithms that are highly effective over a wide range of problems, including numerous practical real-world problems, is possible and remains an important goal for many researchers in the swarm-based optimisation field. The ideal is an algorithm that can be applied effectively to a large class of problems straight out of the box, with no tuning or tweaking required.

The recently published powerful bio-inspired Altruistic Heterogeneous Particle Swarm Optimisation algorithm (AHPSO) [2, 3] has many of the attributes necessary for such a competitive general-purpose search algorithm for the class of single-objective continuous value optimisation problems. However, although the algorithm has relatively low time complexity, has been shown to perform very well on several classes of real-world problems [4, 5, 6, 7] and on multiple demanding test suites (CEC05, CEC13, CEC14 and CEC17 across 4 different dimensions) [3], with particularly good performance on complex high-dimensional problems, it does not perform as well as some competitors on some kinds of lower-dimensional problems. After analysing potential reasons for this shortcoming, we hypothesized that a specific kind of ensemble algorithm, built around AHPSO, would address this issue and further increase general performance. The resulting algorithm would thus be a highly competitive general-purpose optimisation method for the class of single-objective continuous value problems. The development of such an ensemble algorithm is the subject of this paper.

As discussed in more detail in the next section, over the past decade or so there has been a noticeable expansion of research in ensemble techniques in PSO and population-based search [8, 9, 10]. There are numerous different approaches to combining multiple methods, or parts of methods, into an ensemble [11, 12, 10, 13, 14]; but in most cases the constituent algorithms are treated as equal, in the sense that there is no specific central algorithm among the constituents, about which the others revolve. In contrast, the ensemble algorithm proposed here is explicitly based around AHPSO.

Analysis and extensive experimentation have demonstrated how the various interacting mechanisms inherent in AHPSO are very good at maintaining diversity, making it highly explorative, while still allowing efficient convergence to very good solutions on complex high-dimensional problems [15][3][16]. However, there can be an imbalance between exploration and exploitation on some kinds of lower-dimensional problems, leading to the less efficient performance mentioned above. We theorise that the core reason for this limitation is the heterogeneous nature of the AHPSO algorithm. This property powers AHPSO's highly explorative nature, enabling the discovery of promising regions, but limits the algorithm's ability to converge as quickly as simpler exploitation-focused algorithms on some relatively simple, low-dimensional problems, such as mono-modal functions. Overall, this results in highly efficient performance for more complex problems and adequate performance in low-dimensional search spaces. Hence, in the ensemble method proposed in this paper, our aim was to investigate AHPSO collaborating with several relatively simple off-the-shelf PSO variants, which are expected solely to complement the highly explorative AHPSO, aiding in exploitation and convergence. In contrast to AHPSO, the performance of most "simpler" PSO variants diminishes when applied to high-dimensional complex problems, while exhibiting more efficient performance on low-dimensional problems due to their effective exploitation of the best-known solution. Hence, it is reasonable to suggest that such a collaboration should provide the ultimate balance of exploration and exploitation required for a powerful general-purpose method. This hypothesis is supported by a previous study on DE-based ensemble algorithms [17] that indicates that population-based ensembles that incorporate different search strategies for exploration and exploitation significantly improve overall performance. In essence, the new ensemble algorithm utilises AHPSO as the *central mechanism*, whilst the other methods complement it.

Our primary aim is to develop a general-purpose algorithm that is highly competitive across a very wide range of problems and problem dimensions, without any need for alterations or parameter tuning. Instead of employing a selection mechanism, on-line feedback mechanism, probabilistic approach or other adaptive mechanisms within the ensemble to switch or control the operation of its constituent algorithms, we decided to investigate a very simple ensemble approach in which the constituent algorithms are executed sequentially in a fixed order. We adopted an experimental approach to empirically determine the best constituent algorithms and their optimal ordering through rigorous preliminary experiments. The pool of possible constituent PSO algorithms chosen for this study to complement AHPSO comprised the following 7 mature PSO variant: BBPSO[18], LPSO[19], HPSO-TVAC[20], FIPS[21], UPSO[22], MiPSO[23] and MaPSO[23], from which a subset was determined after the preliminary experiments. Each algorithm was chosen based on its simplicity and exploitation capability. It is worth emphasising that these preliminary experiments had the sole purpose of identifying the subset of algorithms that are optimally complementary to AHPSO and the optimal order in which to execute them, so as to efficiently balance the exploration and exploitation phases.

We note that most ensemble algorithms in the literature are designed to enhance the generalisation capability of the constituent algorithms. However, many studies do not include rigorous testing on different test suites and dimensions to verify the scope and level of generalisation capability of the proposed algorithm. In our work, we address this limitation by testing the proposed ensemble algorithms on the CEC'13, CEC'14, CEC'17 and CEC'20 benchmark test suites on 10, 30, 50 and 100-dimensional problems against a large, competitive and diverse list of comparator algorithms. The comparators comprise 14 well-known mature PSO variants, 15 recent PSO variants (published after 2020) and 3 DE variants (L-SHADE[24], CoDE[25] and MTDE[26]), a total of 39 benchmark comparator algorithms, including ensemble methods. It is worth noting that BBPSO, LPSO, HPSO-TVAC, FIPS, and UPSO have been thoroughly studied, including in relation to ensembles, and applied to real-world problems [27][28][29][30][31], whereas AHPSO, MiPSO and MaPSO are less explored and have never been investigated in the context of ensemble algorithms.

The paper is structured as follows: section 2 provides a review of ensemble methods in PSO and other population-based approaches to optimisation. Following that, section 3 details the new ensemble algorithm. Sections 4, 5, and 6 describe the preliminary experiments, main experiments and analysis of the proposed algorithm. The paper concludes with a discussion and summary of the findings.

2. PSO and Population-based Ensembles Review

Once it was discovered that the original PSO algorithm [32] had a tendency to converge prematurely on some problems, the algorithm was extended by introducing the inertia weight parameter [33]. This version of the algorithm, described by Equations 1-2, is often thought of as the canonical PSO algorithm. Particles are initially randomly distributed in the search space; subsequently, they move according to the rules encapsulated by the governing equations, which are as follows:

$$\vec{v}_i^{t+1} = \omega \vec{v}_i^t + c_1 \vec{r}_1 (\vec{pbest}_i^t - \vec{x}_i^t) + c_2 \vec{r}_2 (\vec{gbest} - \vec{x}_i^t) \tag{1}$$

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \vec{v}_i^{t+1} \tag{2}$$

where ω, c_1 and c_2 are control parameters, namely the inertia weight and acceleration coefficients, $\vec{v}_i^t$ is the i^{th} particle's velocity, $\vec{x}_i^t$ is its current position, $\vec{pbest}_i$ is its personal best position, and $\vec{gbest}$ is the current best known global solution. $\vec{r}_1$ and $\vec{r}_2$ are random n-dimensional vectors in the range $[0, 1]$, where n is the dimension of the search problem.

Much of the PSO literature on improvements to the basic canonical algorithm can be divided roughly into four categories: *parameter tuning*, *neighbourhood topologies*, *learning strategies* and *hybridisation*. The latter category includes most ensemble approaches, which will be the main focus of this paper.

In order to improve overall performance and control convergence speed, many of the early variants of PSO involved alterations to the inertia weight parameter [33][22][34][35], including hybrid models [36][37] that utilised additional search techniques to regulate the value of this parameter. Various other adaptive methods, where the parameter varied over time, were introduced [38][39][40][41][20]. Similar approaches have been investigated for the other two main parameters of the canonical PSO, the acceleration parameters c_1 and c_2 (Eqn 1), which maintain the balance between local and global search [20][39][42].

Neighbourhood topology plays a significant role in controlling the information flow in a PSO swarm, and such control can delay loss of population diversity, reducing the algorithm's potential to converge prematurely [43]. Hence, there has been much research on utilising different topologies. [44] investigated the performance impact of various topologies, namely circle, star, wheel and random neighbourhoods, on the canonical PSO algorithm, while other studies investigated a variety of other static topologies [45][46][47][48][49]. [50] introduced the idea of search guidance by a neighbourhood local best (*lbest*) exemplar; search was initiated with *lbest* guidance while guidance from the global best (*gbest*) exemplar was gradually introduced to the entire population. This prevented the entire swarm from continuously exploiting the *gbest* exemplar, thereby contributing to the preservation of population diversity. [51] introduced a dynamic tournament topology that utilises several best particles for guidance as opposed to the *lbest* or *gbest* exemplars. [52] introduced the idea of dynamic topologies by dividing the population into multiple co-evolving sub-swarms, and [53][54][55] extend the multi-swarm mechanism.

In the canonical PSO, a particle's movement is guided by the globally best position, *gbest*, and the particle's own best position, *pbest*. However, different learning strategies have been published that develop more complex guidance strategies to target various limitations of the standard PSO algorithm. For example, comprehensive learning PSO [56] and its extension, heterogenous CLPSO [57], were proposed to deal with fast-depleting population diversity and to enhance exploration and exploitation. Self-learning mechanisms [58], generalised opposition-based learning [59], elitist and orthogonal learning strategies [60], dimensional learning [61], and social learning [62] strategies were all introduced to improve overall performance and to deal with premature convergence. More recent work extending such research includes [63] [64] [65] [66] [67] [68] [69] [70].

Another notable direction for the improvement of PSO is the hybrid, or ensemble, approach, which will be the focus for the rest of this paper. The fundamental idea of ensemble algorithms is to combine different algorithms or components of algorithms to enhance a particular aspect (e.g., convergence, exploration, time complexity) or the overall performance of the baseline algorithms. Some of the earlier PSO ensembles published in the literature include MEPSO [71], introduced for dynamic optimisation. MEPSO divided the particle population into two groups with separate strategies. HPSO and f_k-PSO [11] [12] utilised different PSO variants to create a pool of behaviours. In the case of HPSO, the current active behaviour is randomly selected from the pool, while f_k-PSO employs the idea of a success counter for each behaviour to increase the selection probability of the best-performing behaviours. Other ensemble algorithm selection strategies were introduced in [72] [73] [9]. [74] proposed an ensemble PSO to optimise the hyper-parameters of a classification model. The algorithm employed an adaptive mechanism to generate subgroups, and each subgroup utilised two learning strategies to guide the particles. ELPSO [8] introduced a high-level ensemble that employed three PSO algorithms, GPSO, LPSO and BBPSO, each with its own subpopulation. An ensemble learning mechanism was used to guide the overall search by obtaining information from the different populations.

At this point, it is worth revisiting the term "ensemble" and its usage in the literature. Study [10] defines the ensemble concept as the incorporation of different strategies, operators, parameter values and algorithms that employ either single or parallel populations. Although in the research described in the current paper, the term "ensemble" is used throughout to refer to a particular type of ensemble approach, involving multiple separate search algorithms, it is generally used as an umbrella term in the literature. For instance, study [75] includes multi-method [72], multi-strategy [13], mimetic [76], hybrid [77] and hyper-heuristic [13] algorithms within the concept of ensemble. Some of these categories are relevant to the work presented in this paper and are worth highlighting. For example, multi-method and multi-strategy are specifications of low-level and high-level ensembles, respectively. A low-level ensemble refers to the incorporation of multiple algorithmic components such as topologies, search behaviour and parameters, whereas a high-level ensemble refers to an ensemble of different algorithms or variants. In general terms, the work presented in this paper is in line with the high-level ensemble paradigm, although the central algorithm in the ensemble, AHPSO, could itself be regarded as a low-level ensemble algorithm as it incorporates multiple behaviours and strategies. It is outside the scope of this paper to review the entire ensemble literature; however, the two ensemble types, low-level and high-level, will be discussed further next.

2.1. Low-Level Ensembles

In theory, versatile population-based algorithms that can adopt different search strategies are expected to exhibit better generalisation capability: an ability to perform well on a range of different problem types. This is due to the heterogeneous agent behaviours in the population which provide a level of coping mechanism for different problem types and complexities. As briefly mentioned earlier, the low-level ensemble concept exploits the idea of incorporating different algorithmic components that complement each other. For instance, while it is common practice to incorporate a mutation operator within a single strategy/homogeneous population-based algorithm to enhance its exploration capability, many DE variants involve ensembles with multiple mutation operators [13], [78], [79], [80], [81], [25], [82]. In the case of low-level PSO ensembles, variants with multiple strategies [83] [84], coevolutionary dynamics [85], hybrid methods [86], self-adaptation [87] [58] and reinforcement learning-based adaptation [88] have been introduced.

2.2. High-Level Ensembles

In essence, the high-level ensemble paradigm utilises a number of (typically) specialised constituent algorithms to create a more general-purpose algorithm. Some of these ensemble algorithms use a single population where multiple algorithms act in sequence, others employ multiple populations with different algorithms acting in parallel. For instance, [14] proposed EEA-SLPS that incorporates three self-adaptive learning-based strategies. The algorithm divides the population into three equal sub-populations, one for each of the constituent search methods, which operate in parallel. The work also investigated various information exchange mechanisms to determine the optimal information flow between the constituent algorithms. Study [72] emphasised the limitation of using of a single universal genetic operator and embraced the concept of self-adaptive multi-method optimisation where multiple algorithms were employed simultaneously. In addition, an information exchange mechanism was used to enable the constituent algorithms to learn from each other. MultiEA [89] was proposed to enable the composition of a suitable ensemble from a pre-defined pool of algorithms, where the algorithm returns a 'portfolio' of the best-ranked algorithms selected from the pool. This method can be advantageous in certain scenarios as it does not require parameter control. However, it can be argued that the algorithm requires a test suite that is a *reference model* for the given problem. This is particularly an issue in the case of complex real-world problems, as it can be challenging to find a suitable representation to run an adequate test and build a portfolio. This type of approach can also be time-consuming, requiring portfolio generation for every new problem. [90] proposed an MVC (multi variants coordination) framework. The search process is initiated by dividing the population into non-overlapping equal segments, and different evolutionary algorithm variants are executed independently. Subsequently, MVC is utilised to maintain promising solutions obtained by the multiple algorithms. [91] introduced an ensemble algorithm consisting of modified CoDE and jADE algorithms, exploiting the distinct performance properties of jADE and CoDE. It was found that jADE is particularly suited for solving unimodal and simple multimodal functions due to its comprehensive exploitation strategy. Similarly, CoDE employs exploration mutation strategies, making it suitable for complex multimodal functions. As a result, the proposed ensemble algorithm had improved the generalisability relative to the constituent algorithms by combining their merits. [92] investigated the impact of the cooperative dynamics provided by incorporating four differential evolution variants and [93] proposed a novel DE-based ensemble that employed jADE, CoDE and EPSDE. The search process initiates with the division of the population into four segments including three small indicator populations, one for each DE variant, and one large reward population. At a pre-defined interval, the best-performing constituent DE variant is assigned the reward population. [94] proposed a PSO-based ensemble, EPSO, that employs the basic canonical inertia weight PSO, CLPSO, FDR-PSO, HPSO-TVAC and LIPS. EPSO divides the population into a small sub-population and a larger sub-population. In the small sub-population, CLPSO is utilised on its own to maintain diversity; in the larger sub-population, PSO, CLPSO, FDR-PSO, HPSO-TVAC, and LIPS act as an ensemble. In the larger sub-population, on each iteration, a particle is updated by selecting one of the ensemble algorithms according to a probability based on the algorithm's performance over the past few iterations. The ensemble's constituent algorithms were chosen based on their effectiveness on different problem types. Thus, inertia weight PSO and HPSO-TVAC are included as they are more efficient on unimodal problems, while FDR-PSO, CLPSO and LIPS are included for their ability on multimodal problems.

The EPSO algorithm is related to the work proposed in the current paper, both in terms of being a PSO-based ensemble and regarding the algorithms that comprise it. But there are fundamental differences: EPSO utilises the constituent algorithms based on their effectiveness on different problem types and employs a probabilistic selection method to determine which methods in the ensemble are used. In contrast, we take a different approach and instead focus on designing an ensemble model with constituent algorithms that complement AHPSO (which is always the central ensemble algorithm) to achieve a better exploration and exploitation balance. In our approach the algorithms in the ensemble are applied in sequence, rather than being selected probabilistically. Furthermore, the suitability of constituent algorithms was determined by conducting rigorous preliminary experiments to identify the best *exploitative* algorithms and their best ordering to complement the *explorative* AHPSO algorithm.

Finally, with regard to the literature categorisation of the work presented in this paper, the proposed algorithm can be classified as a high-level single-population ensemble.

3. The Proposed Ensemble Algorithm

This section describes the novel ensemble algorithm built around AHPSO, providing the necessary background on the PSO variants in the ensemble. The general performances of the simpler PSO algorithms in the pool of potential ensemble constituents are typically insufficient to handle high-dimensional and more complex problem types due to a lack of sophisticated stochastic components and/or heterogeneity. Our proposal is to adopt these algorithms as individual mechanisms for exploiting the solutions discovered in more complex solution spaces by the more sophisticated heterogeneous AHPSO algorithm.

An ensemble consists of AHPSO plus some subset of the pool of possible constituents, to be executed sequentially in some fixed order. Detailed preliminary experiments, as described in Section 4, were performed to determine the most promising combinations and orderings for this simple ensemble model. The most promising ensemble algorithm that emerged from the rigorous preliminary experiments we named E-AHPSO. This algorithm was then further studied in detailed comparative experiments to validate its generalisation capabilities, as described in Section 5.

The seven possible PSO variants are consigned to the role of exploitation. In contrast, AHPSO plays the central role of exploring the search space and maintaining population diversity. The main hypothesis we set out to explore was that AHPSO would find highly promising solutions and pass them to the simpler algorithms to exploit, producing a powerful general-purpose ensemble. Below, AHPSO and the algorithms in the pool are described.

3.1. Altruistic Heterogeneous Particle Swarm Optimisation (AHPSO)

AHPSO is a recent bio-inspired PSO variant that introduced two particle models, namely the *altruistic particle model* (APM) and the *paired particle* *model* (PPM). The APM constitutes the primary behaviour model for particles, whereas PPM is directed at maintaining better population diversity. In the AHPSO algorithm, particles are conceptualised as energy-driven entities that are either active or inactive, depending on their current energy-level activation threshold. Hence, a particle is active if $E_{current} > E_{activation}$ and inactive otherwise. The concept of *activation* is utilised to regulate the selection of the movement strategy for particles, which directly controls the swarm's heterogeneity.

3.1.1. Altruistic Particle Model

The APM mechanism assumes that the particles tend to be active; hence, particles in the inactive state are expected to borrow energy from the swarm in order to become active. The sort of altruistic lending behaviour this requires is commonly observed in nature [95, 96, 97], and in this context, it allows stochastic interactions among particles, causing oscillations in the ratio of active and inactive particles within the swarm, which further contributes to the regulation of heterogeneity in the swarm. The idea behind the lending-borrowing ecology is that the circulating energy in the swarm is assumed to be a resource particles can use and contribute towards. The level of altruistic behaviour of the i^{th} particle is represented by the variable A_i which is calculated using:

$$A_i = \frac{L_i^t}{B_i^t} \tag{3}$$

where L_i^t and B_i^t are the number of times the i^{th} particle lent and borrowed energy, respectively, to and from other particles up to time t. Persistent borrowing behaviour in a particle over prolonged periods results in a volatile lending-borrowing ratio and reduces the particle's altruism value (as the particle consumes a lot more resources than it contributes to the swarm). When a particle is inactive it attempts to borrow energy from randomly selected potential lenders. The selected potential lender particles expect the energy-requesting particle to meet an altruism criterion defined by ϕ as described in Equations 4-5.

$$\phi = A_i \times P_2 \tag{4}$$

$$P_2 = \frac{\delta}{N} \tag{5}$$

where A_i is the borrower particle's altruism, δ is the number of particles in the swarm with active status at time t, and N is the population size. P_2 provides a rough measure of the probability that lent energy will be returned to the swarm. In addition to enforcing altruistic behaviour, the criterion ϕ provides a form of altruistic assessment of the borrower particles' probability of returning lent energy. If $\phi \geq \beta$, where β is the average altruism value in the swarm, an equal amount of energy is borrowed from each lender particle to provide the required energy of the borrower particle. This is calculated as:

$$E_{required} = \frac{(E_{activation} - E_{current})}{\alpha} \tag{6}$$

where $E_{required}$ is the amount of energy required from each lender and α is the number of selected lenders.

The altruistic behaviour of particles strongly influences the movement strategy in the AHPSO. Hence, active particles use the canonical PSO update equations shown in Eq. 1 and Eq. 2. In contrast, inactive particles that do not satisfy the criterion ϕ, and therefore cannot borrow energy, use the following equation to update velocity:

$$\vec{v}_i^{t+1} = \omega \vec{v}_i^t + c_1 \vec{r}_1 (\vec{pbest}_i^t - \vec{x}_i^t) + c_2 \vec{r}_2 (\vec{pbest}_{minA}^t - \vec{x}_i^t) \tag{7}$$

where $\vec{pbest}_{minA}^t$ is the personal best position of the least altruistic particle at time t. Particles that do not meet the criterion ϕ are less altruistic at time t and are influenced by similarly less altruist particles. Hence, in this case, guidance towards the least altruistic particle partially enables cooperation on an altruistic basis and supports heterogeneity. Considering the evolutionary characteristics of particles, the least and most altruist particles fluctuate. Energy sharing occurs between the lender particles and the borrower particle that satisfies the criterion ϕ. Post energy-borrowing, the borrower particle can be in two possible states. In the first case, the borrower has borrowed sufficient energy to be active. However, this is not always the case as lenders are randomly selected without any criteria; hence, there is a distinct possibility that some lenders do not have excess energy to lend. Therefore, some particles may still lack sufficient energy to activate even after borrowing energy. In this case, an exemplar for the particle is generated using the lender particles' mean positions. In this case, the velocity is updated as follows:

$$\vec{v}_i^{t+1} = \omega\vec{v}_i^t + c_1\vec{r}_1(\vec{pbest}_i^t - \vec{x}_i^t) + c_2\vec{r}_2(\vec{x}_{mean}^t - \vec{x}_i^t) \quad (8)$$

where $\vec{x}_{mean}^t$ is the mean position of $\frac{\alpha}{2}$ randomly selected lender particles. As commonly seen in certain PSO variants, in our behavioural model, particles do not explicitly exchange positional information, hence, by using the mean position of a proportion of lender particles, we aim to form an implicit communication between lender-borrower particles.

In the first case, where the borrower particle succeeds in borrowing sufficient energy to activate, the particle's velocity is calculated using:

$$\vec{v}_i^{t+1} = \omega\vec{v}_i^t + c_1^t\vec{r}_1(\vec{pbest}_i^t - \vec{x}_i^t) + c_2^t\vec{r}_2(\vec{pbest}_{maxA}^t - \vec{x}_i^t) \quad (9)$$

where $\vec{pbest}_{maxA}^t$ is the personal best position of the most altruistic particle at time t.

3.1.2. Paired Particle Model

The paired particle model (PPM) complements the APM detailed in the previous section. Previous studies [15] indicate that the PPM triggers periodic small improvements in population diversity. The PPM follows the APM and randomly employs one of the two movement strategies on a small proportion of the population, namely, the coupling-based and opposition-based strategies.

Coupling-Based Strategy. The pairs in the coupling-based strategy are distinguished as tightly, loosely coupled, or neutral, which determines the movement strategy. The following rules determine the state of the pairs:

1. *tightly coupled* if both particles are active at time t.
2. *loosely coupled* if both particles are inactive at time t.
3. *neutral* if one particle is active and the other is inactive at time t.

Based on the rules stated above, the tightly and loosely coupled particles update their velocities using Eqs. 10 and 11, respectively.

$$\vec{v}i^{t+1} = \omega\vec{v}i^t + c_1\vec{r}1((\vec{x}_{pair}^i \times E_{current}^i) - \vec{x}_i^t) + \\ c_2\vec{r}_2((\vec{pbest}_{pair}^i \times E_{current}^i) - \vec{x}_i^t) \quad (10)$$

$$\vec{v}_i^{t+1} = \omega \vec{v}_i^t + c_1 \vec{r}_1((\vec{pbest}^i_{pair} \times E^i_{current}) - \vec{x}_i^t) + \\ c_2 \vec{r}_2((\vec{x}^i_{pair} \times E^i_{current}) - \vec{x}_i^t) \quad (11)$$

where $\vec{x}^i_{pair}$ and $\vec{pbest}^i_{pair}$ are the i^{th} particles' pair position and personal best position, respectively, and $E^i_{current}$ is a damping factor to minimize rapid oscillations.

Opposition-Based Strategy. The opposition-based strategy guides both particles of a pair in potentially distinct directions to maintain diversity within pairs, leading to a continued revamp of population diversity within a proportion of the population. The altruism value of the paired particles is utilised to determine the type of movement particles adopt, and if the i^{th} particle attains a greater altruistic value than its coupled pair, Eq. 12 is used to update its velocity; otherwise, Eq. 13 is used to update the velocity.

$$\vec{v}_i^{t+1} = \omega \vec{v}_i^t + c_1 \vec{r}1(\vec{pbest}_i - \vec{x}_i^t) + c_2 \vec{r}2(\vec{x}^{pair}_{maxA} - \vec{x}_i^t) \quad (12)$$

$$\vec{v}_i^{t+1} = \omega \vec{v}_i^t + c_1 \vec{r}_1(\vec{pbest}_i - \vec{x}_i^t) + c_2 \vec{r}_2(\vec{x}^{pair}_{minA} - \vec{x}_i^t) \quad (13)$$

Where $\vec{x}^{pair}_{maxA}$ and $\vec{x}^{pair}_{minA}$ are randomly selected as either personal best or the position of the most altruistic and least altruistic pair at time t, respectively.

The complete pseudo-code for the AHPSO algorithm is shown below, and the MATLAB source code of the AHPSO algorithm can be obtained from github.com/ftvarna/AHPSO.

Below we briefly describe each of the simpler PSO algorithms investigated as possible ensemble constituents. Each was chosen for its exploitation capabilities and potential to complement AHPSO in the ensemble.

3.2. Barebones PSO

The trajectory of particles and the influencing factors have been a major focus of research since the proposal of the PSO algorithm. In [18], Kennedy performed an ablation experiment on the canonical PSO to strip away various traditional features to reveal the key influencing factors on trajectory. As a result, BBPSO was proposed, a new, more compact variant without the velocity terms. In contrast to the canonical PSO, BBPSO updates *pbest* and

Algorithm 1: AHPSO

population size $n, C = 0.15, \omega_{max} = 0.99, \omega_{min} = 0.2$;
$c_1 = 2.5 - (1 : T_{max}) * 2/T_{max}$
$c_2 = 0.5 - (1 : T_{max}) * 2/T_{max}$
$\omega_1^t = \frac{\omega_{max} + (\omega_{min} - \omega_{max})}{1+exp(-5(\frac{2t}{T_{max}}-1))}$
$E_{current}^{1..n} = U(0.1, 1, [1\ n])$
$E_{activation}^{1..n} = U(0.5, 1, [1\ n])$
$\alpha = \lfloor n * 0.2 \rfloor$
for *t=1:*T_{max} **do**
    β=average A value in swarm at t
    δ=number of active particles at t
    for *i=1:n* **do**
        if $f(x_i) \geq \overline{f(x)}$ **then**
            $\omega = \omega_1^t + C$; if $\omega > 0.99, \omega = 0.99,$ *end*
        else
            $\omega = \omega_1^t - C$; if $\omega < 0.20, \omega = 0.20,$ *end*
        end
        if $E_{current} \geq E_{activation}$ **then**
            update $\vec{v}_i$ and $\vec{x}_i$ using Eqs. 1 and 2
        else
            calculate A_i^t, P and ϕ using Eqs. 3, 4 and 5
            if $\phi < \beta$ **then**
                update $\vec{v}_i$ and $\vec{x}_i$ using Eqs. 8 and 2
            else
                randomly select α potential lenders
                calculate $E_{required}$ using Eq.7
                deduct $E_{required}$ from $E_{current}$ of each lender
                update $E_{current}$ of the borrower
                increment L by one for all lenders
                increment B by one for the borrower
                if $E_{current} \geq E_{activation}$ **then**
                    update $\vec{v}_i$ and $\vec{x}_i$ using Eqs. 9 and 2
                else
                    update $\vec{v}_i$ and $\vec{x}_i$ using Eqs. 8 and 2
                end
            end
        end
        if i^{th} *particle is paired* **then**
            if *randi([0 1])==0* **then**
                if *pair is tightly coupled* **then**
                    update $\vec{v}_i$ and $\vec{x}_i$ using Eqs. 11 and 2
                else if *pair is loosely coupled* **then**
                    update $\vec{v}_i$ and $\vec{x}_i$ using Eqs. 12 and 2
                end
            else
                if $A_i > A_{couple}$ **then**
                    update $\vec{v}_i$ and $\vec{x}_i$ using Eqs. 13 and 2
                else
                    update $\vec{v}_i$ and $\vec{x}_i$ using Eqs. 14 and 2
                end
            end
        end
        evaluate the fitness of $\vec{x}_i$
        update the $\vec{pbest}_i$ and $\vec{gbest}$
    end
    reinitialise $E_{current}^{1..n}$ and $E_{activation}^{1..n}$
end

$gbest$ by utilising random numbers generated from the Gaussian distribution as:

$$X_{ij} = N\left(\frac{pbest_{ij} + gbest_j}{2}, |pbest_{ij} - gbestj|\right) \tag{14}$$

Where N denotes the probability density function of the Gaussian distribution with mean $(\frac{pbest_{ij}+gbest_j}{2})$ and standard deviation $(|pbest_{ij} - gbestj|)$.

3.3. Fitness Distance Ratio PSO

Fitness-Distance-Ratio based PSO (FDR-PSO) [98] was proposed to address the premature convergence problem of the canonical PSO algorithm. In contrast to the canonical PSO, where particles are attracted towards the *pbest* and *gbest* positions, the FDR mechanism allows particles to adjust their trajectory towards nearby particles of higher fitness by utilising the ratio of the relative fitness and the distance of other particles. To prevent attraction towards the direction of multiple particle positions, FDR-PSO selects a single particle at a time to update each velocity dimension that is the nearest best (fitness) particle to the particle being updated.

Fitness-Distance-Ratio Calculation:

$$x_{FDR} = \frac{Fitness(pbest_j) - Fitness(x_i))}{|pbest_{jd} - x_{id}|} \tag{15}$$

Subsequently, the velocity is updated using:

$$\vec{v}_i^{t+1} = \omega\vec{v}_i^t + \vec{r}_1(\vec{pbest}_i^t - \vec{x}_i^t) + \vec{r}_2(\vec{gbest} - \vec{x}_i^t) + \vec{r}_3(\vec{x}_{FDR}^t - \vec{x}_i^t) \tag{16}$$

3.4. Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients (HPSO-TVAC)

HPSO-TVAC [20] introduced a new parameter automation strategy: time-varying acceleration coefficients (TVAC). The TVAC strategy aimed to control local search and converge more efficiently by changing the values of the canonical PSO coefficients c_1 and c_2 over time. In essence, the TVAC mechanism reduces the impact of the social exemplar (*pbest*) over time and increases the influence of the cognitive exemplar (*gbest*) by altering the c_1 and c_2 coefficients. This leads to better initial exploration of the search space, and towards the end of the search process, the higher social (c_1) value and lower cognitive (c_2) value leads to more efficient convergence. The mathematical representation of the TVAC mechanism is:

$$c_1 = (c_{1f} - c_{1i}) \frac{t}{T_{max}} + c_{1i} \tag{17}$$

$$c_2 = (c_{2f} - c_{2i}) \frac{t}{T_{max}} + c_{2i} \tag{18}$$

where c_{1i} , c_{1f}, c_{2i} and c_{2f} are the initial and final values of the coefficients, t and T_{max} are the current and maximum number of iterations, respectively. The value of c_1 reduces from 2.5 to 0.5, and the value of c_2 increases from 0.5 to 2.5. HPSO-TVAC is a light-weight PSO variant with good exploitative powers.

3.5. Linearly Decreasing Inertia Weight PSO (LPSO)

LPSO is a well-known simple PSO variant that gradually decreases the inertia weight parameter [33] over the course of the search [19]. [34] further analysed the effects of linearly decreasing inertia weight and concluded that maintaining higher values for the parameter facilitates global search while lower values have the opposite influence on the search behaviour. As a result, an initial value of 0.9 and a final value of 0.4 for the inertia weight converged faster in most cases. In addition, various improvements to the linearly decreasing inertia weight mechanism have been proposed [99] [100] [101]. The following equation gives the calculation of the LDIW as used in this study:

$$\omega_t = (\omega_{start} - \omega_{end}) \left(\frac{T_{max} - t}{T_{max}} \right) + \omega_{end} \tag{19}$$

Where ω_{start} and ω_{end} are the initial and final inertia weight values ($\omega_t \in [0, 1]$), t is the current iteration and T_{max} the maximum number of iterations.

3.6. Macroscopic/Microscopic Adaptive CLPSO (MaPSO/MiPSO)

The MaPSO and MiPSO algorithms [23] are variants of the comprehensive learning PSO (CLPSO) [56] algorithm. The two variants address issues arising from the CLIPSO parameter m which controls the gap between exemplar refreshes. In CLPSO, exemplars are updated using a constant m value, while both MaPSO and MiPSO adaptively regulate the m value. However, MaPSO utilises a single m value for the entire population, whereas MiPSO adaptively determines the m value for each particle. Ultimately, the mechanism utilised in MaPSO provides a better balance of exploration and exploitation, whilst MiPSO is particularly improved in solving more complex multimodal problems. Macroscopic/Microscopic Adaptive CLPSO (MaPSO/MiPSO) utilises a learning automaton for the entire population and for each particle in the swarm. The automata consist of various rules to regulate the aforementioned parameter m through three distinct actions: *increase*,*decrease* and *halt*. The particles continue learning from their exemplars for the regulated m iterations. The reinforcement signal β for MaPSO and MiPSO, respectively, are calculated using the following two equations:

$$\beta = \begin{cases} 0, & gbest^t < gbest^{t-1} \\ 1, & else \end{cases} \tag{20}$$

$$\beta = \begin{cases} 0, & pbest_i^t < pbest_i^{t-1} \\ 1, & else \end{cases} \tag{21}$$

3.7. Unified PSO

The unified PSO (UPSO) [22] introduced a unified scheme that aggregates the local and global properties. The UPSO algorithm utilises a parameter called the unification factor ($u \in [0, 1]$) that determines the impact of the global and local properties. For instance, $u = 1$ essentially equates to sole global influence; likewise, $u = 0$ allows local PSO to dominate the search. Furthermore, for all intermediate values assigned to u, a balance between local and global search is achieved. The velocity update equation for UPSO is as follows:

$$\vec{G}_i^{t+1} = \chi[\vec{v}_i^t + c_1\vec{r}_1(\vec{pbest}_i^t - \vec{x}_i^t) + c2\vec{r}_2(\vec{gbest} - \vec{x}_i^t)] \tag{22}$$

$$\vec{L}_i^{t+1} = \chi[\vec{v}_i^t + c_1\vec{r}_1(\vec{pbest}_i^t - \vec{x}_i^t) + c2\vec{r}_2(\vec{nbest} - \vec{x}_i^t)] \tag{23}$$

$$\vec{U}_i^{t+1} = u\vec{G}_i^{t+1} + (1-u)\vec{L}_i^{t+1} \tag{24}$$

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \vec{U}_i^{t+1} \tag{25}$$

Where u is the unification factor, $\vec{G}$ and $\vec{L}$ denote the velocity update for global and local variants, respectively.

4. Preliminary Experiments: Finding the Most Promising Ensemble

4.1. Selection of PSO Variants and the Number of Algorithms

In this subsection, we present the method used to select PSO variants to collaborate with AHPSO in an ensemble context. The pool of possible ensemble constituent algorithms comprises seven mature, relatively simple PSO variants: BBPSO, LPSO, HPSO-TVAC, FIPS, UPSO, MiPSO and MaPSO. The ensemble model consists of n algorithms: AHPSO and $n-1$ constituent algorithms selected from the pool, to be executed one after the other, each inheriting the population from the previous algorithm in the sequence, and each running for a fixed number of iterations (default $= MaxIterations \times \frac{1}{n}$, where n is the number of constituent algorithms in the ensemble). n was set at 3 after running preliminary experiments where the number of algorithms was systematically varied. Three algorithms gave the best overall mean performance. Increasing the number of algorithms beyond this led to a significant deterioration in performance. We theorise that, as the total number of iterations was equally distributed between the algorithms, this deterioration in performance is a consequence of the algorithms not having a sufficient number of iterations to perform an adequate search as n increases. Hence, in our ensemble model, the AHPSO algorithm collaborates with two algorithms selected from the pool. Each possible ensemble combination is labelled as E_k, as shown in Table 1, which lists all possible triplets of algorithms: AHPSO plus two others from the pool.

4.2. Selection of Ensemble Ordering

Since the ensemble model consists of 3 algorithms: AHPSO (fixed) and two PSO variants from Table 1, there are 6 different possible ordering for each combination, as shown in Table 2. For instance, order O_1 applied to E_3 executes the algorithms in the following sequence: FDR $\rightarrow$ FIPS $\rightarrow$ AHPSO. The resulting ensemble is referred to as $E_3^{O_1}$. Note that the same combination executed with a different order can produce significantly different performance. Hence, it is essential to determine the impact of different execution orders on each combination and to determine if a particular ordering consistently produces notable (better or worse) performance for some, most of, or all of the combinations shown in Table 1.

Table 1: List of possible ensemble algorithm combinations.

Ensemble Key	$Constituent_1$	$Constituent_2$	$Constituent_3$
E_1	BBPSO	HPSO-TVAC	AHPSO
E_2	FDR	BBPSO	AHPSO
E_3	FDR	FIPS	AHPSO
E_4	FDR	HPSO-TVAC	AHPSO
E_5	FDR	LPSO	AHPSO
E_6	FDR	UPSO	AHPSO
E_7	FIPS	BBPSO	AHPSO
E_8	FIPS	HPSO-TVAC	AHPSO
E_9	FIPS	LPSO	AHPSO
E_{10}	FIPS	UPSO	AHPSO
E_{11}	LPSO	BBPSO	AHPSO
E_{12}	LPSO	HPSO-TVAC	AHPSO
E_{13}	UPSO	BBPSO	AHPSO
E_{14}	UPSO	HPSO-TVAC	AHPSO
E_{15}	UPSO	LPSO	AHPSO
E_{16}	MaPSO	BBPSO	AHPSO
E_{17}	MaPSO	FDR	AHPSO
E_{18}	MaPSO	FIPS	AHPSO
E_{19}	MaPSO	LPSO	AHPSO
E_{20}	MaPSO	UPSO	AHPSO
E_{21}	MaPSO	HPSO-TVAC	AHPSO
E_{22}	MiPSO	BBPSO	AHPSO
E_{23}	MiPSO	FDR	AHPSO
E_{24}	MiPSO	FIPS	AHPSO
E_{25}	MiPSO	LPSO	AHPSO
E_{26}	MiPSO	UPSO	AHPSO
E_{27}	MiPSO	HPSO-TVAC	AHPSO
E_{28}	MiPSO	MaPSO	AHPSO

Table 2: List of Ensemble Ordering.

Key			
O_1	$Constituent_1$	$Constituent_2$	$Constituent_3$
O_2	$Constituent_1$	$Constituent_3$	$Constituent_2$
O_3	$Constituent_2$	$Constituent_1$	$Constituent_3$
O_4	$Constituent_2$	$Constituent_3$	$Constituent_1$
O_5	$Constituent_3$	$Constituent_1$	$Constituent_2$
O_6	$Constituent_3$	$Constituent_2$	$Constituent_1$

4.3. Experimental Setup

Our preliminary experiments aimed to find the most promising ensemble (constituent algorithms and corresponding order) to be further investigated in a full comparative study over multiple test suites and against many comparator algorithms. Hence, we experimented with each combination using the six possible orderings (see Tables 1-2) – a total of 168 ensembles – on the CEC'17 test suite at 10, 30 and 50 dimensions, comparing each with a challenging set of 39 benchmark algorithms. The parameter settings of the constituent algorithms, and comparator algorithms, were unchanged on all the experiments; the best published general settings were used in each case (see Table 3). Each problem was tested 51 times, and mean, median, standard deviation, maximum and minimum error values were recorded for each problem.

4.4. Experimental Results

Due to the large amount of data obtained from the experiments, it is impossible to present the complete results in the main body of the paper. However, the raw experimental results are provided in the supplementary material. Since each of the 28 combinations, E_{key} in Table 1, was investigated in 6 different orders, we have selected a single best *version* of each combination with the corresponding best order, denoted as E^{O}_{key}, 28 in all. Selection was based on average performance across all problem types and dimensions, as we aim to develop a general-purpose method. Results for these 28 ensembles are displayed in the plots in this section.

Figure 1 presents a summary of the statistical-significance results for these 28 ensembles at $\alpha = 0.05$ against the 39 highly competitive benchmark algorithms shown in Table 3 on the 10, 30 and 50-dimensional CEC'17 problems. In summary, as shown in Figure 1, we observed a pattern where orderings O_1 and O_2 yield better results for 10-dimensional problems. However, the opposite pattern emerged with increased problem dimension, where O_1 and O_2 provide relatively poor performance.

Generalisation capability was used as the primary selection criterion for choosing the most promising of these 28 ensembles for further detailed investigation.

It is worth briefly assessing the results for the 10-dimensional problems before proceeding further. The combinations that are evidently better on 10-dimensional problems (Figure 1) are highly unlikely to have even reasonable generalisation capability to handle more complex scenarios. This

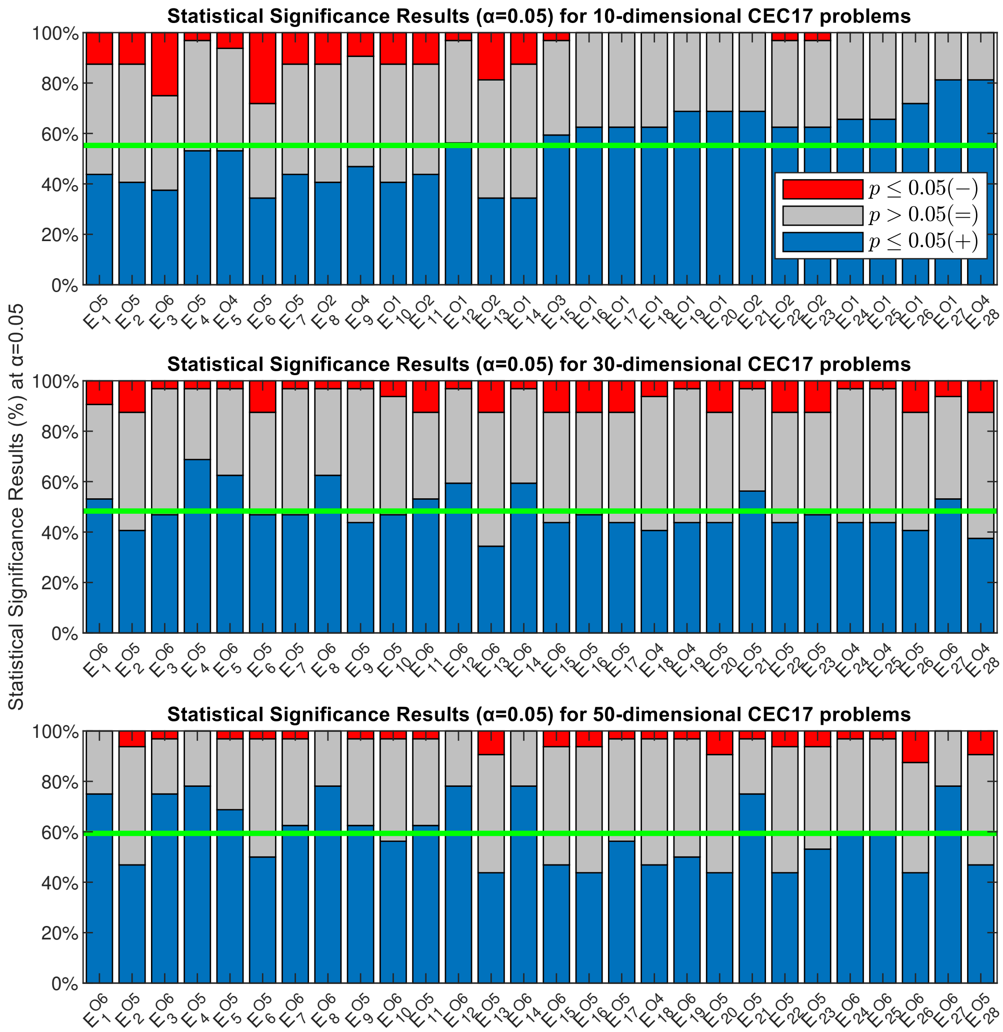

Figure 1: Summary of the Wilcoxon signed-rank test results for the best-performing ensemble derived from each of the 28 candidate combinations on the 10-, 30-, and 50-dimensional CEC'17 problems. The bars show the percentages of comparator algorithms against which the tested ensemble is statistically significantly better (+), not statistically significantly different (=), or statistically significantly worse (−) at a significance level of $\alpha = 0.05$.

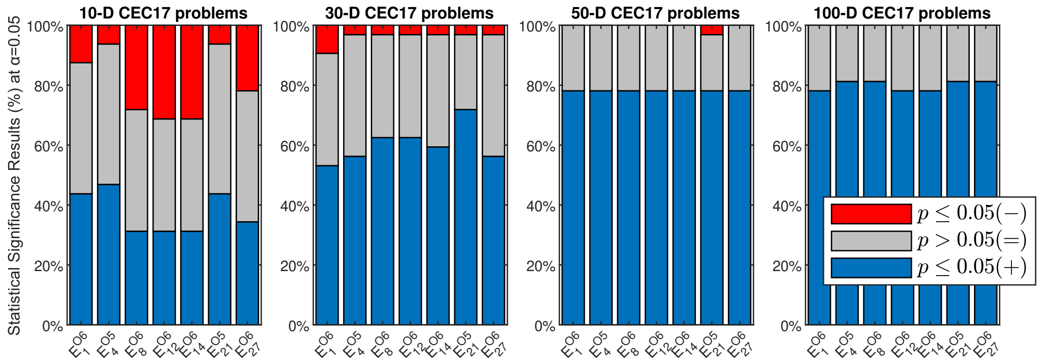

Figure 2: Summary of the statistical significance results ($\alpha = 0.05$) for the 7 short-listed combinations when the proportion of iterations allocated to AHPSO is randomly set (at 50%, 60% or 70% of T_{max}). The symbols +, =, and − indicate significantly better performance, no statistically significant difference, and significantly worse performance, respectively, at $\alpha = 0.05$.

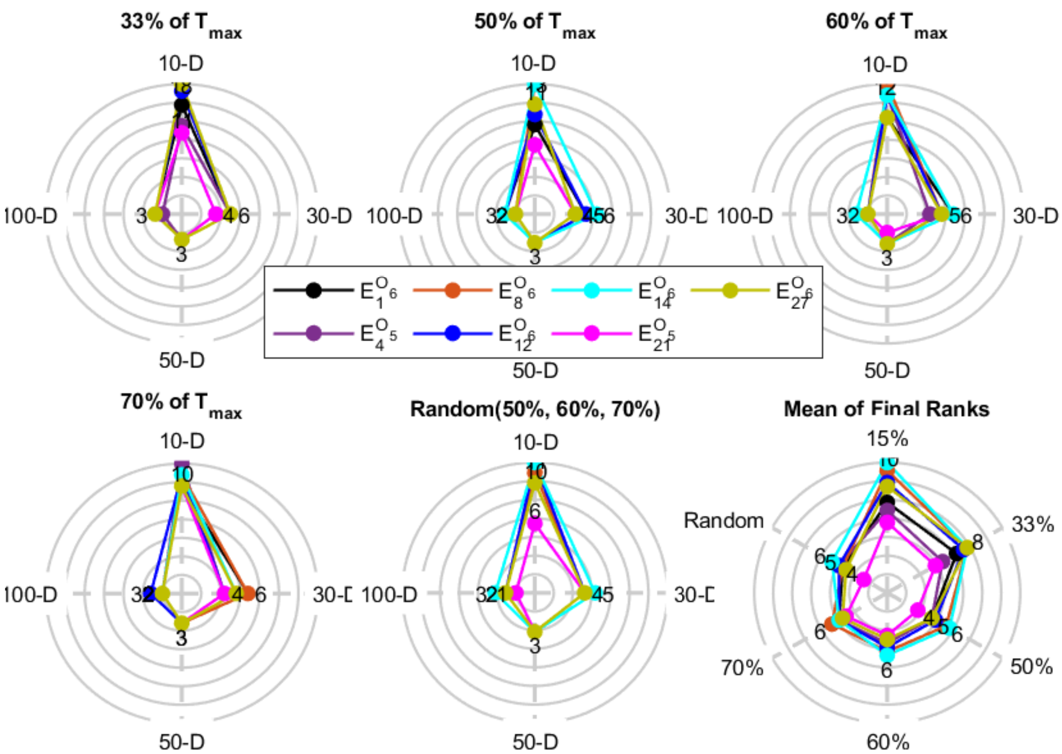

Figure 3: Final ranks attained by $E_1^{O_6}$, $E_4^{O_5}$, $E_8^{O_6}$, $E_{12}^{O_6}$, $E_{14}^{O_6}$, $E_{21}^{O_5}$ and $E_{27}^{O_6}$ with varying numbers of iterations allocated to AHPSO for CEC'17 problems.

is verified by the rest of the results obtained (as shown in Figure 1) for 30 and 50-dimensional problems. For instance, $E_{18}^{O_1}$, $E_{19}^{O_1}$, $E_{20}^{O_1}$, $E_{21}^{O_2}$, $E_{22}^{O_2}$, $E_{23}^{O_2}$, $E_{24}^{O_1}$, $E_{25}^{O_1}$, $E_{26}^{O_1}$ and $E_{27}^{O_1}$ were statistically significantly better than 60% to 80% of the 39 benchmark algorithms on 10-dimensional problems. However, on the 30 and 50-dimensional cases, these ensembles performed extremely inefficiently and were statistically significantly better than only 19%–22% of the benchmark algorithms at $\alpha = 0.05$, exhibiting a large decrease in performance with increased problem dimension. However, in contrast, the ensembles that cope with complex cases still yield good results for less complex (e.g. 10-dimensional) problems. Hence, performance on the 30 and 50-dimensional problems was weighted more heavily in assessing generalisation capabilities in choosing the most promising ensembles to further validate on the 100-dimensional CEC'17 problems.

Moreover, with regard to the ensemble ordering shown in Table 2, the best results for 30 and 50-dimensional CEC'17 problems were obtained with the orderings O_5 and O_6 (predominantly O_6), which initiate the search process with AHPSO. This reveals a particularly important point and verifies our initial hypothesis on utilising AHPSO as an explorative mechanism in our ensemble model to allow the discovery of promising regions in the search space (especially in high-dimensional cases). We theorise that the next algorithm inherits the discovered solutions and the relatively diverse population to further explore, exploit and converge to an optimal or near-optimal solution. The importance of AHPSO's role is also verified by the deteriorating performance of the same combination when executed in a different order. This performance deterioration is particularly apparent on high-dimensional problems when AHPSO is the last algorithm in the specified order. Hence, it is evident that initiating the search with the AHPSO algorithm greatly contributes to the overall performance through initial exploration and discovery of promising regions.

As shown in Figure 1, several combinations, namely, $E_1^{O_6}$, $E_4^{O_4}$, $E_8^{O_6}$, $E_{12}^{O_6}$, $E_{14}^{O_6}$, $E_{21}^{O_5}$ and $E_{27}^{O_6}$, perform consistently well while improving performance with increased problem complexity. Hence, these 7 combinations were selected for further testing on the 100-dimensional CEC'17 problems. Moreover, the default version of the ensemble algorithm distributes the maximum number of iterations T_{max} (defined within the test suites) equally between the three constituent algorithms. Considering the impact of AHPSO as the initial algorithm, additional experiments were conducted to assess the response of the 7 combinations to an asymmetric distribution of the number of iterations, allowing AHPSO to dominate the search process. The number of iterations assigned to AHPSO was experimented with as 15%, 33%(default), 50%, 60%, 70%, and randomly assigned (as either 50%, 60% or 70%) of the maximum number of iterations, with the remaining iterations divided equally between the other two algorithms in the ensemble.

Figure 3 displays the final ranks and the mean final ranks attained for each iteration distribution case for the 7 ensembles highlighted in the previous paragraph. In addition, Figure 2 summarises the percentages of statistically significant results for each combination for 10, 30, 50, and 100-dimensional CEC'17 problems with AHPSO assigned the number of iterations randomly (50%, 60%, or 70% of T_{max}).

It is evident from Figure 3 that $E_{21}^{O_5}$ consistently maintained lower final ranks across 10, 30, 50 and 100-dimensional CEC'17 problems compared to the rest of the ensembles. Moreover, $E_{21}^{O_5}$ also appears to be the most responsive to the asymmetrical distribution of the maximum number of iterations for AHPSO. $E_{21}^{O_5}$'s best-in-class mean final rank of 3.5 was attained when the number of iterations for AHPSO was allocated randomly as either 50%, 60% or 70% of T_{max}. With this setting, $E_{21}^{O_5}$ ranked 6^{th}, 4^{th}, 3^{rd} and 1^{st} on the 10, 30, 50 and 100-dimensional CEC'17 problems, respectively, against a comprehensive list of 39 benchmark algorithms.

Overall, based on the results obtained on the CEC'17 problems, we concluded that $E_{21}^{O_5}$ with the number of iteration for AHPSO randomly allocated (as either 50%, 60%, or 70% of T_{max}) is the best ensemble out of the 7 that were shortlisted and tested from the initial pool of 168 potential ensembles. Hence, our investigation proceeded with further validation and analysis of the $E_{21}^{O_5}$ ensemble on different test suites. A high-level flow chart of the proposed algorithm $E_{21}^{O_5}$ – **which from this point on we shall refer to as E-AHPSO** – is shown in Figure 4.

The detailed results of the preliminary experiments are presented in the supplementary material for interested readers.

4.5. Sequential and Parallel Ensemble Approach

In the literature, two main typical approaches are used for ensemble algorithms' search process: *sequential* and *parallel*. In the sequential approach, an algorithm initiates the search using the whole population; after the given iterations are completed, the population is passed on to the subsequent algorithm to resume the search. In the parallel approach, each algorithm in the ensemble model evolves a (randomly or equally assigned) proportion of the

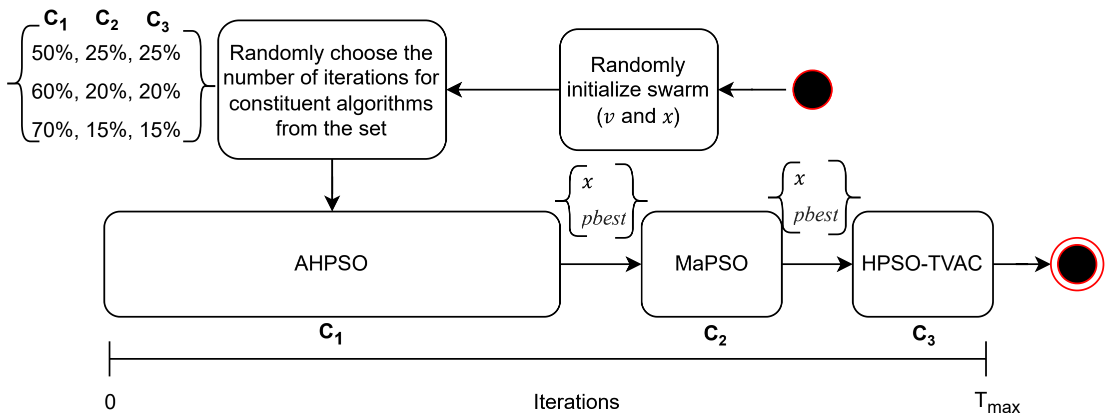

Figure 4: Flow chart of the proposed E-AHPSO algorithm ($E_{21}^{O_5}$).

population in parallel, with communication via mechanisms such as a shared *gbest* and periodic migration.

Our work, which uses AHPSO as the central mechanism to build our ensemble algorithm, aligned better with the sequential approach. However, to counter any hesitation about this approach, we also replicated our experiments using the parallel approach. In the parallel approach, the population size was set to 40 (the same as in the sequential approach) for all constituent algorithms, and the number of iterations was distributed equally.

For the shared *gbest* approach, the constituent algorithms run in parallel with a shared *gbest* exemplar, and in the latter approach (periodic migration): *gbest* is migrated at specific intervals: 50, 250, 500, 750, and 1000. The experiments were conducted on all dimensions of the CEC'13, CEC'14 and CEC'17 test suites for both parallel approaches. Figure 5 visualise the statistical results for the 10, 30, 50 and 100-dimensional CEC'13 problems for the parallel E-AHPSO with shared *gbest*. The results reveal that the performance deteriorates significantly relative to the sequential approach across all test suites and dimensions, with many comparator algorithms outperforming the parallel version that uses a shared *gbest*. Similarly, Figures 6-7 display the results for the parallel E-AHPSO with periodic migration intervals of 50 and 1000 for the 10, 30, 50 and 100-dimensional CEC'13 problems. Once again, the parallel version of the E-AHPSO with periodic migration exhibits performance comparable to that of the shared *gbest* version, with some improvement observed in 100-dimensional cases. We also observe a similar performance trend across other dimensions and test suites. Although we observe slight differences among parallel E-AHPSO versions, overall performance is significantly inferior to that of the sequential E-AHPSO.

We believe that this is because the explicit initial explorative/population diversifying role of the AHPSO algorithm in the sequential version is not fully realised in the parallel version. Since the results obtained for the parallel approach do not change the narrative of our work, we have only provided them in summary form. Interested readers can find the raw results of the parallel approach ensemble model in the supplementary material.

5. Main Experiments: Detailed Comparative Validation of E-AHPSO

In this section, the performance of the ensemble that demonstrated the strongest overall performance in the preliminary experiments, hereafter referred to as E-AHPSO (Section 4.4), is evaluated on a broader range of problems using the CEC'13, CEC'14, and CEC'20 test suites.

5.1. Experimental Setup

The effectiveness of the proposed algorithm, E-AHPSO, particularly its ability to perform well over a wide range of problems, was further evaluated on the 10, 30, 50 and 100-dimensional CEC'13[102], CEC'14[103] and CEC'20[104] (*10 and 20-dimensional*) test suites against the same very competitive 39 benchmark algorithms used in the preliminary experiments (section 4.4), consisting of 15 mature PSO variants (including the Bare-bones PSO, Comprehensive Learning PSO, Dynamic Multi-Swarm PSO, Ensemble PSO, Fitness-Distance Ratio PSO, Fully Informed PSO, Heterogeneous Comprehensive Learning PSO, Hierarchical PSO with Time-Varying Coefficients, (*see Table 3*) that have been extensively investigated, cited and applied to many real-world problems, 14 recently published (2020 and after) powerful and highly cited PSO variants (including advanced heterogenous multi-swarm and multi-strategy learning variants such as HCLDMS-PSO, HIDMS-PSO, FFQ-HIDMS-PSO, A-HIDMSPSO, GA-HIDMS-PSO, AHPSO, BEPSO, TL-SPSO, XPSO) and 10 non-PSO algorithms including three differential evolution variants (IEEE CEC competition winner L-SHADE, CoDE and MTDE).

The full details of the benchmark algorithms and the corresponding parametric settings are shown in Table 3. The best published general settings were used for each algorithm. Each problem was tested 51 times to compute

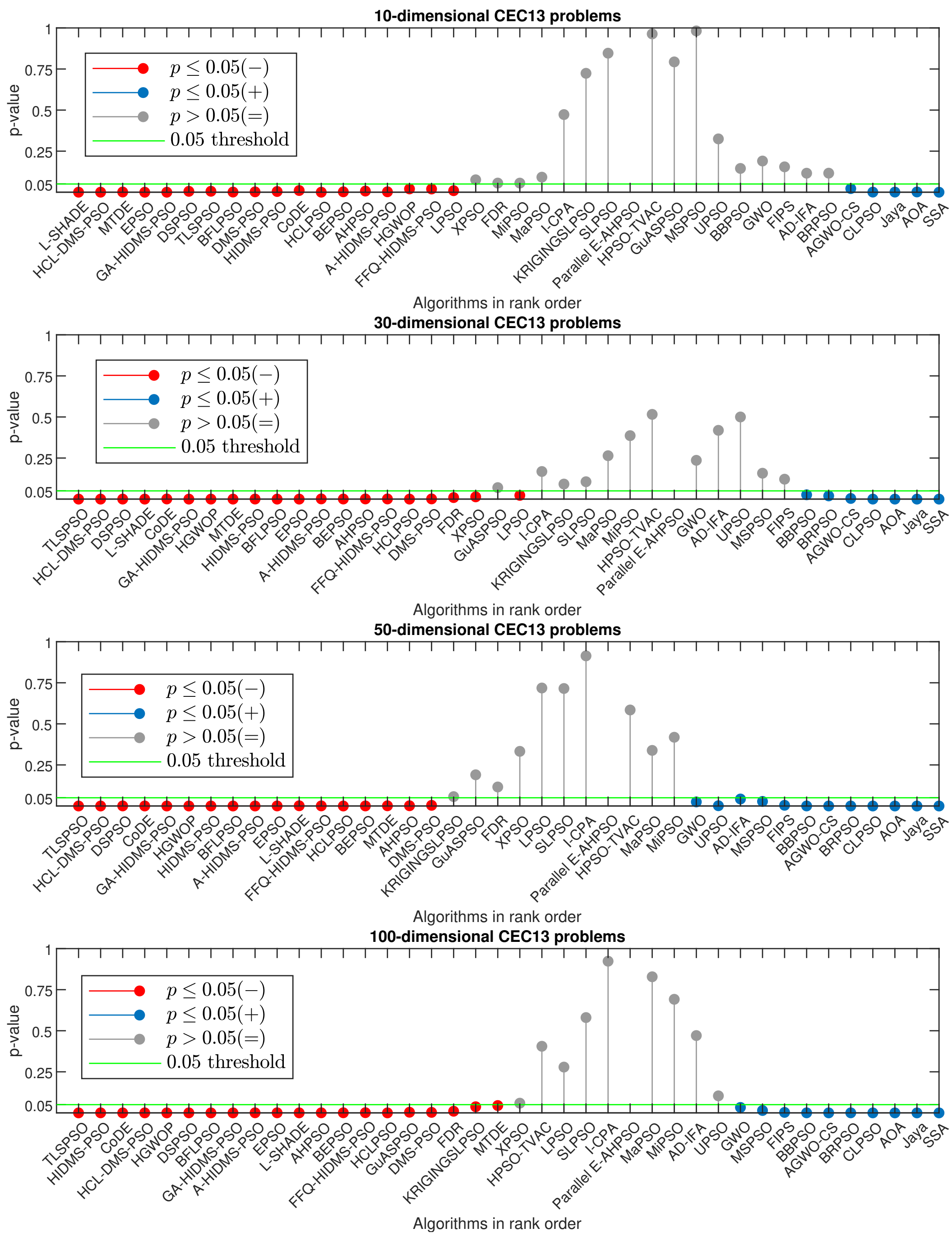

Figure 5: Statistical results for parallel E-AHPSO with shared *gbest* against the 39 benchmark algorithms on 10, 30, 50, and 100-dimensional CEC'13 problems. The symbols $+$, $=$, and $-$ indicate significantly better performance, no statistically significant difference, and significantly worse performance, respectively, at $\alpha = 0.05$.

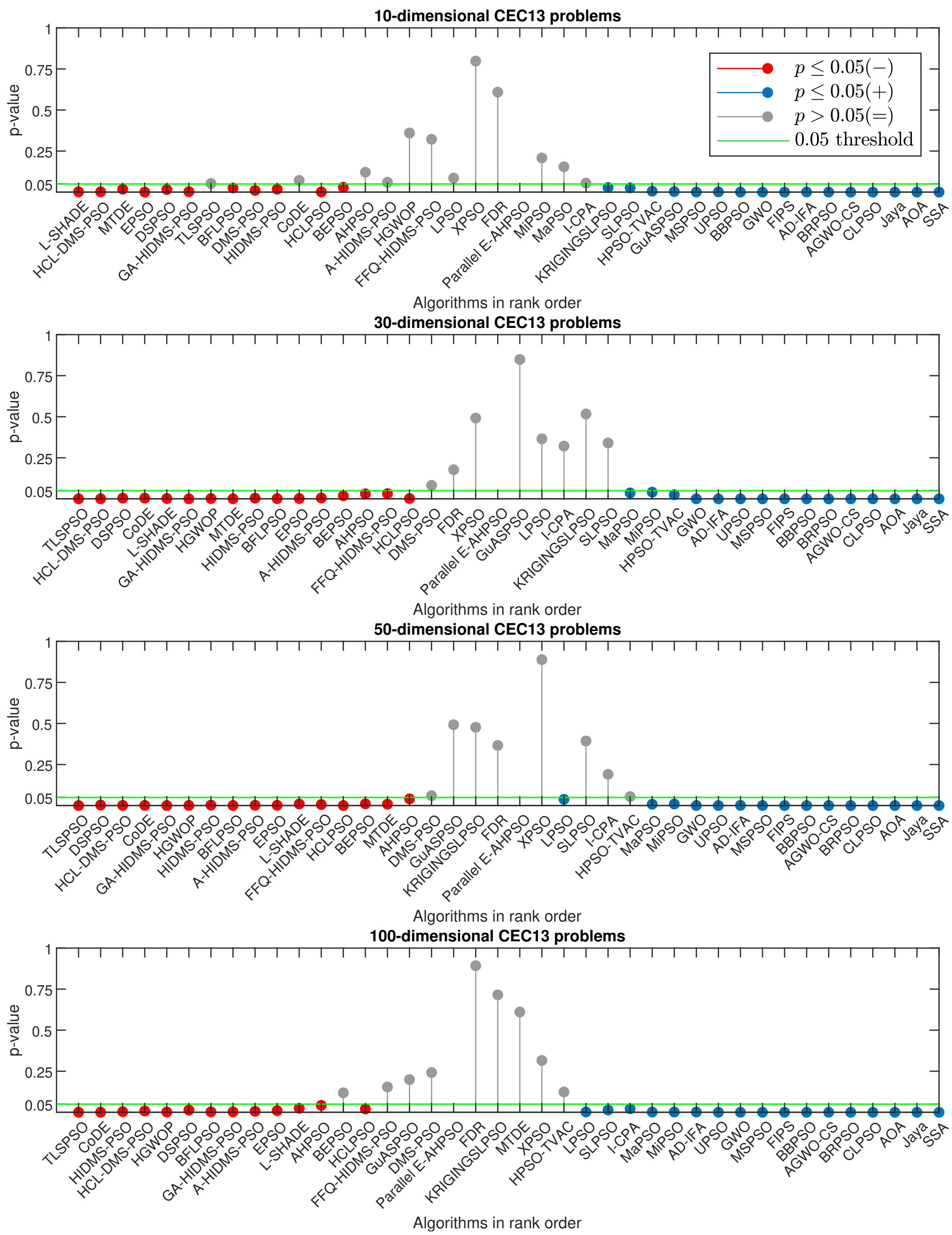

Figure 6: Statistical results for parallel E-AHPSO with periodically shared *gbest* (migration interval=50) against the 39 benchmark algorithms on 10, 30, 50, and 100-dimensional CEC'13 problems. The symbols +, =, and − indicate significantly better performance, no statistically significant difference, and significantly worse performance, respectively, at $\alpha = 0.05$.

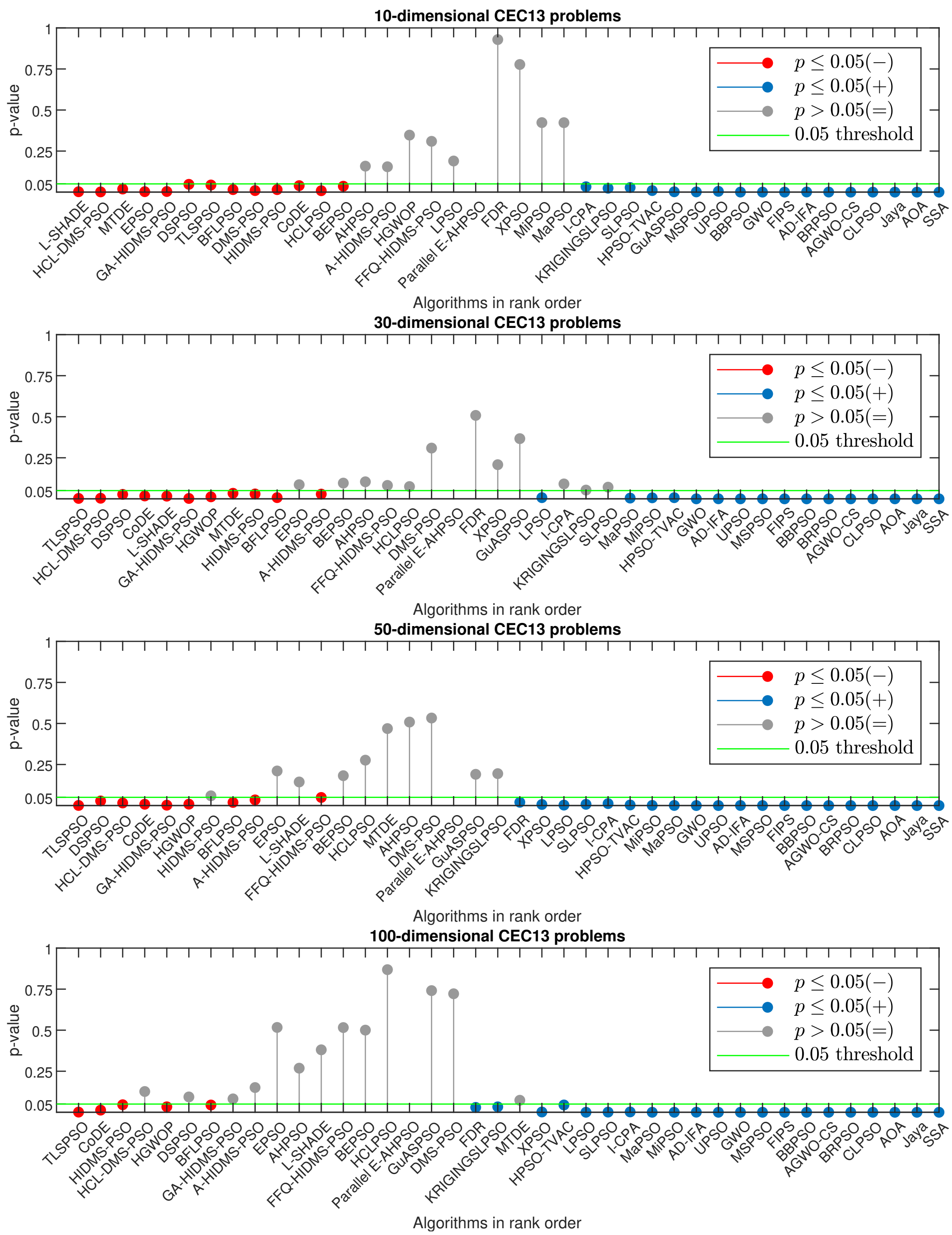

Figure 7: Statistical results for parallel E-AHPSO with periodically shared *gbest* (migration interval=1000) against the 39 benchmark algorithms on 10, 30, 50, and 100-dimensional CEC'13 problems. The symbols +, =, and − indicate significantly better performance, no statistically significant difference, and significantly worse performance, respectively, at $\alpha = 0.05$.

Table 3: Benchmark Algorithms and Parameters.

Key	Parameters
E-AHPSO*	*see corresponding settings of the constituent algorithms*
AD-IFA (2020) [105]	$n = 40, \alpha = 0.2, \beta_{min} = 1, \gamma = 1$
AGWOCS (2021) [106]	$n = 100, \alpha = 2 - 2(\frac{t}{t_{max}})$
A-HIDMS-PSO (2021) [107]	$n = 40, \omega_1 = \frac{0.99+(0.2-0.99)}{1+exp(-5(\frac{2t}{T_{max}}-1))}$ $c_1 = 2.5 \rightarrow 0.5, c_2 = 0.5 \rightarrow 2.5$
AHPSO (2021) [2]	$n = 40, \omega_1 = \frac{0.99+(0.2-0.99)}{1+exp(-5(\frac{2t}{T_{max}}-1))}, c_1 = 2.5 \rightarrow 0.5$ $c_2 = 0.5 \rightarrow 2.5, \alpha = 20, E_r = 1$
AOA (2020) [108]	$n = 30, \alpha = 5, \mu = 0.499$
BBPSO (2003) [18]	$n = 40, \phi = 4.1, \lambda = 0.729, c_1 = c_2 = 2.05$ $r_1 = r_2 = U(0, 1)$
BEPSO (2021) [109]	$n = 40, \omega_1 = \frac{0.99+(0.2-0.99)}{1+exp(-5(\frac{2t}{T_{max}}-1))}, c_1 = 2.5 \rightarrow 0.5$ $c_2 = 0.5 \rightarrow 2.5, phase = 10, SR_{max} = 0.01$ $SR_{min} = 0.001, bias_{threshold} = 20$
BFLPSO (2021) [110]	$n = 40, \omega = 0.9 \rightarrow 0.2, c = 1.49445$ $I = E = 1, G = 5$
BRPSO (2005) [111]	$n = 40, \omega = 0.8 \rightarrow 0.6, c_1, c_2 = 1.49445$
CLPSO (2006) [56]	$n = 40, \omega = 0.4 \rightarrow 0.9, c_1 = c_2 = 1.49445, m = 7$
CoDE (2011) [25]	$CR = [0.1, 0.9, 0.2], F = [1, 1, 0.8]$
DMS-PSO (2006) [52]	$n = 40, \omega == 0.729, c_1, c_2 = 1.49445, M = 3, R = 15$
DSPSO (2019) [112]	$n = 40, c_1 = 2, F_{min} = 0.7$
EPSO (2017) [94]	$n = 40, \omega = 0.9 \rightarrow 0.2, \omega_1 = 0.9 \rightarrow 0.4, c_1 = 3 \rightarrow 1.5$ $c_{2_1}, c_{3_1}, c_{4_1} = 2.5 \rightarrow 0.5, c_{2_2}, c_{3_2}$ $c_{4_2} = 0.5 \rightarrow 2.5, Pc = 0.5, n_{size} = 3$
FDR (2003) [98]	$n = 40, \omega = 0.9 \rightarrow 0.4, c_1 = c_2, c = 2$
FIPS (2004) [21]	$n = 40, \chi = 0.729, \sum c_i = 4.1$
FFQ-HIDMS-PSO (2021) [113]	$n = 40, \omega_1 = \frac{0.99+(0.2-0.99)}{1+exp(-5(\frac{2t}{T_{max}}-1))}, c_1 = 2.5 \rightarrow 0.5$ $c_2 = 0.5 \rightarrow 2.5, FF_{period} = 100$
GA-HIDMS-PSO (2021) [114]	$n = 40, \omega_1 = \frac{0.99+(0.2-0.99)}{1+exp(-5(\frac{2t}{T_{max}}-1))}, c_1 = 2.5 \rightarrow 0.5$ $c_2 = 0.5 \rightarrow 2.5, phase_{PSO} = 100, phase_{GA} = 50, Pc = 0.7$ $\gamma = 0.4, Pm = 0.3, \mu = 0.1, Tournament_{size} = 3$
GuASPSO (2020) [115]	$n = 50, n_{cluster} = 5, cluster_{max} = n$ $cluster_{max} = 2, LR_{max} = 0.1, w_{mode} = 2, k_{min} = 0.4$ $k_{max} = 0.9, SOM^{iter} = 4 * n$
GWO (2014) [116]	$n = 100, \alpha = 2 - 2(\frac{t}{t_{max}})$
HCL-DMS-PSO (2020) [117]	$n = 40, \omega = 0.29 \rightarrow 0.99$ $\omega_2 = c_1 = c_2 = 0.5 \rightarrow 2.5, p_m = 0.1$
HCLPSO (2015) [57]	$n = 40, \omega = [0.2, 0.9], c_1, c_2 = [0.5, 2.5], c = [1.5, 3]$
HGWOP (2021) [118]	$n = 100, \alpha_{max} = 2, \alpha_{min} = 0, Cr_{max} = 1, Cr_{min} = 0$
HIDMS-PSO (2020) [119]	$n = 40, \omega_1 = \frac{0.99+(0.2-0.99)}{1+exp(-5(\frac{2t}{T_{max}}-1))} c_1 = 2.5 \rightarrow 0.5$ $c_2 = 0.5 \rightarrow 2.5$
HPSO-TVAC (2004) [20]	$n = 40, c_1 = 2.5 \rightarrow 0.5, c_2 = 0.5 \rightarrow 2.5$
GrSAPSO (2020) [120]	$n = 40, \alpha = 0.5, \beta = 0.01, c_3 = d/n * 0.01$
I-CPA (2023) [121]	$nCPlant = 2, nPrey = 8, n = nCPlant + nPrey$
JAYA (2016) [122]	$n = 50$
LPSO (1998) [19]	$n = 40, \omega = 0.9 \rightarrow 0.4, c_1 = c_2 = 2$
L-SHADE (2014) [24]	$N^{init} = round(D \times r^{N^{init}}), \lvert A \rvert = round(N^{init} \times r^{arc})$ $r^{arch} = 2.6, p = 0.11, H = 6$
maPSO (2014) [23]	$n = 40, \omega = 0.9 \rightarrow 0.4, c = 1.49445$
miPSO (2014) [23]	$n = 40, \omega = 0.9 \rightarrow 0.4, c = 1.49445$
MSCPSO (2020) [123]	$n = 40, c_1, c_2 = 1.4, k_1 = 5, k_2 = 10, M = 5$
MSPSO (2023) [124]	$n = 100, \omega = 0.729, c_1, c_2 = 2$
MTDE (2020) [26]	$WinIter = 20, H = 5, initial = 0.001, final = 2$ $\mu = log(D), \mu f = 0.5, \sigma = 0.2$
SLPSO (2015) [125]	$n = 100, \alpha = 0.5, \beta = 0.01$
SSA (2019) [126]	$n = 50, N_{fs} = 4, G_c = 1.9, P_{dp} = 0.1$
TLSPSO (2022) [70]	$n = 100, c_2 = 2$
UPSO (2004) [22]	$n = 30, \chi = 0.729, c_1, c_2 = 2.05, u = 0.5$
XPSO (2020) [127]	$n = 50, \omega = 0.4 \rightarrow 0.9, \eta = 0.2, Stag_{max} = 5, p = 0.2$

the mean, median, standard deviation, minimum, and maximum error values. The population was initialised using uniform random initialisation within the search space. Randomness was handled using MATLAB's `rng('shuffle')` command as a time-dependent seed. Therefore, the 51 runs were performed as independent stochastic trials with different random states, rather than using a predefined fixed seed sequence. The rest of the experimental settings are replicated as stated in the original publications of the test suites [102][103][128][104].

5.2. Experimental Results

In this section, the experimental results for the CEC'13, CEC'14 and CEC'20 problems are discussed in detail, while the results obtained in the preliminary experiment for the CEC'17 problems are overviewed in conjunction with the main experimental results to observe any patterns or issues regarding the generalisation capability of the proposed algorithm, E-AHPSO.

Figures 8-11 display the results obtained for the CEC'13, CEC'14 and CEC'20 test suites. Based on the mean and final ranks obtained by E-AHPSO, it is evident that for both CEC'14 and CEC'17 problems, while the proposed algorithm performs very well at all dimensions, it generally improves with increased problem dimensions; E-AHPSO ranked 1^{st} out of the 39 benchmark algorithms for 100-dimensional CEC'14 and CEC'17 problems, and attained the 6^{th}, 5^{th} and 4^{th} ranks on 10, 30, and 50-dimensional CEC'14 problems, respectively. We observe a similar pattern in the case of the CEC'17 problems, shown in section 4.3, where E-AHPSO ranked 5^{th}, 4^{th}, 3^{rd} and 1^{st} for 10, 30, 50 and 100-dimensional problems, respectively.

However, we observe an opposite trend in the case of the CEC'13 problems, which appears to be a common pattern in the performance of the majority of the benchmark algorithms; E-AHPSO ranked 3^{rd}, 4^{th}, 2^{nd} and 9^{th} on 10, 30, 50 and 100-dimensional CEC'13 problems, respectively.

Moreover, E-AHPSO ranked 7^{th} and 4^{th} on 10 and 20-dimensional CEC'20 problems out of 39 benchmark algorithms, respectively. It is worth noting that the CEC'20 test suite is much less complex (with regards to the number of problems and dimensions) compared to the CEC'13, CEC'14 and CEC'17 test suites and is comprised of 10 problems testable solely at 5, 10, 15 and 20 dimensions, making it a more competitive test suite. The final ranks obtained by E-AHPSO on CEC'20 problems further verify the consistent performance and generalisation capability of the proposed algorithm.

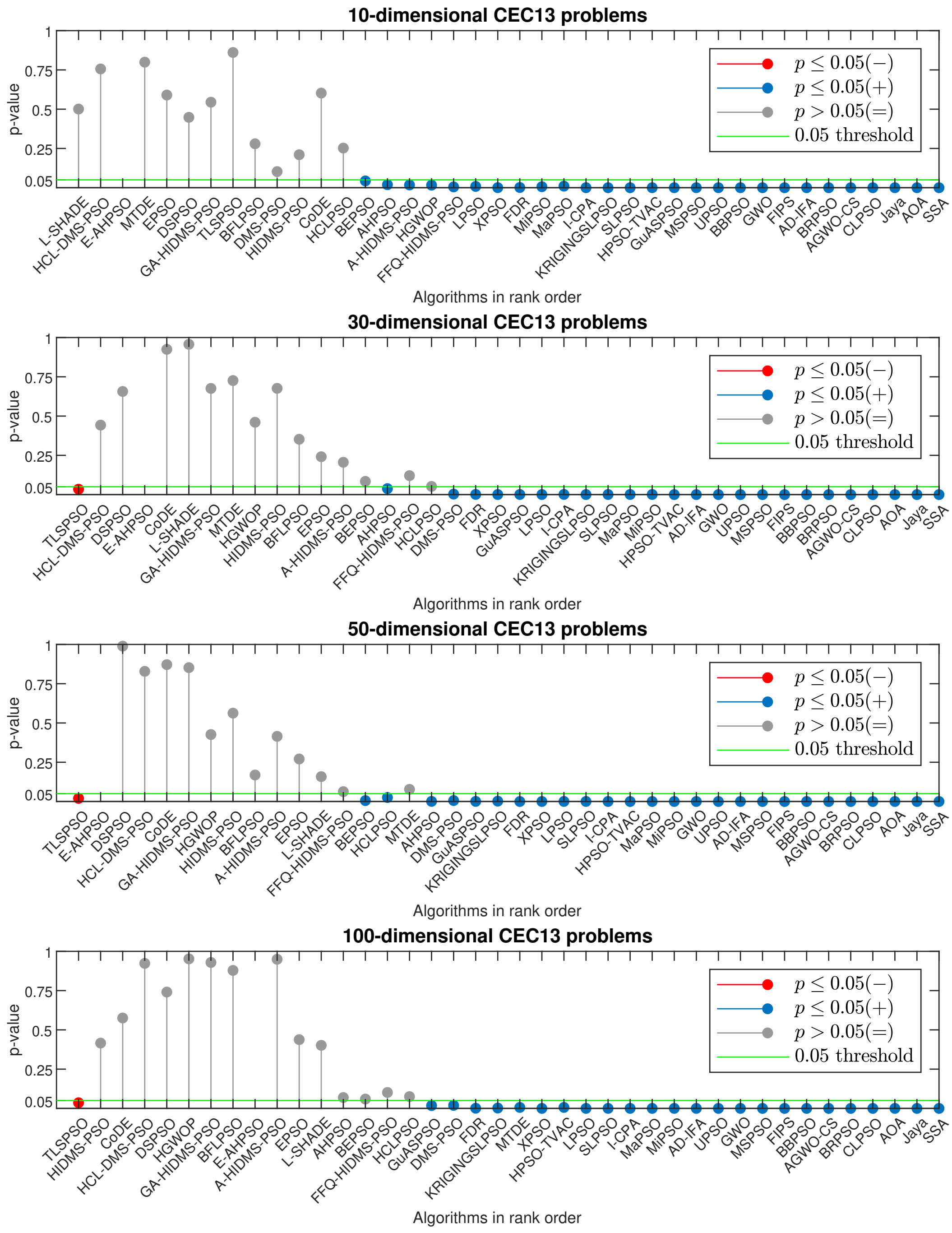

Figure 8: Statistical results for E-AHPSO against the 39 benchmark algorithms on 10, 30, 50 and 100-dimensional CEC'13 problems. The symbols $+$, $=$, and $-$ indicate significantly better performance, no statistically significant difference, and significantly worse performance, respectively, at $\alpha = 0.05$.

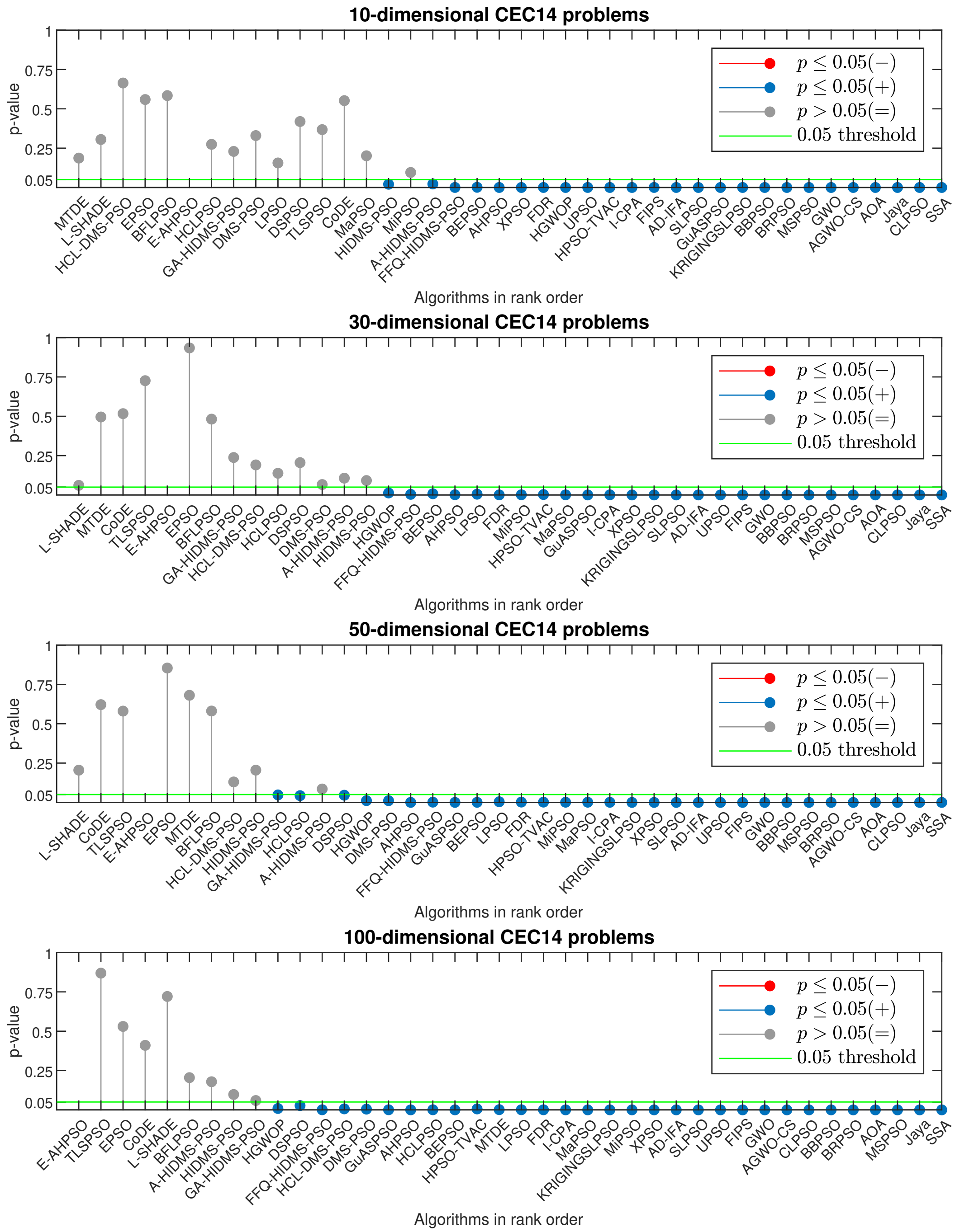

Figure 9: Statistical results for E-AHPSO against the 39 benchmark algorithms on 10, 30, 50 and 100-dimensional CEC'14 problems. The symbols +, =, and − indicate significantly better performance, no statistically significant difference, and significantly worse performance, respectively, at $\alpha = 0.05$.

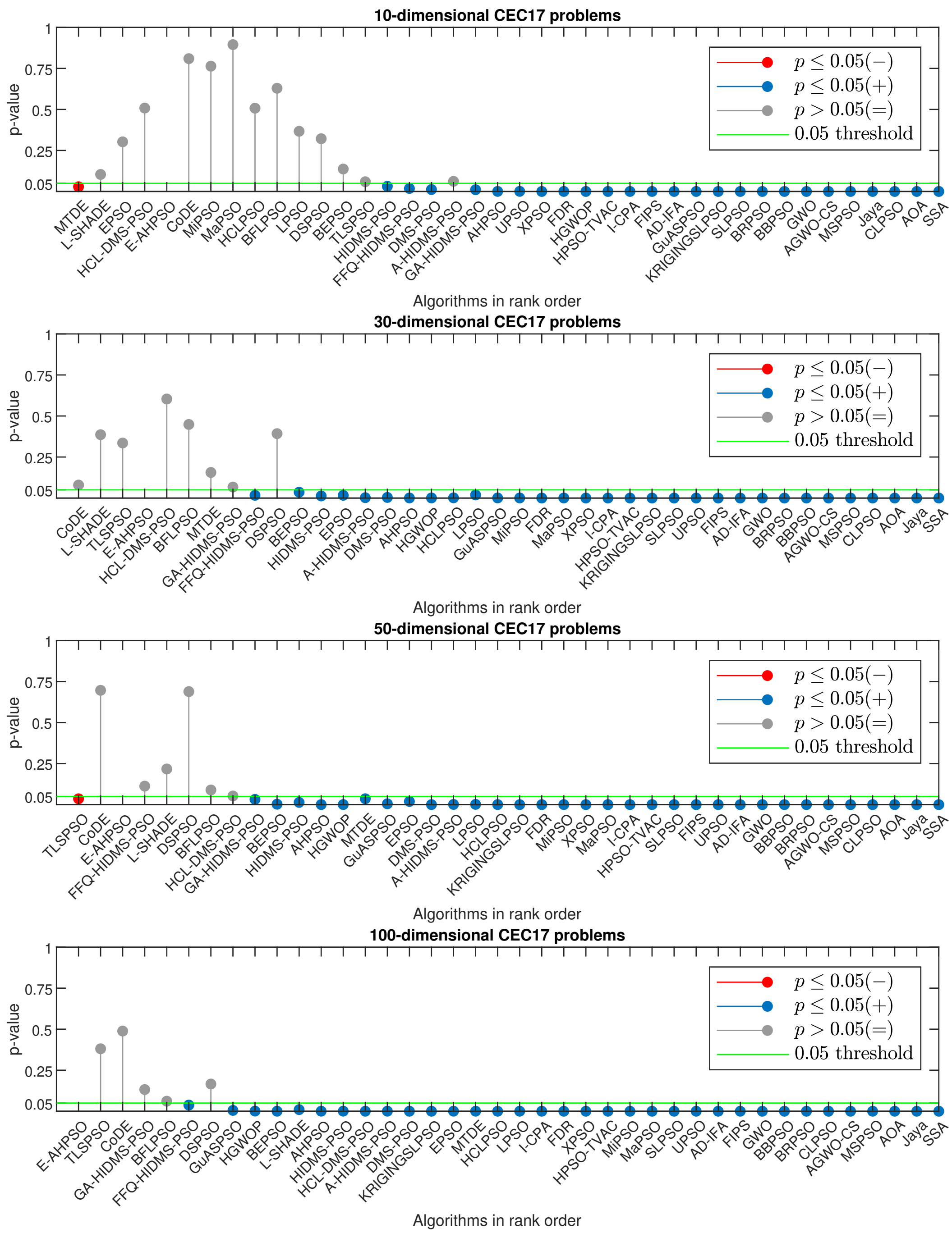

Figure 10: Statistical results for E-AHPSO against the 39 benchmark algorithms on 10, 30, 50, 100-dimensional CEC'17 problems. The symbols +, =, and − indicate significantly better performance, no statistically significant difference, and significantly worse performance, respectively, at $\alpha = 0.05$.

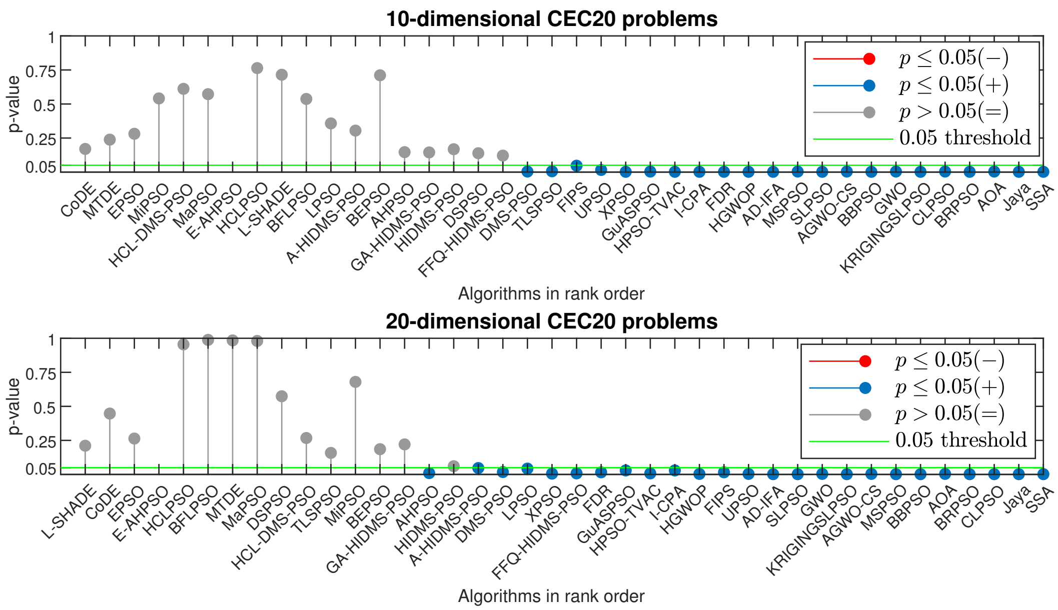

Figure 11: Statistical results for E-AHPSO against the 39 benchmark algorithms on 10 and 20-dimensional CEC'20 problems. The symbols $+$, $=$, and $-$ indicate significantly better performance, no statistically significant difference, and significantly worse performance, respectively, at $\alpha = 0.05$.

The Wilcoxon signed-rank test was applied to the results obtained for the 10-, 30-, 50-, and 100-dimensional CEC'13, CEC'14, and CEC'20 test suites using a significance level of $\alpha = 0.05$. Throughout the statistical analyses, the symbols $+$, $=$, and $-$ indicate that the reference algorithm performs statistically significantly better than, is not statistically significantly different from, or performs statistically significantly worse than the comparator algorithm, respectively. The symbols $+$ and $-$ therefore correspond to $p \leq \alpha$, with the direction determined by the comparative performance of the algorithms, whereas $=$ denotes $p > \alpha$.

Due to the large volume of data, the results presented in the paper are summarised; however, the detailed statistical results are provided in the supplementary material for interested readers.

Figures 8-11 present a summary of the comparative statistical significance results against the 39 benchmark algorithms at a significance level of 0.05 for the CEC'13, CEC'14 and CEC'20 problems. We note that *none* of the 39 benchmark algorithms were statistically significantly better than E-AHPSO for *any* of the dimensions of the CEC'14 and CEC'20 problems. In addition, the results obtained for the CEC'13 problems reveal that TLSPSO is the sole algorithm to be statistically significantly better than E-AHPSO on 30, 50, and 100-dimensional CEC'13 problems. However, as shown in Figure 11, E-AHPSO was statistically significantly better than TLSPSO on 10-dimensional CEC'20 problems.

A closer inspection of the statistical results compared with various powerful well-known benchmark algorithms, including CoDE, L-SHADE, MTDE, HCL-DMS-PSO and EPSO (a prominent PSO-based ensemble), indicates that E-AHPSO was statistically significantly better than EPSO on 30-dimensional, EPSO and MTDE on 50-dimensional, EPSO, MTDE and L-SHADE on 100-dimensional CEC'17 problems. Moreover, E-AHPSO was significantly better than HCL-DMS-PSO and MTDE on the 100-dimensional CEC'14 problems. Figure 12 displays the final ranks attained by the same top-performing algorithms (CoDE, L-SHADE, MTDE, EPSO, HCL-DMS-PSO, TLSPSO) on the CEC'13/14/17 problems and the average of final ranks obtained across different dimensions of the three test suites. The Figure shows that the proposed algorithm, E-AHPSO exhibits noticeable robustness and generalisation capabilities by maintaining comparable final ranks across all four dimensions of the CEC'13, CEC'14, and CEC'17 test suites, while attaining better ranks compared to various top-performing benchmark algorithms. In addition, assessment of the statistical results for the constituent algorithms that comprise

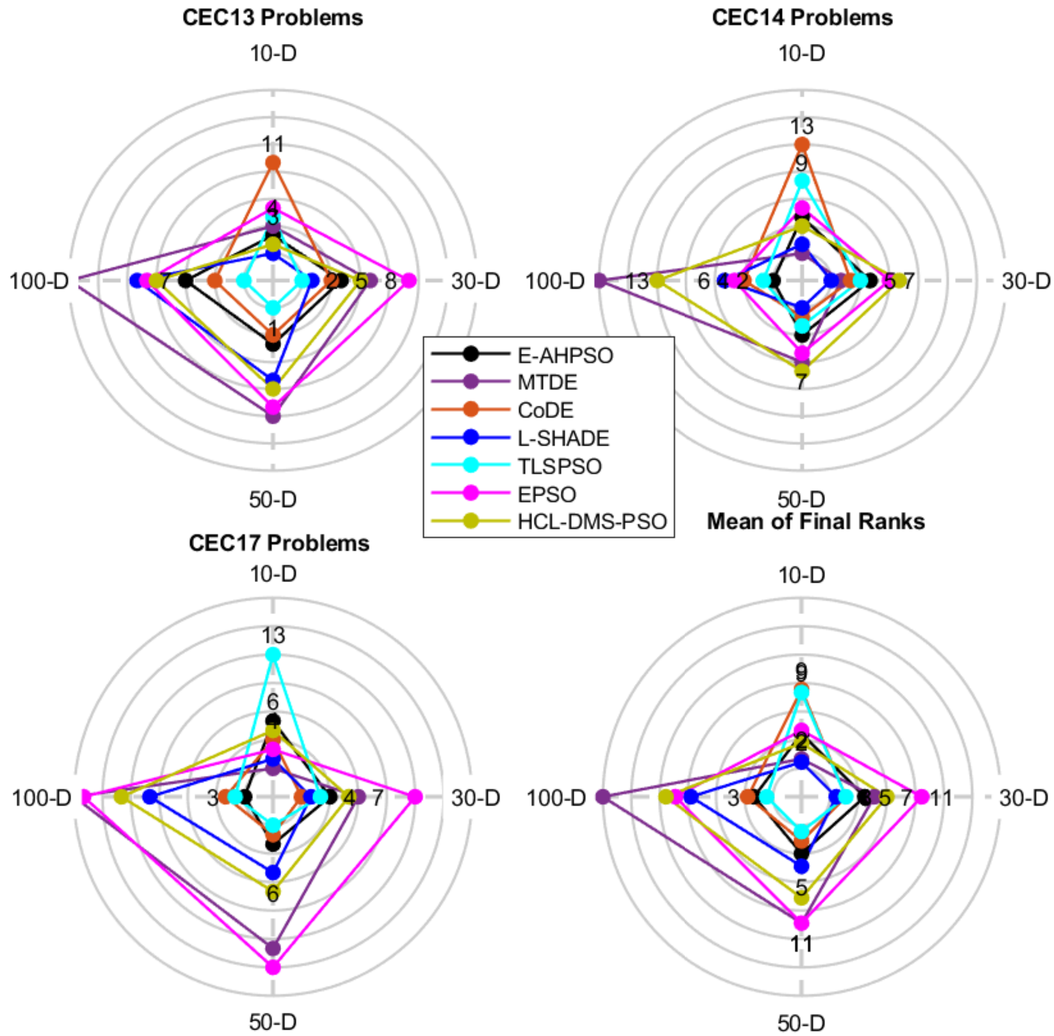

Figure 12: Comparison of the final ranks obtained on CEC'13, CEC'14 and CEC'17 test suites for 10, 30, 50 and 100-dimensional cases by the proposed algorithm (E-AHPSO) and several top-performing algorithms.

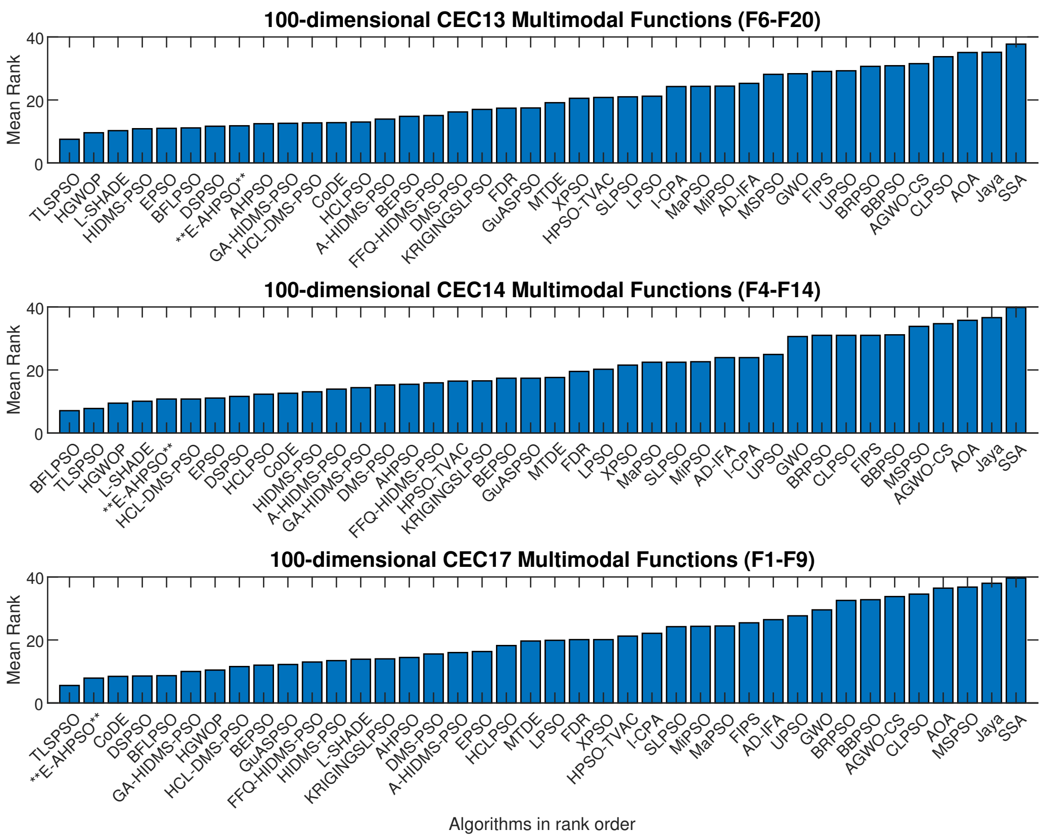

Figure 13: Mean ranks obtained by the proposed algorithm (E-AHPSO) and the 39 benchmark algorithms for 100-dimensional multimodal problems from the CEC'13, CEC'14 and CEC'17 test suite.

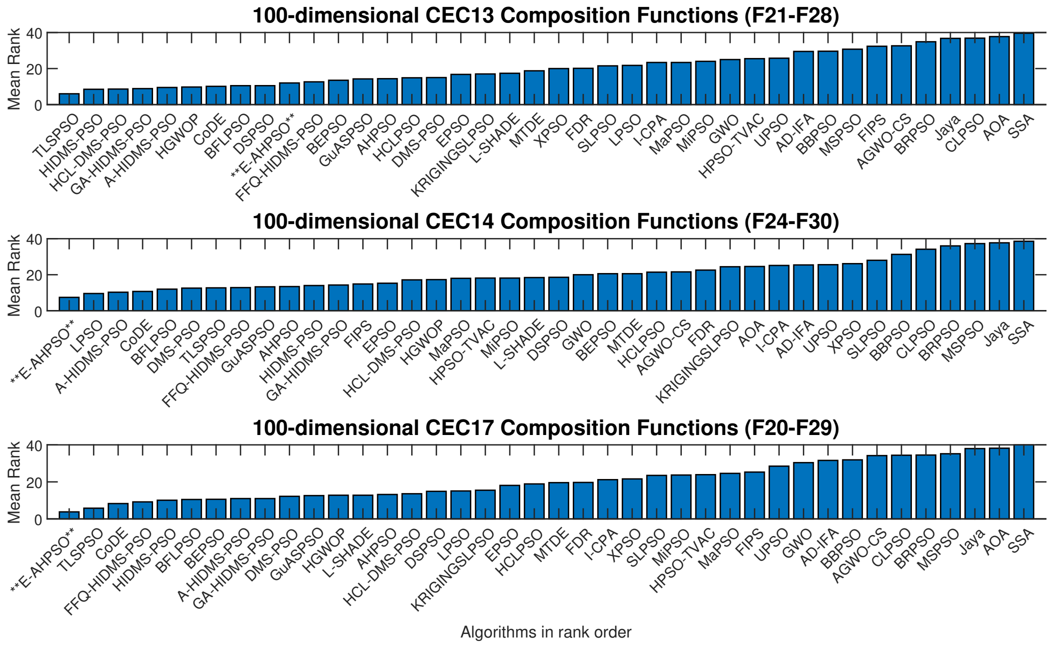

Figure 14: Mean ranks obtained by the proposed algorithm (E-AHPSO) and the 39 benchmark algorithms for 100-dimensional composition problems from the CEC'13, CEC'14 and CEC'17 test suite.

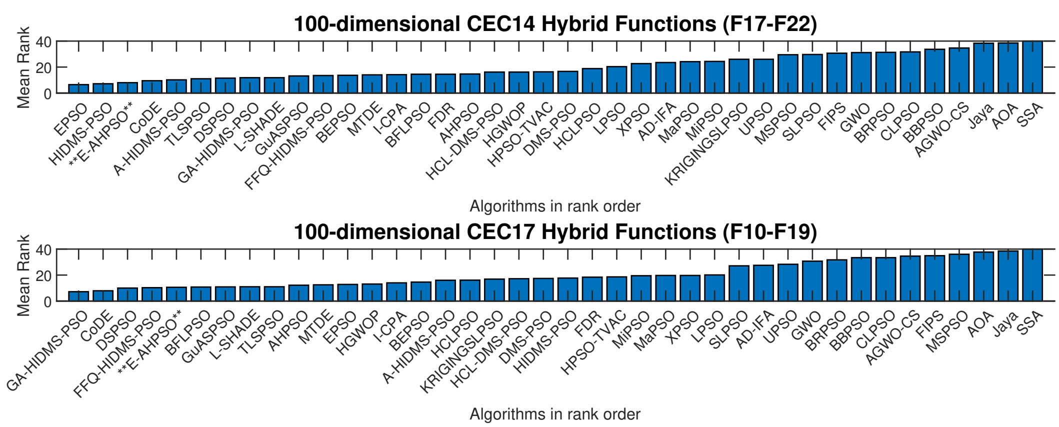

Figure 15: Mean ranks obtained by the proposed algorithm (E-AHPSO) and the 39 benchmark algorithms for 100-dimensional hybrid problems from the CEC'14 and CEC'17 test suite.

E-AHPSO reveals that the proposed ensemble algorithm is statistically significantly better than its constituents, AHPSO, MaPSO and HPSO-TVAC, in all of the experiments except for two cases where no significant difference was observed for AHPSO (on 100-d CEC'13 problems) and MaPSO (on 10-d CEC'14 problems).

Overall, on average, the proposed algorithm E-AHPSO was statistically significantly better than 64% (69%, 61%, 66%, 61% at dimensions 10, 30, 50, 100) on CEC'13, 71% (64%, 66%, 76%, 79% at dimensions 10, 30, 50, 100) on CEC'14 and 78% (64%, 82%, 82%, 87% at dimensions 10, 30, 50, 100) on CEC'17 and 60% (56%, 64% at dimensions 10 and 20) of the 39 benchmark algorithms on the CEC'20 test suite.

Furthermore, E-AHPSO's performance on various problem types was assessed to gain a deeper understanding of its limitations and suitability for specific problem types. Figures 13,14,15 display the average of the ranks obtained on the 100-dimensional CEC'13, CEC'14, CEC'17 multimodal, hybrid and composition functions. In summary, E-AHPSO obtained the lowest mean ranks on the composition functions of the CEC'14 and CEC'17 test suites (7.4 and 3.8). For multimodal CEC'14 and CEC'17 problems, E-AHPSO obtained the 5^{th} and 2^{nd} lowest average ranks, respectively. The CEC'14 and CEC'17 test suites contain hybrid problems that are randomly divided into groups and evaluated by distinct functions. This type of partial separability is a common characteristic observed in real-world problems. Hence, the performance of an algorithm on hybrid functions may indicate its potential for real-world problems. Bearing this in mind, the proposed algorithm achieved the 3^{rd} lowest mean rank (8), behind HIDMS-PSO (7.1) and EPSO (6.5), for the hybrid problems of CEC'14, and the 5^{th} lowest mean rank for the hybrid problems of the CEC'17 test suite. In summary, E-AHPSO performed very competitively on all problem types.

At this point it is is worth revisiting the central hypothesis of combining exploitative algorithms with AHPSO to enhance the overall search process. It is important to note that the AHPSO algorithm's exploration capability is very general and robust, independent of the control parameters and rather embedded in the algorithm through bio-inspired heterogeneous behaviour, whereas the selected pool of candidate algorithms (FDR-PSO, UPSO, BBPSO, LPSO, HPSO-TVAC) primarily rely on control parameters to balance exploration and exploitation.

As MaPSO and HPSO-TVAC were selected as the exploitative algorithms in the ensemble based on the results of our preliminary experiments, it will be valuable to justify the behavioural differences that yield efficient performance with emphasis on the execution order. As we have previously discovered from our investigations, the optimal execution order begins with AHPSO, which exhibits highly exploratory behaviour, allowing the swarm to maintain greater population diversity and higher velocity magnitudes during the early stages of the search, allowing AHPSO to perform an initial broad global search, reducing the risk of premature convergence and aiding the swarm in identifying multiple regions of high potential before exploitation begins.

As shown in Figures 27-29, the subsequent constituent algorithm, HPSO-TVAC, inherits a highly active swarm to initiate the intermediate search phase. Due to HPSO-TVAC's reliance on the gbest, particles' focus intensifies towards the gbest exemplar as it reaches the end of the search phase. This enables the swarm to briefly explore the AHPSO-inherited solutions, then focus on the most promising regions before handing over to MaPSO. It is worth highlighting the importance of HPSO-TVAC's mediating role in E-AHPSO, which enables it to inherit AHPSO's output and, in effect, narrow it down to more selective and refined solutions for exploitation.

The late-stage exploitation performed by MaPSO is beyond a simple convergence and complements the previous two stages carried out by AHPSO and HPSO-TVAC. MaPSO's comprehensive learning mechanisms prevent the swarm from converging immediately at the onset of handover. This behaviour of MaPSO is particularly handy in multimodal and complex search spaces where solely relying on the gbest exemplar could lead the swarm to collapse on a local optimum. Hence, the constituent algorithms (AHPSO $\rightarrow$ HPSO-TVAC $\rightarrow$ MaPSO) and their particular ordering yield far better performance than the other ensemble configurations tested in preliminary experiments because each constituent contributes distinctly and sequentially complementary behaviours to the overall search process. These can be summarised as: AHPSO discovers promising regions while preserving diversity, HPSO-TVAC exploits and refines AHPSO inherited solutions, followed by MaPSO's late-stage exploitation towards convergence, while preventing over-exploitation and premature convergence via the comprehensive learning mechanism. The result is an efficient and robust optimiser which performs extremely well across different problem types and dimensions.

Table 4: Statistical performance results on CEC13 10D benchmark functions.

Function	Statistic	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
F1	avg	0	0	0	0	0	0	0	0	0
	std	0	0	0	0	0	0	0	0	0
	min	0	0	0	0	0	0	0	0	0
	max	0	0	0	0	0	0	0	0	0
F2	avg	4.77e4	4.72e5	0	2.54e4	1.16e5	1.94e4	0	0	1.15e4
	std	37087	2.96e5	0	1.57e4	1.17e5	1.44e4	0	0	6.79e3
	min	1.01e4	5.91e4	0	4.67e3	1.92e4	1.28e3	0	0	978.36
	max	1.87e5	1.12e6	0	6.25e4	5.19e5	4.97e4	0	0	2.40e4
F3	avg	424.27	417	0.69	5.19e5	451.56	265.69	0.47	1.4	0.67
	std	1.26e3	708.16	1.45	5.71e5	776.82	696.87	1.54	2.57	1.25
	min	0	0.2	0	1.23	5.05	0.03	0	0	0.01
	max	5.35e3	3.09e3	6.32	1.85e6	2.76e3	2.99e3	6.32	6.32	5.68
F4	avg	868.86	5.26e3	0	225.2	2.25e3	1.21e3	0.01	0	714
	std	610.23	1.90e3	0	136.66	1.31e3	703.73	0.06	0	694.52
	min	38.42	2.29e3	0	26.8	689.41	322.27	0	0	58.06
	max	2.16e3	9.98e3	0	639.65	6.26e3	3.24e3	0.27	0	2.73e3
F5	avg	0	0	0	0	0	0	0	0	0
	std	0	0	0	0	0	0	0	0	0
	min	0	0	0	0	0	0	0	0	0
	max	0	0	0	0	0	0	0	0	0
F6	avg	3.5	8.11	0.49	1.15	5.52	3.01	5.4	0.98	9.48
	std	0.86	3.53	2.19	2.98	4.46	4.57	5.01	3.02	2.17
	min	0.02	0.07	0	0	0.02	0	0	0	0.3
	max	4	9.82	9.81	9.81	9.83	9.81	9.81	9.81	10.14
F7	avg	0.14	0.06	0.23	3.89	0.05	0.2	0.57	0.18	0
	std	0.17	0.1	0.18	3.74	0.05	0.49	2.33	0.22	0
	min	0.01	0	0.06	0.48	0	0.01	0	0.01	0
	max	0.71	0.37	0.85	13.74	0.19	2.17	10.46	0.8	0.01
F8	avg	20.37	20.35	20.62	20.37	20.39	20.31	20.23	20.31	20.35
	std	0.07	0.09	0.63	0.08	0.08	0.09	0.18	0.11	0.1
	min	20.22	20.15	20.01	20.26	20.19	20.14	20	20.1	20.08
	max	20.49	20.5	21.61	20.5	20.53	20.47	20.48	20.51	20.48
F9	avg	1.86	1.33	0.86	3.04	1.14	1.72	3.74	1.94	1.04
	std	0.64	1.04	0.79	1.13	0.91	0.81	1.77	0.86	0.33
	min	0.39	0.01	0	0.44	0.07	0.31	0	0.23	0
	max	2.82	4.48	2.5	4.39	2.93	3.13	6.22	3.73	1.74
F10	avg	0.08	0.23	0.05	0.57	0.09	0.14	0.01	0.06	0.05
	std	0.06	0.14	0.02	0.38	0.04	0.06	0.01	0.04	0.03
	min	0.02	0.08	0	0.12	0.01	0.07	0	0	0.01
	max	0.25	0.52	0.09	1.5	0.19	0.31	0.06	0.15	0.12
F11	avg	1.14	0.17	0	0	2.25	3.63	0	0.05	2.09
	std	1.45	0.37	0	0	1.5	1.75	0	0.22	1.2
	min	0	0	0	0	0	0.99	0	0	0
	max	4.97	0.99	0	0	4.97	8.95	0	0.99	4.97
F12	avg	4.19	4.65	6.77	9.32	3.8	6.12	5.24	11.49	3.68
	std	1.98	0.98	2.55	3.21	1.35	1.94	1.98	4.41	2.22
	min	0.99	2.97	1.99	4.97	1.99	2.98	2.99	4.97	0.99
	max	9.12	6.75	10.94	17.1	5.97	9.95	8.97	18.9	8.95
F13	avg	5.92	5.16	9.86	14.6	4.94	11.57	7.4	18.79	6.78
	std	4	2.5	5.94	5.82	2.72	5.45	3.97	10.05	2.82
	min	1	1.69	2.98	4.64	0	2.78	1.99	1.99	1.99
	max	15.07	10.16	21.37	22.56	9.89	24.22	18.63	42.27	11.4
F14	avg	159.47	6.59	0.03	15.99	83.15	415.45	0.06	3.93	320.81
	std	127.16	5.14	0.04	30.55	74.01	144.53	0.05	3.86	95.76
	min	0.36	0.86	0	0.12	0.45	142.42	0	0.12	135.26
	max	386.23	23.05	0.12	136.5	243.23	689.08	0.12	11.95	544.7
F15	avg	251.68	566.59	606.31	609.08	301.11	375.85	528.06	454.28	190.68
	std	128.61	150.18	361.96	249.62	174.4	211.42	182.45	241.92	100.53
	min	126.97	121.43	125.72	196.7	31.83	136.78	260.43	164.28	123.37
	max	505.02	834.8	1.83e3	1.02e3	694.32	882.92	810.31	1111	544.25
F16	avg	0.66	1.09	6.63	0.95	0.35	0.44	0.31	1.05	0.88
	std	0.19	0.18	5.77	0.25	0.14	0.24	0.13	0.25	0.28
	min	0.39	0.85	0	0.46	0.15	0.18	0.06	0.56	0.48
	max	1.03	1.58	16.4	1.35	0.68	1.14	0.5	1.47	1.43
F17	avg	10.7	11.73	9.62	11.63	14.19	15.37	9.62	5.33	13.97
	std	0.5	0.57	2.26	2.82	1.51	2.77	2.26	5.11	1.91
	min	10.15	10.88	0.01	1.31	11.77	11.76	0	0.04	10.94
	max	11.72	12.93	10.12	15.73	16.91	23.99	10.12	10.73	17.8
F18	avg	20.5	26.31	50.78	21.92	16.48	17.19	15.52	13.7	18.1
	std	3.58	3.92	40.57	5.78	1.54	4.02	3.3	6.35	5.14
	min	14.6	18.81	6.41	12.44	14.29	12.96	5.11	3.72	11.6
	max	28.06	31.8	134.29	32.63	20.29	25.92	22.06	28.49	32.68
F19	avg	0.53	0.63	4.49	0.56	0.54	0.81	0.25	0.43	0.71
	std	0.09	0.09	3.01	0.18	0.1	0.33	0.04	0.12	0.23

Function	Statistic	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
	min	0.37	0.4	0.49	0.24	0.37	0.36	0.18	0.22	0.45
	max	0.69	0.78	12.05	0.91	0.67	1.77	0.34	0.59	1.23
F20	avg	1.7	2.02	3.82	2.23	1.97	2.13	2.74	2.76	1.47
	std	0.65	0.4	1.2	0.43	0.47	0.59	0.5	0.43	0.62
	min	0.64	1.63	1.99	1.5	0.84	0.79	1.9	2	0.62
	max	3.04	3.08	5.19	2.92	2.71	3.01	3.46	3.48	3.43
F21	avg	400.19	393.7	300.1	309.55	400.19	400.19	400.19	200	400.19
	std	0	29.06	102.7	119.78	0	0	0	0	0
	min	400.19	270.25	200	100	400.19	400.19	400.19	200	400.19
	max	400.19	400.19	400.19	400.19	400.19	400.19	400.19	200	400.19
F22	avg	101.25	35.29	670.36	38.03	84.46	411.25	1.21	108.81	171.56
	std	97.92	37.04	717.32	28.02	62.72	240.07	2.85	20.94	139.38
	min	5.02	5.02	9.6	1.4	22.87	42.74	0	49.15	10.19
	max	360.27	109.54	2.55e3	106.99	211.65	847.88	8.84	145.46	517.92
F23	avg	344.24	231.43	972.85	564.21	199.7	618.48	603.4	527.46	196.1
	std	162.39	140.65	1.08e3	284.45	173.89	183.39	297.25	240.81	142.29
	min	37.29	84.29	53.89	162.26	19.17	154.22	91.51	95.91	12.83
	max	638.22	585.74	3.48e3	1.08e3	518.6	987.78	1.24e3	944.74	420.59
F24	avg	187.89	199.31	193.46	142.5	200.15	175.84	204.5	206.91	202.38
	std	33.99	9.86	27.2	39.04	0.05	39.95	5.17	0.57	4.9
	min	111.09	164.83	103.9	109.13	200.07	108.79	200	205.94	200
	max	216	209.2	206.8	211.1	200.23	207.53	214.19	208.01	213.21
F25	avg	201.78	199.35	187.18	193.2	200.17	200.63	204.65	204.81	200
	std	3.17	5.6	33.84	29.79	0.07	0.82	4.76	0.56	0
	min	200.04	177.5	107.19	110.32	200.09	200.02	200	204.54	200
	max	210.61	208.1	204.55	215.66	200.36	202.9	214.77	206.97	200.01
F26	avg	114.82	120.16	134.19	113.48	118.26	118.56	124	191.56	209.17
	std	29.96	32.71	44.25	6.87	35.27	28.39	39.02	26.26	89.69
	min	100.04	103.17	101.99	103.98	101	101.99	102.99	104.97	101.99
	max	200.03	200.03	200.02	127.88	200.03	200.02	200.02	200.02	308.73
F27	avg	395.22	300.08	310.15	332.7	301.25	328.36	314.52	434.26	395
	std	21.36	0.19	30.73	28.73	0.41	43.41	45.61	91.8	22.36
	min	304.46	300	300.07	301.9	300.74	300.15	300	300	300
	max	400	300.8	400	400	302.47	400	473.93	518.08	400
F28	avg	305	295.04	260	250	290	295	300	260	325
	std	82.56	14.2	82.08	88.85	44.72	51.04	0	82.08	44.43
	min	100	243.54	100	100	100	100	300	100	300
	max	400	300	300	300	300	400	300	300	400

Table 5: Statistical performance results on CEC13 100D benchmark functions.

Function	Stat.	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
F1	avg	0	0	0	0	0	0	0	0	0
	std	0	0	0	0	0	0	0	0	0
	min	0	0	0	0	0	0	0	0	0
	max	0	0	0	0	0	0	0	0	0
F2	avg	3.34e6	2.86e7	9.42e5	1.25e6	1.05e7	1.98e6	4.38e5	3.87e6	2.64e6
	std	4.17e5	5.53e6	2.66e5	2.71e5	3.65e6	3.93e5	1.74e5	9.46e5	6.06e5
	min	2.63e6	2.07e7	5.30e5	8.68e5	5.58e6	1.44e6	1.91e5	2.46e6	1.59e6
	max	4.07e6	4.01e7	1.62e6	1.86e6	2.46e7	2.89e6	7.40e5	5.46e6	3.77e6
F3	avg	2.50e8	1.15e8	2.34e8	2.74e9	5.29e7	2.31e8	1.05e9	2.60e9	1.10e9
	std	1.25e8	6.59e7	2.50e8	2.48e9	3.92e7	1.75e8	4.88e8	1.82e9	7.30e8
	min	7.18e7	2.48e7	2.25e7	5.55e8	1.09e7	4.13e7	1.90e8	6.76e8	6.35e7
	max	4.54e8	2.70e8	1.09e9	1.10e10	1.44e8	6.36e8	1.96e9	7.09e9	2.41e9
F4	avg	1.29e3	1.36e3	1.12	8.88	5.41e3	233.4	4.48e4	0.62	501.51
	std	371.85	389.06	1.21	5.46	1.05e3	96.47	3.16e4	0.59	118.91
	min	705.33	849.78	0.11	2.2	3.75e3	117.88	29.04	0.05	278.52
	max	1928.9	2.51e3	5.38	21.36	7292.7	509.09	1.09e5	2.29	734.79
F5	avg	0	0	0	0	0.01	0	0	0	0
	std	0	0	0	0	0	0	0	0	0
	min	0	0	0	0	0.01	0	0	0	0
	max	0	0	0	0	0.01	0	0	0	0
F6	avg	189.37	202.31	148.04	118.63	207.2	101.2	162.75	207.2	219.83
	std	33.31	25.9	53.49	67.14	35.18	70.88	61.45	28.24	7.8
	min	98.53	167.15	14.37	13.91	168.98	11	15.63	158.22	215.17

Function	Stat.	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
	max	242.95	250.88	235.8	256.63	294.05	223.74	245.71	276.55	247.78
F7	avg	22.85	45.57	55.85	102.91	16.24	37.13	111.65	128.29	10.25
	std	3.67	8.88	8.58	16.43	5.26	5.47	18.99	24.21	3.9
	min	16.8	30.67	38.83	78.75	8.81	26.14	71.7	83.61	3.35
	max	30.14	63.36	69.49	129.02	26.91	46.05	134.95	184.68	20.35
F8	avg	21.31	21.25	21.48	21.27	21.32	21.29	21.26	21.25	21.3
	std	0.03	0.03	0.2	0.06	0.03	0.03	0.05	0.04	0.03
	min	21.24	21.2	21.11	21.17	21.28	21.24	21.13	21.17	21.25
	max	21.35	21.31	21.75	21.34	21.37	21.33	21.33	21.33	21.35
F9	avg	71.91	125.33	79.66	112.76	57.75	66.51	141.08	119.23	30.62
	std	8.14	4.05	9.46	6.53	13.21	9.51	3.09	7.61	3.42
	min	53.94	116.17	66.95	100.5	41.52	54.77	133.42	105.46	22.89
	max	94.29	132.11	98.62	125.8	93.2	85.56	145.85	130.58	36.5
F10	avg	0.28	0.14	0.07	0.12	1.79	0.06	0.05	0.27	1.18
	std	0.25	0.1	0.04	0.09	0.21	0.03	0.03	0.28	4.79
	min	0.03	0.01	0.03	0.01	1.52	0.01	0.01	0.03	0.01
	max	1.04	0.38	0.21	0.43	2.39	0.16	0.11	1.06	21.52
F11	avg	78.5	4.06	9.65	3.74	175.46	247.69	12.39	396.34	62.05
	std	19.12	2.26	9.32	1.63	20.32	29.03	22.55	53.34	11.21
	min	51.74	0.03	2.98	0	145.27	199.99	0.99	323.36	44.77
	max	117.41	8.95	46.81	6.29	215.91	314.41	101.48	534.29	85.19
F12	avg	101.55	173.79	240.08	401.76	129.74	267.79	304.64	481.56	69.34
	std	15.94	30.33	30.21	69.59	17.31	44.78	57.58	74.2	13.81
	min	67.66	120.35	191.03	275.6	99.62	201.98	216.03	334.3	45.11
	max	129.34	227.9	326.35	527.32	166.36	371.12	418.94	609.9	94.79
F13	avg	332.48	340.26	481.26	678.58	321.65	539	627.26	823.54	195.85
	std	60.64	55.54	65.64	86.66	48.9	73.67	60.49	88.07	30.39
	min	191.26	201.98	373.42	521.59	236.1	414.45	520.06	649	150.35
	max	465.21	427.66	604.37	845.26	423.94	694.15	719.86	1.00e3	278.19
F14	avg	3.72e3	691.09	4.28e3	65.19	8.89e3	9.92e3	23.82	8.98e3	6.38e3
	std	1.12e3	397.26	2.11e3	52.3	775.67	842.39	48.62	989.33	1.55e3
	min	1.69e3	34.69	1.45e3	33.48	7.76e3	8.27e3	0.09	6.81e3	4091.5
	max	5.76e3	1.24e3	9.26e3	207.11	1.02e4	1.17e4	118.63	1.08e4	1.04e4
F15	avg	1.55e4	1.55e4	1.41e4	1.43e4	1.19e4	1.26e4	1.40e4	1.54e4	9.01e3
	std	2.23e3	1.07e3	983.62	1.00e3	976.55	1.22e3	636.16	1.12e3	1.38e3
	min	1.16e4	1.37e4	1.19e4	1.21e4	1.04e4	1.06e4	1.26e4	1.27e4	7.25e3
	max	2.09e4	1.88e4	1.63e4	1.59e4	1.45e4	1.46e4	15169	1.73e4	1.20e4
F16	avg	3.91	2.49	1.4	2.05	1.24	1.85	1.75	1.9	2.23
	std	0.24	0.33	0.3	0.34	0.28	0.96	0.17	0.56	0.91
	min	3.27	1.79	0.86	1.35	0.84	0.65	1.45	1.13	1.31
	max	4.25	3.02	2.02	2.59	1.71	3.61	2.09	3.63	4.02
F17	avg	143.66	149.41	278.69	140.3	258.66	288.81	101.6	568.24	168.89
	std	20.64	10.31	80.98	15.93	50.16	31.88	0.16	48.72	11.59
	min	122.3	134.02	162.93	115.9	181.45	235.58	101.56	488.26	146
	max	197.44	175.45	397.55	189.29	395.77	369.48	102.26	667.86	193.36
F18	avg	708.18	489.26	297.37	389.94	327.92	267.73	438.51	697.32	197.94
	std	134.72	40.83	36.73	48.58	24.41	26.72	40.15	77.16	22.28
	min	408.65	398.48	231.07	293.9	291.71	217.86	380.98	560.81	164.08
	max	871.65	563.67	372.71	491.3	402.71	314.67	514.72	846.59	241.3
F19	avg	8.47	9.28	18.43	11.11	17.74	14.32	10.02	147.39	11.66
	std	0.79	0.9	9.43	2.04	4.17	2.33	1.69	39.54	1.51
	min	7.07	8.1	7.87	8.45	12.89	9.88	7.38	76.44	9.03
	max	10.15	11.45	43.03	17.04	28.53	20.07	12.66	247.76	13.93
F20	avg	49.94	50	50	50	50	49.93	50	50	47.84
	std	0.16	0	0	0.02	0	0.18	0	0	4.6
	min	49.51	50	50	49.93	50	49.51	50	50	36.43
	max	50	50	50	50	50	50	50	50	50
F21	avg	400.05	350	345	370	401.24	400	390	335.01	400
	std	0.2	51.3	51.04	47.02	40.54	0	30.78	48.94	0
	min	400	300	300	300	326.09	400	300	300	400
	max	400.91	400	400	400	426.62	400	400	400.12	400
F22	avg	3.69e3	554	4.37e3	155.91	1.05e4	1.36e4	54.11	9.16e3	5.52e3
	std	1.04e3	292.52	2.68e3	92.9	1.11e3	1.77e3	49.89	832.2	1.48e3
	min	2.23e3	79.31	1.50e3	55.06	8.64e3	1.11e4	13.52	7.75e3	3.00e3
	max	5.93e3	1077	1.24e4	371.96	1.29e4	1.66e4	121.21	1.07e4	9.30e3
F23	avg	1.57e4	1.64e4	1.63e4	2.07e4	1.44e4	1.57e4	1.79e4	1.59e4	9.10e3
	std	2.01e3	1.19e3	1.23e3	1.94e3	1.50e3	1.82e3	1.71e3	1483	1.54e3
	min	1.27e4	1.44e4	1.43e4	1.70e4	1.11e4	1.22e4	1.53e4	1.33e4	6.40e3
	max	2.03e4	1.84e4	1.80e4	2.36e4	1.70e4	1.84e4	2.26e4	1.88e4	1.22e4
F24	avg	396.96	305	333.78	505.7	245.38	269.58	452.57	455.25	241.67
	std	29.59	24.53	17.33	20.61	14.72	11.16	32.83	25.97	23.5
	min	341.21	266.01	299.47	461.62	217.25	251.26	412.07	395.97	179.72
	max	445.72	345.58	373.92	535.64	276.93	293.79	539.26	494.66	284.71
F25	avg	479.68	505.53	463.01	644.75	447.73	411.32	510.62	599.99	390.33
	std	26.43	51.98	19.66	23.14	12.88	115.05	40.42	26	11.69

Function	Stat.	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
	min	435.18	437.63	410.31	589.3	418.94	216.31	471.57	549.59	369.42
	max	552.17	576.82	501.69	681	473.98	503.96	639.69	641.93	412.87
F26	avg	468.11	444.33	465.76	333.27	388.11	404.34	636.47	601	353.89
	std	14.79	46.81	20.82	186.23	20.36	11.93	19.59	23.76	46.46
	min	442.4	407.61	421.74	200.09	349.39	390.43	576.92	557.99	178.84
	max	496.99	581.17	499.25	597.94	428.55	428.79	657.47	636.95	387.33
F27	avg	1980.6	2141.5	2.04e3	3.37e3	1.18e3	1.18e3	3.48e3	3.31e3	773.93
	std	324.98	496.22	226.16	203.12	322.03	88.14	488.68	202.96	315.36
	min	1.29e3	1.66e3	1.68e3	2920.7	621.23	1.02e3	2.43e3	2.94e3	398.74
	max	2.36e3	3.31e3	2.40e3	3.73e3	1.67e3	1347	3.96e3	3.65e3	1.32e3
F28	avg	2.95e3	3.59e3	3.31e3	4.09e3	3.14e3	2.57e3	3.85e3	5876.2	3.46e3
	std	776.27	1.09e3	1.05e3	957.43	983.8	23.88	1.16e3	1.99e3	1.09e3
	min	2.59e3	2.50e3	2.55e3	3.22e3	2.49e3	2542.8	2.79e3	3.57e3	2.48e3
	max	4.78e3	4.70e3	4.98e3	6.06e3	4.66e3	2.61e3	5.64e3	9.83e3	4.68e3

Table 6: Statistical performance results on CEC14 10D benchmark functions.

Function	Statistic	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
F1	avg	2.64e4	2.17e4	0	1.66e3	1.62e4	1.17e4	0	0	4.24e3
	std	3.15e4	2.13e4	0	1.43e3	1.18e4	2.17e4	0	0	6.22e3
	min	297.38	644.47	0	154.76	1.62e3	49.57	0	0	29.36
	max	1.01e5	8.53e4	0	5852	3.95e4	9.00e4	0	0	2.65e4
F2	avg	592.37	653.39	0	482.4	745.7	383.43	0	0	1.06e3
	std	540.69	761.22	0	664.68	738.91	534.21	0	0	2.00e3
	min	0.27	0.06	0	0.05	33.96	11.71	0	0	0
	max	1.82e3	2.41e3	0	2.57e3	2284.3	2.29e3	0	0	8936.8
F3	avg	34.3	276.99	0	24.17	148.88	81.96	0	0	23.88
	std	62.13	459.48	0	44.2	243.53	140.38	0	0	35.57
	min	0.01	0.35	0	0.03	2.22	0.83	0	0	0.77
	max	240.1	1.95e3	0	144.28	953.1	598.12	0	0	116.35
F4	avg	4.96	13.7	7.39	0.28	18.42	21.37	28.04	1.08	34.78
	std	1.89	13.66	14.11	0.96	16.85	16.9	13.86	1.93	0
	min	0.25	0.4	0	0	0.12	0.15	0	0	34.78
	max	6.38	34.78	34.78	4.34	34.78	34.78	34.78	4.34	34.78
F5	avg	18.15	16.86	20.98	16.2	18.01	20.05	20.01	20.12	20.29
	std	5.78	6.67	0.39	8.11	6.16	0.1	0	0.03	0.1
	min	0	0	20.09	0	0	20	20	20.06	20.11
	max	20.09	20.22	21.57	20.23	20.02	20.36	20.01	20.19	20.48
F6	avg	0.04	0	0	0.7	0.02	0	0.21	0.9	0
	std	0.08	0	0.01	0.61	0.01	0.01	0.35	0.24	0
	min	0	0	0	0	0.01	0	0	0	0
	max	0.35	0	0.06	1.88	0.04	0.02	1.14	1.35	0
F7	avg	0.02	0.02	0.03	0.08	0.04	0.13	0.02	0.05	0.04
	std	0.02	0.02	0.02	0.04	0.02	0.06	0.02	0.02	0.02
	min	0	0	0.01	0.01	0.01	0.02	0	0.01	0
	max	0.06	0.06	0.08	0.18	0.1	0.26	0.07	0.11	0.09
F8	avg	2.44	0	0	0	2.54	6.02	0	0	2.24
	std	1.63	0	0	0	0.99	2.05	0	0	1.36
	min	0	0	0	0	0	2.98	0	0	0
	max	6.96	0	0	0	3.98	9.95	0	0	4.97
F9	avg	3.63	1.93	7.56	6.4	3.78	6.27	3.73	6.91	3.78
	std	1.34	1.31	10.36	1.98	1.53	2.37	1.16	3.19	1.43
	min	0.99	0	0.99	2.98	0.99	1.99	1.99	1.99	0
	max	5.97	3.75	35.67	10.25	5.97	11.94	5.97	12.93	5.97
F10	avg	196.89	1.69	0.02	2.38	39.21	293.89	0.04	4.34	250.85
	std	87.79	2.55	0.03	3.01	62.96	157.37	0.05	6.01	46.63
	min	11.83	0.12	0	0.19	0.31	6.83	0	0.06	125.39
	max	366.21	10.24	0.06	10.24	224.04	548.17	0.12	21.82	362.14
F11	avg	179.39	29.6	796.01	201.83	85	276.42	145.78	156.72	111.81
	std	88.6	28.99	991.61	140.71	98.11	190.49	110.14	125.28	112.91
	min	11.77	3.51	7.02	11.95	0.38	11.89	10.29	10.12	11.83
	max	350.64	126.73	3.80e3	491.18	265.65	667.32	347.63	458.89	370.37
F12	avg	0.18	0.33	3.97	0.24	0.11	0.19	0.07	0.17	0.1
	std	0.1	0.08	3.72	0.12	0.06	0.17	0.02	0.12	0.07
	min	0.06	0.21	0.07	0.07	0.04	0.01	0.04	0.02	0.01

Function	Statistic	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
	max	0.5	0.54	13.1	0.57	0.24	0.68	0.1	0.47	0.24
F13	avg	0.06	0.07	0.69	0.12	0.06	0.06	0.1	0.1	0.09
	std	0.02	0.02	0.38	0.04	0.02	0.03	0.03	0.03	0.03
	min	0.03	0.03	0.15	0.06	0.02	0.01	0.05	0.04	0.05
	max	0.09	0.1	1.21	0.19	0.09	0.13	0.15	0.16	0.14
F14	avg	0.08	0.08	2.22	0.11	0.1	0.1	0.12	0.1	0.05
	std	0.03	0.02	2.09	0.04	0.03	0.05	0.04	0.03	0.02
	min	0.04	0.05	0.12	0.02	0.03	0.02	0.06	0.03	0.03
	max	0.17	0.14	7.56	0.18	0.14	0.22	0.23	0.16	0.09
F15	avg	0.65	0.78	2.39	0.8	0.62	0.8	0.42	0.61	0.73
	std	0.15	0.14	3.09	0.23	0.13	0.32	0.1	0.21	0.21
	min	0.28	0.54	0.4	0.24	0.33	0.17	0.2	0.31	0.35
	max	0.78	1.04	12.15	1.18	0.83	1.31	0.69	1.13	1.12
F16	avg	1.57	0.73	4.03	1.55	0.97	2.01	1.75	1.38	1.55
	std	0.51	0.55	1.03	0.49	0.45	0.42	0.33	0.45	0.87
	min	0.75	0.26	0.67	0.88	0.4	1.3	1.23	0.44	0.21
	max	2.49	2	5.34	2.59	1.8	2.73	2.4	1.99	2.94
F17	avg	685.46	3.61e3	120.41	1.07e3	762.89	800.73	76.3	4.05	726.79
	std	472.48	2.78e3	350.55	876.91	370.24	488.94	90.32	6.36	492.16
	min	105.34	466.54	0	158.09	236.89	181.39	0	0.21	62.64
	max	1.62e3	1.04e4	1270.2	3.36e3	1.68e3	2.34e3	353.49	22.55	2.01e3
F18	avg	2511.4	1361.8	1.34	168.13	1.61e3	3.98e3	1	1.05	2.81e3
	std	1.63e3	2.12e3	1.61	197.43	1.71e3	2.31e3	1.23	0.57	1.93e3
	min	532.92	58.72	0.06	8.01	25.02	448.43	0.03	0.01	700.59
	max	6.15e3	9.08e3	4.94	600.66	6.04e3	8.82e3	5.15	2.14	7.39e3
F19	avg	0.82	0.14	2.14	0.5	0.9	1.5	0.5	0.05	1.02
	std	0.46	0.21	1.05	0.4	0.42	0.15	0.56	0.03	0.38
	min	0.03	0.02	0.23	0.03	0.27	0.98	0.02	0.01	0.05
	max	1.5	0.91	4.3	1.03	1.61	1.66	1.8	0.11	1.5
F20	avg	171.86	230.06	1.33	23.41	277.11	169.75	0.32	0.19	112.36
	std	351.03	258.41	1.67	19.02	509.96	175.27	0.3	0.26	138.61
	min	7.03	11.03	0.01	2.85	13.19	13.43	0.05	0	6.76
	max	1.23e3	1.05e3	5.83	72.08	2.21e3	536.03	1.3	1.08	449.27
F21	avg	82.47	275.7	0.75	59.89	151.9	435.14	19.91	0.36	67.22
	std	66.69	474.55	0.9	65.99	129.78	217.27	38.25	0.26	75.48
	min	17.49	1.57	0.03	0.18	37.65	96.37	0	0	2.03
	max	283.28	1.90e3	4.25	193.28	408.41	1.06e3	119.23	0.81	250.82
F22	avg	56.87	1.38	12.93	9.99	9.87	29.1	2.23	0.26	127.33
	std	48.98	4.44	38.51	9.84	9.74	27.05	6.17	0.25	26.01
	min	0.39	0.02	0.2	0.22	0.22	17.35	0.08	0	24.62
	max	140.57	20.2	171.66	20.85	20.68	141.02	20.35	0.73	141.44
F23	avg	329.46	319.52	329.46	329.46	329.46	329.46	329.46	329.46	329.46
	std	0	44.45	0	0	0	0	0	0	0
	min	329.46	130.65	329.46	329.46	329.46	329.46	329.46	329.46	329.46
	max	329.46	329.46	329.46	329.46	329.46	329.46	329.46	329.46	329.46
F24	avg	108.87	106.78	110.59	114.37	108.84	107.99	111.07	117.61	108.32
	std	2.38	2.79	3.77	3.68	2.14	4.97	2.48	3.8	3.03
	min	105.43	100.02	100	107.18	105.43	100	107.18	110.81	100
	max	112.51	112.1	116.8	124.53	111.58	117.1	117.47	123.19	113.25
F25	avg	123.97	170.01	119.33	129.89	129.55	146.44	148.25	170.61	110.37
	std	24.03	33.31	19.56	14.35	25.31	22.99	39.29	39.65	15.78
	min	100	110.96	108.71	113.35	110.02	118.88	111.5	109.96	100
	max	199.04	201.35	201.13	176.72	199.3	199.36	201.52	201.37	152.65
F26	avg	100.06	100.07	100.93	100.11	100.06	100.05	100.09	100.11	100.09
	std	0.02	0.02	0.52	0.04	0.02	0.02	0.02	0.03	0.03
	min	100.02	100.03	100.1	100.06	100.03	100.01	100.06	100.06	100.05
	max	100.1	100.12	102.11	100.2	100.1	100.09	100.14	100.18	100.15
F27	avg	172.38	116.18	85.38	123.37	1.53	63.3	205.91	61.34	300.1
	std	178.66	171.86	137.99	171.33	0.41	125.45	175.09	146.03	79.29
	min	0.71	1.05	2.17	1.01	0.86	1.04	1.14	0.72	1.12
	max	400.28	400.3	400.13	400.18	2.61	311.96	400.14	400.17	400.35
F28	avg	326.02	394.15	363.4	394.7	371.8	301.6	422.51	361.23	226.89
	std	111.27	50.74	27.06	34.25	28.76	143.33	52.51	6.17	162.25
	min	100.13	356.83	356.83	357.02	356.83	100.58	369.37	356.83	100.11
	max	515.89	486.36	477.99	496.82	478.06	484.46	487.89	375.62	504.39
F29	avg	343.36	296.58	236.18	283.97	333.44	341.05	8.64e4	215.68	8.66e4
	std	57.57	44.6	86.39	56.16	68.92	52.09	3.86e5	21.52	3.86e5
	min	219.21	235.12	131.75	213.02	237.61	272.63	131.77	147.99	246.91
	max	404.79	388.22	593.02	436.2	517.92	460.99	1.72e6	226.71	1.72e6
F30	avg	885.67	564.56	464.85	676.35	635.66	1.52e3	497.84	485.67	922.27
	std	355.86	89.05	1.17	128.92	64.47	330.89	43.99	22.08	388.54
	min	447.63	499.17	462.73	523.05	546.2	975.62	454.36	462.52	493.02
	max	1.86e3	857.55	467.38	961.47	773.68	2.18e3	594.42	525.03	1.70e3

Table 7: Statistical performance results on CEC14 100D benchmark functions.

Function	Stat.	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
F1	avg	3.19e6	5.37e7	8.82e5	9.83e5	1.41e7	3.49e6	2.57e5	3.15e6	2.11e6
	std	7.59e5	1.34e7	2.21e5	3.04e5	3.41e6	4.48e5	1.94e5	1.36e6	6.22e5
	min	1.76e6	3.09e7	5.02e5	4.78e5	7.80e6	2.82e6	3.41e4	1.31e6	9.56e5
	max	5.22e6	8.28e7	1.35e6	1.55e6	2.20e7	4.23e6	1.08e6	6.57e6	3.09e6
F2	avg	1.14e4	1.80e4	60.33	2.03e4	1.68e4	1.97e4	0	1.45e4	1.59e4
	std	6.02e3	1.37e4	132.03	1.50e4	1.42e4	1.16e4	0	1.56e4	1.41e4
	min	3.28e3	2.06e3	0.24	175.67	2.02e3	1.94e3	0	282.16	0
	max	2.81e4	4.81e4	601.02	6.67e4	4.94e4	4.04e4	0	6.10e4	4.73e4
F3	avg	989.46	117.73	794.14	0.19	2937.5	93.58	0.92	279.86	79.49
	std	502.89	123.78	740.09	0.25	1.18e3	48.61	2.94	366.26	60.16
	min	131.61	0.91	18.05	0.01	953.02	6.5	0	7.65	4.71
	max	1834.5	531.86	2.27e3	0.87	5.90e3	188.74	11.89	1.27e3	249.12
F4	avg	202.72	182.28	141.79	142.92	221.44	229.03	128.56	199.87	219.41
	std	63.73	43.06	36.49	51.68	36.75	57.51	46.7	51.77	29.63
	min	92.63	144.36	92.21	76.25	144.58	100.9	0.11	92.39	158.45
	max	303.19	254.28	245.44	248.79	291.05	295.69	205.38	291.81	256.19
F5	avg	21.09	20.61	20.12	20.1	20.06	20.3	20.04	20.31	20.61
	std	0.06	0.06	0.24	0.1	0.08	0.42	0	0.3	0.23
	min	20.96	20.48	20	20	20	20	20.03	20	20.32
	max	21.17	20.7	20.93	20.3	20.23	21.25	20.04	20.86	21.32
F6	avg	33.74	28.52	32.15	75.68	18.86	25.17	57.99	85.62	2.45
	std	3.69	14.63	6.77	6.88	5.04	3.27	6.63	6.73	1.6
	min	27.83	15.59	20.42	64.88	10.65	18.62	47.13	71.71	0.23
	max	42.11	68.42	43.78	91.45	29.32	31.55	71.41	104.67	5.43
F7	avg	0.01	0	0	0.01	0.01	0	0.01	0.01	0
	std	0.02	0	0	0.03	0	0	0.01	0.02	0
	min	0	0	0	0	0.01	0	0	0	0
	max	0.06	0	0.01	0.12	0.02	0.01	0.06	0.07	0
F8	avg	70.79	9.08	5.52	0.9	184.17	268.19	0.96	194.66	64.08
	std	17.15	5.58	2.91	0.92	28.43	22.43	1.91	21.41	8.25
	min	50.74	1.22	0	0	142.28	239.78	0	143.27	51.74
	max	111.44	21.89	10.94	3.11	264.67	325.35	8.95	227.85	78.6
F9	avg	101.74	174.46	192.67	354.82	155.19	257.99	213.94	271.97	72.59
	std	17.62	32.83	36	60.15	22.69	27.67	39.75	47.45	12.05
	min	59.7	110.59	131.33	270.63	114.42	199.99	159.2	196.01	57.71
	max	141.28	238.23	255.7	498.47	196.01	310.43	317.4	378.08	96.51
F10	avg	3.17e3	363.16	3.17e3	28.53	7.98e3	1.03e4	27.93	7.79e3	4724.7
	std	577.33	333.57	2.61e3	48.22	1.16e3	855.68	59.62	799.88	624.14
	min	2.12e3	12.97	365.23	3.14	5.23e3	8.68e3	0.02	6.24e3	3.32e3
	max	4.26e3	966.04	1.06e4	123.27	9.80e3	1.19e4	236.98	8.94e3	6.38e3
F11	avg	9.96e3	1.08e4	1.21e4	1.08e4	1.09e4	1.02e4	1.05e4	1.36e4	6.09e3
	std	1.29e3	840.76	1.07e3	666.69	677.06	1.25e3	550.4	1.39e3	1.11e3
	min	7704.4	9.44e3	9.68e3	9.71e3	9.74e3	7.52e3	9.30e3	1.02e4	3751.6
	max	1.19e4	1.25e4	1.37e4	1.22e4	1.23e4	12522	1.12e4	1.57e4	7.88e3
F12	avg	1.74	0.49	0.2	0.21	0.36	1.03	0.18	0.67	0.75
	std	0.4	0.07	0.11	0.06	0.12	0.96	0.02	0.23	0.35
	min	0.68	0.34	0.07	0.13	0.21	0.23	0.15	0.33	0.39
	max	2.37	0.59	0.58	0.36	0.64	3.5	0.21	1.06	1.48
F13	avg	0.39	0.28	0.71	0.46	0.38	0.46	0.5	0.44	0.34
	std	0.04	0.02	0.45	0.05	0.04	0.06	0.08	0.07	0.03
	min	0.34	0.23	0.35	0.38	0.31	0.34	0.38	0.32	0.28
	max	0.46	0.33	1.63	0.56	0.44	0.57	0.68	0.55	0.4
F14	avg	0.31	0.31	0.93	0.34	0.26	0.23	0.36	0.31	0.46
	std	0.02	0.02	1.65	0.03	0.02	0.03	0.11	0.03	0.28
	min	0.26	0.27	0.24	0.28	0.23	0.19	0.27	0.25	0.26
	max	0.34	0.36	6.19	0.37	0.3	0.29	0.79	0.37	0.98
F15	avg	14.6	31.22	18.61	42.79	22.68	14.68	38.71	133.07	12.67
	std	4.84	3.22	3.13	6.69	5.29	2.95	7.47	45.29	1.53
	min	9.8	24.05	13.39	29.89	15.84	9.48	23.92	60.01	8.62
	max	28.82	37.21	23.88	55.01	31.36	20.84	52.85	246.42	15.13
F16	avg	41.05	39.61	42.14	40.52	40.24	41.91	40.01	43.07	36.65
	std	1.46	0.72	0.87	0.75	1.39	1.27	0.72	1.62	1.27
	min	38.08	37.99	40.35	38.86	36.5	39.66	38.45	38.67	34.49
	max	43.51	40.68	43.63	41.51	42.26	44.39	41.21	44.89	39
F17	avg	5.53e5	2.99e6	1.56e5	1.95e5	1.68e6	4.92e5	3.49e4	4.22e5	4.81e5
	std	1.31e5	1.02e6	6.43e4	5.98e4	7.38e5	3.17e5	3.16e4	1.02e5	1.19e5
	min	2.11e5	1.65e6	6.31e4	9.64e4	8.63e5	2.77e5	8.15e3	2.08e5	2.85e5
	max	8.43e5	5.66e6	2.85e5	3.20e5	4.16e6	1.42e6	1.12e5	5.70e5	6.59e5
F18	avg	773.84	456.29	927.48	690.27	1.36e3	940.2	1.49e3	2.14e3	8.67e3
	std	609.2	409.68	1.14e3	314.23	758.92	305.53	1.56e3	1.78e3	6.21e3
	min	254.85	67.87	94.49	374.51	535.07	592.85	342.62	308.58	296.88
	max	2.29e3	1.23e3	3.84e3	1.75e3	3.11e3	1.57e3	5.54e3	5.99e3	1.70e4
F19	avg	57.89	92.78	86.31	70.71	99.48	79.52	95.26	99.58	97.2

Function	Stat.	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
	std	11.74	10.7	20.8	26.73	11.73	20.74	23.81	19.85	3.21
	min	48.7	54.27	17.48	31.87	65.58	38.2	31.44	60.13	93.55
	max	106.08	109.2	106.27	116.59	108.26	127.63	120.63	124.32	104.68
F20	avg	718.3	1.49e3	1.55e3	516.75	2.00e3	568.87	6.78e3	781.04	414.77
	std	164.44	1187.6	729.97	77.69	739.09	73.21	5.30e3	319.79	89.15
	min	363.46	404.85	675.34	363.54	807.46	483.8	446.98	327.55	262.22
	max	1.03e3	5.52e3	3.68e3	656.86	3.48e3	721.87	2.10e4	1.79e3	538.3
F21	avg	3.25e5	2.48e6	6.16e4	8.05e4	1.32e6	1.99e5	1.77e4	2.12e5	2.48e5
	std	7.85e4	1.32e6	2.90e4	2.84e4	4.45e5	4.66e4	2.91e4	8.04e4	8.02e4
	min	1.77e5	4.72e5	2.65e4	2.41e4	4.72e5	1.23e5	2.58e3	8.37e4	9.75e4
	max	4.68e5	4.64e6	1.21e5	1.39e5	2.10e6	2.84e5	1.27e5	3.76e5	4.27e5
F22	avg	1.24e3	1.32e3	1.76e3	1.95e3	1.18e3	1.17e3	1.67e3	1.82e3	523.69
	std	296.08	270.96	414.35	345.96	254.8	328.86	250.39	475.36	252.3
	min	537.95	886.08	866.49	1.20e3	688.17	552.59	1.22e3	915.66	152.71
	max	1.71e3	1.95e3	2.36e3	2.60e3	1.67e3	1.61e3	2.21e3	2.99e3	1.10e3
F23	avg	345.04	348.23	348.23	348.23	348.43	348.23	348.23	348.23	348.23
	std	0	0	0	0	0.13	0	0	0	0
	min	345.04	348.23	348.23	348.23	348.27	348.23	348.23	348.23	348.23
	max	345.05	348.23	348.23	348.23	348.78	348.24	348.23	348.23	348.23
F24	avg	205.87	362.12	378.77	359.36	372.57	370.45	417.77	424.48	371.02
	std	1.92	3.49	4.43	7.53	4.57	4.8	8.05	7.03	3.59
	min	202.58	357.85	372.62	329.68	364.46	361.9	400.65	412.18	362.83
	max	208.29	369.53	388.17	367.36	383.13	380.2	437.77	435.08	375.42
F25	avg	200	249.88	229.71	255.34	250.03	200	283.49	264.85	240.09
	std	0	3.02	28.47	5.2	4.03	0	12.34	32.21	9.02
	min	200	242.24	200	247.57	239.49	200	242.66	200	228.05
	max	200	255.87	269.21	269.61	257.13	200	300.93	306.92	253.96
F26	avg	200	197.21	195.12	200.18	200.94	200.18	200.09	177.47	200.23
	std	0	22.71	22.29	0.03	0.26	0.05	0.01	137.09	0.07
	min	200	100.75	100.44	200.15	200.54	200.14	200.08	100.34	200.13
	max	200	203.14	200.13	200.26	201.75	200.37	200.12	507.64	200.45
F27	avg	1.13e3	652.06	891.66	2.08e3	489.59	584.72	1.59e3	2.23e3	523.79
	std	130.31	189.96	103.35	736.44	65.78	66.58	144.26	138.13	108.49
	min	889.76	457.44	728.91	408.77	386.42	412.09	1.36e3	2.04e3	360.11
	max	1.46e3	1.35e3	1.04e3	2.77e3	618.45	668.88	1.95e3	2.47e3	684.76
F28	avg	2.92e3	2.48e3	2.23e3	5.33e3	3222.2	4.92e3	3.31e3	5.37e3	2.57e3
	std	618.21	312.69	156.4	643.75	957.2	2.01e3	620.37	1.43e3	274.57
	min	2218.8	2.12e3	1.79e3	3.95e3	1.87e3	400	2.47e3	3.64e3	2.11e3
	max	4.31e3	3.14e3	2.44e3	6.33e3	5.63e3	7.81e3	4.76e3	8.14e3	3.34e3
F29	avg	474.45	2.84e3	1.54e3	1.49e3	5.85e3	3.81e3	1.45e3	2.02e3	5.16e3
	std	330.98	407.67	188.86	267.66	1.67e3	824.3	168.14	101.92	3.62e3
	min	262.46	2351.6	1.21e3	1.04e3	2.36e3	2910.5	805.95	1864.3	2.23e3
	max	1.26e3	4.12e3	1.84e3	1.96e3	9553.6	6.24e3	1.77e3	2.18e3	1.46e4
F30	avg	3.63e3	1.60e4	8.02e3	9382.7	1.08e4	1.17e4	9.87e3	2.16e4	7.25e3
	std	516.08	2.69e3	1.05e3	1038	1.26e3	1.16e3	1.26e3	1.04e4	1.08e3
	min	2.50e3	1.12e4	5.61e3	6.61e3	8657.3	9.28e3	6.91e3	1.01e4	5.33e3
	max	4.39e3	2.02e4	9472	1.17e4	1.28e4	1.42e4	1.17e4	4.71e4	9.16e3

Table 8: Statistical performance results on CEC17 10D benchmark functions.

Function	Statistic	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
F1	avg	847.27	1007.1	0	962.02	735.92	272.33	0	0	1.34e3
	std	1.34e3	1.90e3	0	904.69	1.14e3	375.21	0	0	1.84e3
	min	0.44	0.84	0	9.88	10.1	6.76	0	0	75.34
	max	5.38e3	6766.7	0	3.20e3	4.07e3	1.67e3	0	0	7.13e3
F2	avg	0	0	0	0	0	0	0	0	0
	std	0	0	0	0	0	0	0	0	0
	min	0	0	0	0	0	0	0	0	0
	max	0	0	0	0	0	0	0	0	0
F3	avg	1.22	4.36	0	0.19	4.18	1.07	0	0	0.4
	std	0.38	0.3	0	0.07	0.85	0.33	0	0	0.08
	min	0.38	3.95	0	0.06	1.39	0.37	0	0	0.27
	max	1.87	4.9	0	0.39	5.07	1.52	0	0	0.57
F4	avg	3.25	1.54	8.29	6.34	3.13	7.26	3.98	6.62	3.13
	std	1.61	1.02	12.81	2.42	1.56	2.22	1.71	3.25	1.42
	min	1	0	0	1.99	0	3.98	1.99	1.99	0.99

Function	Statistic	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
	max	6.96	3.52	44.26	13.93	5.97	11.94	7.96	12.93	6.96
F5	avg	0	0	0	0	0	0	0	0	0
	std	0	0	0	0	0	0	0	0	0
	min	0	0	0	0	0	0	0	0	0
	max	0.01	0	0	0	0	0	0	0	0
F6	avg	12.98	14.53	30.45	19.09	13.86	17.94	14.02	10.55	14.19
	std	1.27	1.32	25.27	2.95	0.92	3.68	1.06	5.23	1.57
	min	10.9	12.33	11.76	13.75	12.07	11.86	11.86	1.99	12.11
	max	15.64	17.26	89.33	25.67	15.33	27.12	16.01	17.32	16.92
F7	avg	3.85	1.5	8.8	6.61	4.23	5.12	4.23	7.11	2.79
	std	1.05	0.98	10.93	2.08	1.2	2.38	1.11	2.69	1.43
	min	1.99	0.01	0.99	2.98	1.99	0.99	1.99	1.99	0
	max	5.97	3.65	35.46	10.94	6.96	9.95	6.97	10.94	4.97
F8	avg	0	0	0	0	0	0	0	0	0
	std	0	0	0	0	0	0	0	0	0
	min	0	0	0	0	0	0	0	0	0
	max	0	0	0	0	0	0	0	0	0
F9	avg	250.91	15.53	903.02	203.72	165.78	500.05	136.17	119.83	269.26
	std	202.94	32.8	1.08e3	125.83	151.5	211.88	108.98	81.74	146.4
	min	0.37	1.47	0.44	3.73	0.25	125.21	3.74	18.6	3.73
	max	694.55	150.59	3.22e3	545.09	438.71	853.18	347.95	312.7	544.81
F10	avg	2.44	1.86	0.1	3.18	2.6	4.63	1.06	1.44	0.61
	std	1.49	0.87	0.31	1.59	1	2.75	1.29	0.99	0.75
	min	1.02	0.35	0	1	0.38	0.17	0	0	0
	max	6.06	3.38	0.99	6.69	4.06	11.11	3.7	3.98	2.01
F11	avg	1.05e4	9.34e3	40.46	9.55e3	7.48e3	7.83e3	317.18	52.31	8.20e3
	std	9.71e3	6.86e3	130.25	1.02e4	4.00e3	2.93e3	211.31	64.78	5.17e3
	min	531.84	3.26e3	0	963.21	1.11e3	2.98e3	11.59	0	2.29e3
	max	4.45e4	2.48e4	543.15	4.03e4	1.64e4	1.26e4	917.65	157.45	2.01e4
F12	avg	3.73e3	1.19e3	2.22	901.87	1.70e3	4.70e3	5.75	2.73	4.35e3
	std	2.32e3	1.32e3	4.21	1.74e3	1.37e3	3.13e3	2.6	1.96	2.44e3
	min	537.07	23.49	0	13.21	127.47	598.02	0	0.99	1.81e3
	max	8.00e3	4690	16.75	7.50e3	5.59e3	1.24e4	9.05	6.14	1.00e4
F13	avg	29.92	58.9	0	29.58	28.12	55.47	6.04	0.95	37.78
	std	11.44	93.34	0	17.83	12.85	53.2	8.76	0.76	12.85
	min	9.57	0.06	0	6.08	5.49	23.28	0	0	12.97
	max	46.14	420.37	0	87.13	49.04	241.44	21.82	1.99	65.95
F14	avg	19.81	119.25	0.05	14.56	50.37	86.3	0.26	0.77	51.38
	std	25.28	226.98	0.07	9.68	37.82	64.98	0.35	0.6	34.33
	min	3.37	2.63	0	3.3	12.94	18.5	0	0	17.44
	max	120.31	784.93	0.33	35.11	131.72	260.95	1.49	2.18	147.06
F15	avg	38.52	0.8	17.1	0.67	1.29	9.69	0.69	0.37	180.49
	std	55.32	0.35	45.99	0.27	0.47	26.41	0.41	0.29	99.77
	min	0.66	0.11	0.27	0.03	0.43	1.48	0.09	0.01	0.44
	max	121.52	1.34	161.94	1.11	2.14	121.76	2.07	1	373.31
F16	avg	19.07	0.86	76.16	7.4	9.54	27.82	0.35	0.98	31.78
	std	11.61	0.61	115.06	6.86	8.18	5.79	0.3	0.65	13.66
	min	2.43	0.31	0.64	0.19	1.4	14.68	0	0	6.09
	max	41.86	2.13	447.69	25.61	25.34	46.28	1.19	2.32	48.19
F17	avg	1.66e3	735.54	0.1	1.37e3	1061.2	3.60e3	10.45	0.26	7.33e3
	std	2.00e3	596.39	0.12	1.49e3	893.3	2.08e3	10.41	0.19	6.89e3
	min	57.94	30.17	0	38.28	141.05	360.6	0.01	0	442.13
	max	6731.7	2.21e3	0.41	5.85e3	2712.5	7.53e3	24.04	0.49	2.39e4
F18	avg	28.65	259.85	0.93	7.49	77.48	153.82	0.3	0.01	70.91
	std	66.31	510.74	1.83	9.16	236	233.21	0.7	0.01	130.47
	min	1.28	2.8	0	1.27	2.23	7.11	0	0	3.55
	max	295.91	2308.1	6.71	43.03	1.08e3	950.55	2.5	0.02	492.56
F19	avg	15.79	0.26	0	1.08	4.38	24.52	0.11	0.32	114.22
	std	9.99	0.18	0	1.25	7.35	8.21	0.15	0.3	46.74
	min	1.31	0	0	0	0	0.99	0	0	1.99
	max	27.96	0.62	0	5.29	21.67	36.52	0.31	1.31	149.65
F20	avg	147.87	146.05	153.44	149.25	142.07	132.34	171.37	154.37	178.04
	std	53.29	40.32	54.84	55.92	52.74	50.72	49.78	55.82	45.86
	min	100	100	100	100	100	100	100	100	100
	max	206.48	204.36	210.89	218.49	207.35	214.78	209.68	213.06	206.41
F21	avg	100.45	99.06	95.32	77.96	96.59	92.57	96.16	73.07	100.46
	std	0.44	3.33	22.44	33.73	17.37	27.51	18.91	43.67	0.35
	min	100	88.42	0	11.56	22.79	0	15.83	0	100
	max	101.73	100.7	100.98	102.94	101.07	102.11	101.03	101.68	101.1
F22	avg	307.16	302.46	304.32	308.26	305.05	309.29	305.89	307.78	306.2
	std	2.13	1.91	2.19	2.17	1.88	3.88	1.97	2.82	2.13
	min	303.88	300	300	305.08	300	304.83	300	304.34	303.83
	max	313.12	305.01	307.31	312.58	308.62	319.86	309.39	315.98	311.54
F23	avg	193.28	276.73	300.11	247.65	285.16	191.63	322.48	327.3	238.48
	std	117.21	77.06	86.32	117.51	95	107.27	52.42	53.66	116.02

Function	Statistic	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
	min	100	147.92	100	13.23	100	100	100	100	100
	max	337.43	333.8	341.7	347.86	335.42	332.23	337.16	347.12	336.19
F24	avg	407.52	415.59	397.9	401.53	410.01	413.93	423.63	398.05	404.74
	std	18.48	20.71	0.15	10.14	20.4	22.2	23.67	0.11	16.66
	min	397.74	397.74	397.74	397.76	397.74	397.74	397.74	397.74	397.74
	max	443.73	446.13	398.09	444.36	445.94	443.43	445.83	398.21	443.42
F25	avg	275	300	300	229.38	300	290.78	300	300	275
	std	44.43	0	0	120.65	0	28.5	0	0	44.43
	min	200	300	300	0.02	300	200	300	300	200
	max	300	300	300	300	300	300	300	300	300
F26	avg	384.85	390.97	388.51	393.37	393.71	390.28	389.53	390.36	392.34
	std	6.05	2.93	0.89	3.31	1.65	0.91	0.17	2.07	3.23
	min	374.21	386.93	386.89	387.3	388.87	389.52	389.01	387.32	389.64
	max	390.26	395.07	389.52	398.68	395.01	393.06	389.95	394.23	398.57
F27	avg	310.08	357.24	300	314.19	300	314.19	413.52	300	493.1
	std	122.51	82.52	0	63.45	0	63.44	148.6	0	135.35
	min	0	300	300	300	300	300	300	300	300
	max	488.83	583.73	300	583.75	300	583.73	611.82	300	611.89
F28	avg	255.56	248.72	390.01	254.61	248.79	264.05	244.43	234.12	250.51
	std	9.49	8.35	184.93	10.48	7.45	13.42	9.08	3.39	9.88
	min	238.98	235.68	230.09	237.4	237.57	244.16	234.67	229.11	238.03
	max	269.91	268.39	914.54	276.93	269.86	287.49	267.36	240.32	274.2
F29	avg	5.91e3	5.16e4	397.89	2.91e3	4.06e3	4.28e3	1.23e5	438.6	2.31e3
	std	4.79e3	2.00e5	4.72	3.06e3	2.69e3	2.71e3	2.99e5	67.75	936.36
	min	1.31e3	1.51e3	394.87	777.46	759.98	1.96e3	394.5	394.58	811.85
	max	1.74e4	8.99e5	410.8	1.34e4	1.13e4	1.21e4	8.18e5	602	4.12e3

Table 9: Statistical performance results on CEC17 100D benchmark functions.

Function	Stat.	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
F1	avg	4.04e3	3.20e3	1.40e3	4.85e3	8.25e3	8.98e3	0	6.28e3	5.17e3
	std	3.18e3	3.99e3	2.47e3	4.91e3	8.04e3	1.35e4	0	6.80e3	5.79e3
	min	85.81	7.23	1.62	12.3	1.64e3	13.07	0	17.38	3.47
	max	1.08e4	1.23e4	1.04e4	1.76e4	2.33e4	5.47e4	0	2.51e4	1.94e4
F2	avg	1.70e4	5.70e4	0.94	6.07e4	1.17e5	7.59e3	1.32e5	0.08	1.85e4
	std	4.70e3	9.59e3	1.15	1.62e4	1.89e4	2.93e3	1.07e5	0.11	4.14e3
	min	1.20e4	4.08e4	0.04	3.18e4	9.43e4	4.27e3	0	0	1.06e4
	max	2.59e4	7.64e4	3.82	9.26e4	1.67e5	1.45e4	4.89e5	0.47	2.68e4
F3	avg	208.47	216.85	151.23	207.2	245.9	225.38	130.57	248.06	235.09
	std	38.69	23.79	43.37	61.69	37.38	12.18	47.18	62.92	11.65
	min	138.28	197.36	72.44	78.02	193.06	197.36	53.97	79.56	224.36
	max	248.9	281.91	214.3	305.89	310.23	254.06	226.35	358.19	273.02
F4	avg	101.09	164.27	197	321.25	152.73	200.27	222.59	272.17	66.67
	std	18.1	34.05	38.23	59.3	19.42	49.51	23.61	41.23	7.88
	min	65.67	98.71	128.35	208.94	115.42	100.18	178.11	213.92	56.71
	max	145.26	218.99	257.69	455.69	182.09	303.46	275.61	387.04	87.56
F5	avg	0.31	0	0	0	0.06	0.88	1.57	15.22	0
	std	0.19	0	0	0	0.02	0.41	1.18	2.88	0
	min	0.09	0	0	0	0.03	0.35	0.16	9.29	0
	max	0.9	0	0	0	0.14	1.98	4.35	19.38	0.02
F6	avg	211.83	326.99	315.89	493.38	225.77	371.09	416.6	786.86	181.8
	std	16.23	33.51	44.02	86.56	17.81	37.96	40.09	109.85	16.14
	min	189.43	273.03	259.45	356.71	192.86	295.35	343.4	586.79	150.79
	max	248.4	404.36	411.42	674.9	259.21	468.1	504.16	953.95	210.01
F7	avg	97.66	164.29	191.88	379.99	164.26	187.53	232.06	286.95	67.71
	std	16.98	30.98	28.08	71.07	20.33	35.79	33.58	36.76	9.97
	min	69.65	121.53	145.26	278.59	124.37	128.78	157.21	243.76	50.74
	max	140.29	257.35	258.69	564.32	198.73	252.72	290.54	397.98	85.57
F8	avg	99.39	3.42	165.56	2.50e3	1.37	84.94	1.88e3	4.71e3	0.84
	std	35.95	1.78	165.67	1.58e3	0.96	35.04	744.72	803.51	0.67
	min	48.62	0.27	8.99	461.86	0.11	35.53	708.18	2.70e3	0
	max	151.04	7.26	650.37	6.77e3	3.36	157.04	3.50e3	5828.2	2.18
F9	avg	1.08e4	1.00e4	1.17e4	1.21e4	1.08e4	1.14e4	1.00e4	1.33e4	6.49e3
	std	1.14e3	707.58	1.28e3	1628.1	881.2	1.06e3	487.39	1.09e3	1.30e3
	min	9.19e3	8723.5	9.56e3	9.12e3	9.08e3	9.10e3	9.21e3	1.10e4	4.74e3

Table 9: Statistical performance results on CEC17 100D benchmark functions (continued).

Function	Stat.	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
	max	1.27e4	1.13e4	1.47e4	1.53e4	1.25e4	1.33e4	1.10e4	1.59e4	9.09e3
F10	avg	798.52	340.76	163.18	700.59	761.34	1015.6	1199.4	691.22	557.47
	std	122.56	64.73	47.51	209.59	95.26	131.38	283.81	155.65	102.89
	min	638.59	227.77	81.53	315.76	607.53	814.3	643.62	433.56	401.25
	max	1011.3	512.58	249.53	1140.8	935.41	1.23e3	1.71e3	993.94	763.78
F11	avg	9.59e5	6.15e6	2.03e5	6.03e5	1.30e7	6.65e6	7.83e4	8.77e5	6.65e5
	std	4.68e5	3.18e6	8.78e4	4.71e5	5.12e6	3.90e6	7.85e4	3.32e5	2.67e5
	min	3.19e5	2.05e6	7.82e4	1.93e5	3.00e6	1.37e6	1.06e4	3.92e5	2.46e5
	max	1.96e6	1.47e7	3.60e5	2.41e6	2.37e7	2.01e7	2.69e5	1.81e6	1.24e6
F12	avg	4.80e3	2.64e3	2.04e3	2.12e3	9.52e3	1.89e4	5.30e3	3.69e3	1.73e4
	std	2.38e3	2.10e3	1.75e3	1.86e3	3.26e3	1.09e4	6.19e3	3.14e3	1.54e4
	min	1.83e3	70.99	156.52	150.65	4.15e3	6390.4	588.4	659.56	202.49
	max	1.05e4	7.06e3	6.14e3	5.82e3	1.71e4	4.25e4	2.34e4	1.49e4	3.54e4
F13	avg	1.19e5	9.49e5	9.25e3	2.54e5	4.26e5	1.32e5	4513.7	2.25e4	6.69e4
	std	3.75e4	3.77e5	7.66e3	1.14e5	2.35e5	7.82e4	1.56e4	1.32e4	2.24e4
	min	5.91e4	3.34e5	1.84e3	1.10e5	1.08e5	4.44e4	406.64	3.45e3	2.28e4
	max	1.78e5	1.73e6	3.10e4	5.09e5	9.13e5	3.22e5	7.06e4	4.74e4	1.23e5
F14	avg	674.1	1.42e3	2.17e3	871.2	3.23e3	5.83e3	419.06	5.20e3	6874.8
	std	507.3	1.10e3	2.24e3	445.42	2.71e3	6.83e3	84.04	5.79e3	1.05e4
	min	269.83	139.28	136.19	307.57	543.37	1.15e3	263.66	208.96	226.92
	max	2.11e3	4.26e3	6.32e3	2.01e3	1.03e4	2.70e4	582.33	2.19e4	2.73e4
F15	avg	2.15e3	2.30e3	3.08e3	3.16e3	2.34e3	2.16e3	2.77e3	2.98e3	929.58
	std	610.92	457.17	499.63	376.26	598.06	602.97	308.95	544.86	458.46
	min	779.9	1.27e3	2.23e3	2.37e3	953.91	1.31e3	2.05e3	2.26e3	216.22
	max	3.02e3	2.94e3	3.88e3	3.76e3	3.34e3	3.32e3	3.27e3	4.43e3	1.80e3
F16	avg	1881.3	1.62e3	1.75e3	2.53e3	1.75e3	1.63e3	2.19e3	2.34e3	845.37
	std	434.84	214.26	331.62	477.65	299.83	372.03	285.78	365.96	332.95
	min	1.20e3	1.17e3	1.04e3	1.37e3	1.36e3	1.08e3	1.50e3	1.65e3	369.71
	max	2.81e3	1.93e3	2.17e3	3.18e3	2.55e3	2.54e3	2575.7	2.92e3	1.46e3
F17	avg	3.28e5	1.07e6	5.19e4	4.53e5	8.37e5	2.38e5	1.00e5	3.74e4	1.69e5
	std	8.74e4	5.43e5	2.23e4	2.46e5	2.69e5	1.02e5	6.58e4	1.76e4	8.28e4
	min	1.89e5	4.14e5	1.98e4	1.64e5	3.05e5	1.02e5	669.21	1.28e4	8.12e4
	max	5.08e5	2.57e6	8.73e4	9.56e5	1.55e6	4.76e5	2.34e5	7.18e4	3.37e5
F18	avg	969.08	758.82	2.32e3	668.24	2.07e3	8.77e3	2.51e3	2.10e3	2.54e4
	std	831.42	1.37e3	2.99e3	521.12	2.55e3	1.04e4	4.38e3	3.06e3	1.78e4
	min	206.55	56.35	95.91	252.81	344.29	1.06e3	183.51	164.76	282.79
	max	2.85e3	5911.7	1.21e4	2.42e3	9.87e3	3.99e4	1.69e4	1.35e4	3.95e4
F19	avg	1.68e3	1.59e3	1.73e3	2.66e3	1.66e3	2.05e3	1.96e3	2.50e3	832.48
	std	155.25	247.85	419.33	312.69	224.28	365.94	276.99	503.37	254.28
	min	1.31e3	1.18e3	982.51	1.87e3	1.32e3	1.50e3	1410.1	1.58e3	506.62
	max	1.92e3	2.05e3	2.64e3	3.16e3	2.17e3	2849.4	2.61e3	3.46e3	1.37e3
F20	avg	331.82	399.26	408.9	587.98	390.66	426.23	462.34	508.77	300.79
	std	18.95	19.41	28.74	44.88	18.96	31.28	33.46	31.76	11.72
	min	297.69	364.79	335.9	515.9	358.95	346.49	397.29	454.43	281.62
	max	361.89	429.29	456.07	679.46	430.8	496.12	522.15	573.01	325.7
F21	avg	101.29	1.14e4	1.27e4	1.30e4	9.93e3	1.09e4	1.13e4	1.40e4	106.39
	std	2.11	754.62	1.41e3	1.68e3	5.10e3	4.85e3	710.84	1.61e3	28.58
	min	100	1.02e4	9.85e3	9.26e3	100.19	100	9.84e3	1.12e4	100
	max	105.93	1.26e4	1.55e4	1.61e4	1.38e4	1.52e4	1.24e4	1.72e4	227.81
F22	avg	662.06	641.56	680.56	852.72	772.39	690.42	716.88	849.5	627.57
	std	18.18	22.39	33.17	36.8	45.01	27.22	36.26	33.54	21.74
	min	632.49	598.99	609.08	799.91	715.16	631.88	665.4	777.59	597.68
	max	700.65	675.05	735.46	934.69	863.15	753.93	780.15	913.06	669.36
F23	avg	995.96	1.05e3	1.05e3	1.39e3	1.05e3	1.05e3	1.16e3	1.30e3	963.51
	std	24.8	20.97	37.87	54.25	21.2	37.96	43.46	71.51	19.44
	min	951.9	1.01e3	1.00e3	1.28e3	1.02e3	995.29	1.09e3	1.16e3	927.27
	max	1.04e3	1.09e3	1.14e3	1.51e3	1.09e3	1.11e3	1.22e3	1.40e3	999.24
F24	avg	728.29	780.09	749.6	791.06	789.26	713.66	755.2	801.41	744.89
	std	50.85	48.05	42.07	66.21	45.34	61.65	80.66	70.12	46.19
	min	660.82	639.99	659.01	676.14	692.41	601.73	590.14	637.69	647.25
	max	850.9	830.84	822.93	899.63	849.28	805.54	880.97	950.96	819.98
F25	avg	1.37e3	4850.3	4.79e3	7.84e3	4.64e3	5128.4	5.96e3	7.59e3	2.74e3
	std	1.67e3	308.19	300.98	674.98	318.25	416.96	514.39	691.35	1.46e3
	min	300	4.06e3	4.28e3	6.69e3	4.07e3	4.60e3	5.28e3	6.21e3	582.18
	max	4.10e3	5.21e3	5.36e3	8973.9	5.05e3	5.97e3	7.54e3	8.76e3	3.99e3
F26	avg	646.05	654.93	654.33	855.04	700.72	632.74	764.39	992.28	588.91
	std	31.03	22	28.2	46.41	46.46	18.9	60.6	118.87	21.09
	min	581.28	618.85	607.24	744.23	646.1	593.11	660.64	701.8	554.5
	max	701.15	692.74	702.97	943.93	804.1	677.95	901.52	1.19e3	638.85
F27	avg	543.26	571.14	547.07	554.9	585.84	535.32	541.13	578.8	562.57
	std	17.37	31.41	36.84	26.21	34.21	14.49	38.85	31.35	21.04
	min	518.76	526.64	486.68	497.04	529.91	508.38	479.81	548.56	533.85
	max	576.95	665.04	636.17	609	665.84	565.45	636.17	672.64	612.58

Table 9: Statistical performance results on CEC17 100D benchmark functions (continued).

Function	Stat.	E-AHPSO	BFLPSO	CoDE	EPSO	HCL-DMS-PSO	HIDMS-PSO	L-SHADE	MTDE	TLSPSO
F28	avg	1.94e3	1.88e3	2.04e3	3021.4	2.09e3	2.40e3	2.47e3	3.14e3	1.52e3
	std	358.94	298.46	388.09	409.14	367.91	498.95	370.31	467.86	306.26
	min	1.32e3	1.35e3	1.28e3	2.27e3	1417.9	1.50e3	1.62e3	2.16e3	1.11e3
	max	3028.1	2.27e3	2.54e3	3.64e3	2.84e3	3.29e3	3.06e3	3.80e3	2.19e3
F29	avg	2.67e3	7.92e3	3.02e3	6.15e3	5.40e4	3.74e4	3.10e3	7.14e3	1.21e4
	std	2.19e3	2.94e3	966.9	2.37e3	1.91e4	2.08e4	454.56	3.07e3	5.67e3
	min	778.08	3.36e3	2.38e3	2797.1	2.94e4	1.60e4	2348.5	3.70e3	3.30e3
	max	9.07e3	1.32e4	6.17e3	1.14e4	9.69e4	8.15e4	4.31e3	1.40e4	1.81e4

6. Analysis of E-AHPSO

In this section, we observe and compare the convergence behaviour, pattern of population diversity depletion, consistency of the discovered solution and time complexity of the proposed algorithm against the top-performing comparator algorithms and the constituent algorithms adopted in the proposed ensemble algorithm (AHPSO, MaPSO, HPSO-TVAC). In addition, a substitution experiment was conducted in which AHPSO was replaced with several comparable algorithms to identify, analyse, and compare its contribution in the ensemble.

6.1. Convergence Analysis

Figures 16-17 display the convergence rate comparison of the proposed algorithm (E-AHPSO) against L-SHADE, HIDMS-PSO, TLSPSO, AHPSO, EPSO and HCL-DMS-PSO on selected 100-dimensional multimodal, hybrid, and composition CEC'14 and CEC'17 problems. E-AHPSO consistently converges faster than, or is on par with, these high-performing comparative algorithms. The 18 problems selected are the most complex instances of 100-dimensional problems in the CEC'14 and CEC'17 test suites.

It should be noted that a significant part of the convergence for E-AHPSO ($50\% - 70\%$) occurs within $750 - 1050$ iterations executed by AHPSO, while we mainly observe fine-tuning during the remainder of the search carried out by the other ensemble constituent algorithms. This consistent convergence pattern also reflects the significant role undertaken by AHPSO in both the algorithm's solution discovery and converging behaviours. Interested readers can visit the supplementary material to observe the detailed convergence rate figures provided for each problem and dimension of the CEC'14 and CEC'17 problems.

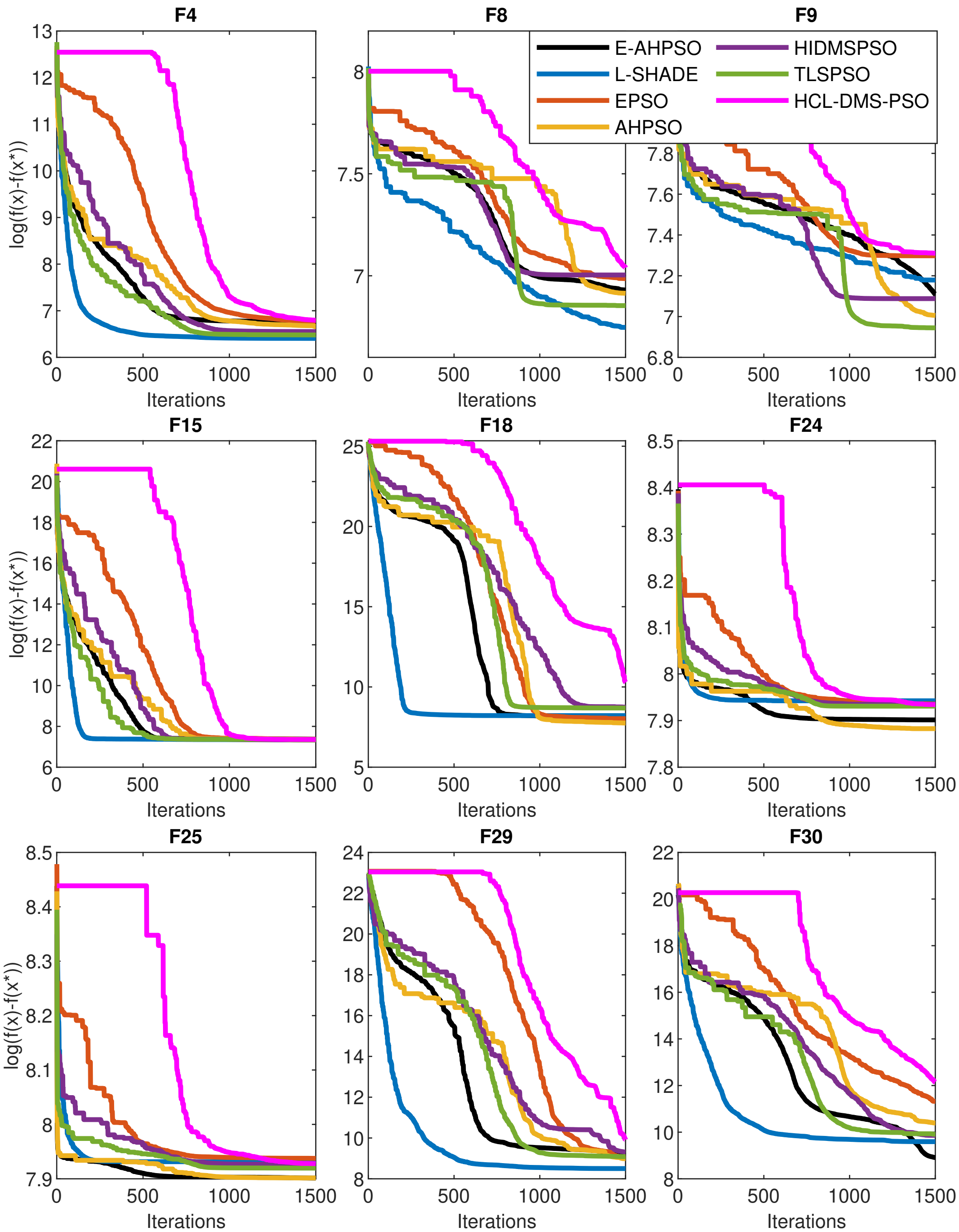

Figure 16: Convergence rate comparison of E-AHPSO on complex 100-dimensional multimodal, hybrid and composition CEC'14 problems.

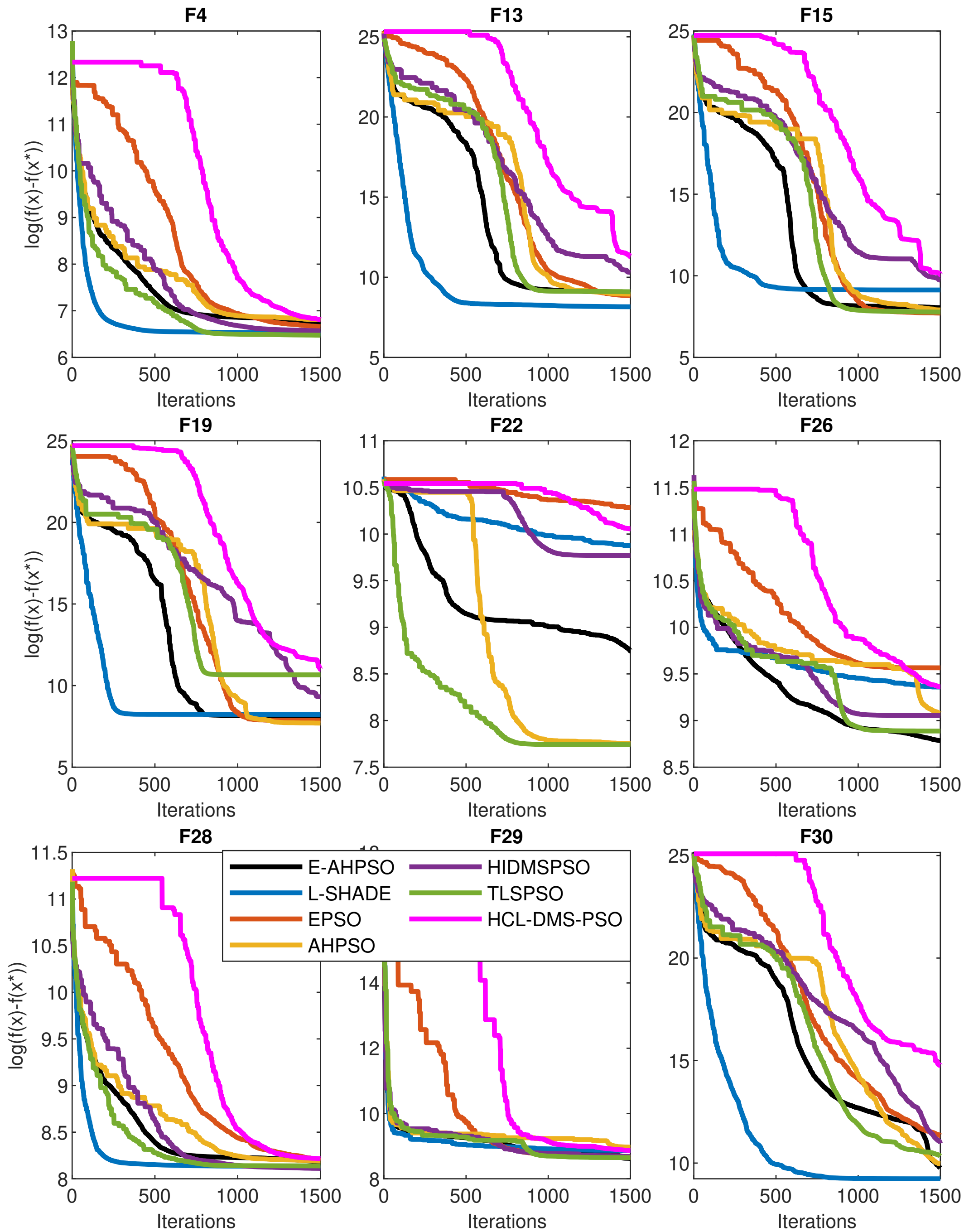

Figure 17: Convergence rate comparison of E-AHPSO on complex 100-dimensional multimodal, hybrid and composition CEC'17 problems.

6.2. Population Diversity Analysis

Figures 18-19 display the population diversity pattern recorded simultaneously with the convergence pattern on the 100-dimensional CEC'14 and CEC'17 problems. The population diversity was calculated based on the instructions provided in the study [94]. As with convergence, we compared the population diversity of E-AHPSO against that of L-SHADE, EPSO, AHPSO, HIDMS-PSO, TLSPSO, and HCL-DMS-PSO.

Overall, the pattern observed in the depletion of population diversity approximately correlates with the convergence pattern of the algorithms (see Figures 16-17). We consistently observe that HCL-DMS-PSO initiates and maintains a more diverse population that revamps throughout the search process. However, among multiple strategies adopted for population topology and velocity update, HCL-DMS-PSO also utilises a dual mutation operator to re-diversify the population, which increases the time complexity and prevents faster convergence to a better solution in higher-dimensional problems (see Figures 16-17).

On the other hand, E-AHPSO does not substantially maintain population diversity throughout the search. However, in this case, we do not observe any correlation between maintaining better population diversity and the overall performance of the algorithms. AHPSO is good at maintaining diversity, but the other constituent algorithms in E-AHPSO are not. The diversity maintained by AHPSO in the initial phase of the ensemble's search seems to be sufficient to enable the other, exploitative, algorithms in the ensemble to converge to very high-quality solutions when they take over the search in the later phases.

6.3. Box-Plot Analysis

In this section, we assess the consistency of solutions obtained by E-AHPSO and the best performing comparator benchmark algorithms, CoDE, L-SHADE, TLSPSO, BFLPSO, EPSO, HCLDMS-PSO, MTDE, and the ensemble constituent algorithms, AHPSO, HPSO-TVAC and MaPSO, over 50 consecutive runs on each 50-dimensional unimodal (F_1), multimodal (F_3), hybrid ($F_{10} - F_{19}$) and composition ($F_{20} - F_{30}$) problem selected from the CEC'17 test suite.

E-AHPSO has good consistency across all problem types, with a relatively tight distribution of error values in all cases. It is difficult to highlight a single most consistent algorithm from Figure 20; however, it can be observed that

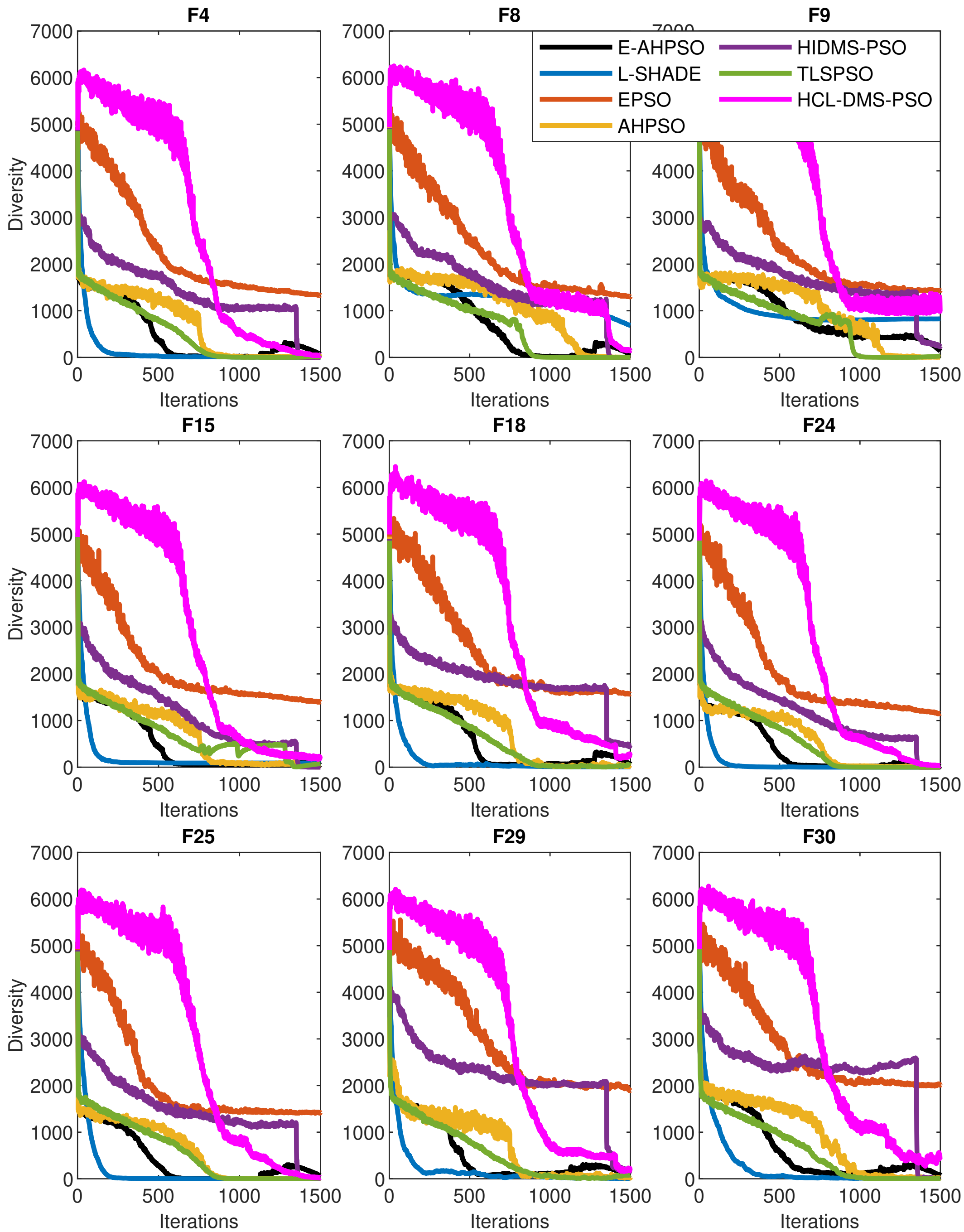

Figure 18: Population diversity comparison of E-AHPSO and the top ranking benchmark algorithms on selected complex 100-dimensional CEC'14 problems.

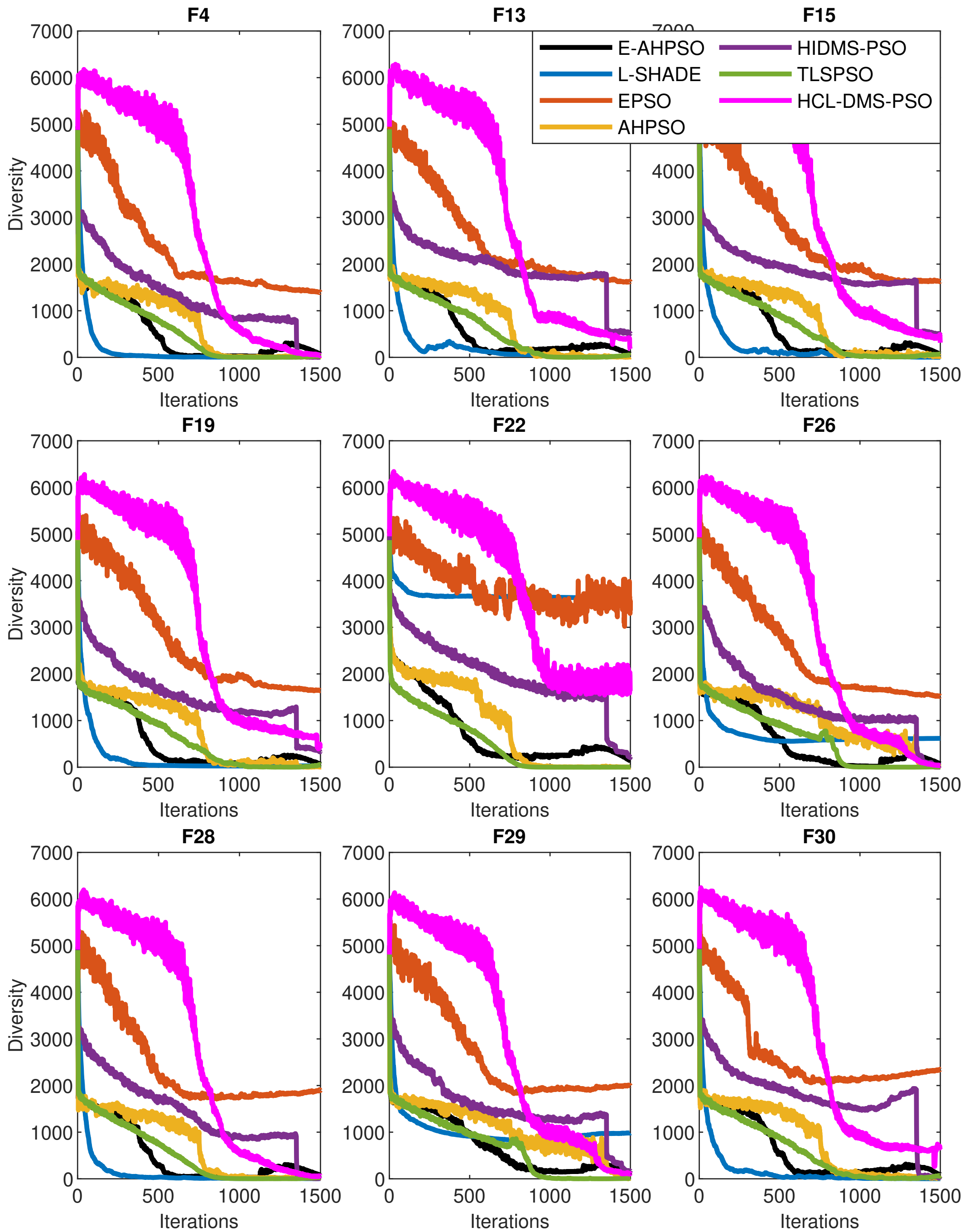

Figure 19: Population diversity comparison of E-AHPSO and the top ranking benchmark algorithms on selected complex 100-dimensional CEC'17 problems.

E-AHPSO exhibited consistency comparable to the other top performing algorithms.

6.4. Time Complexity Analysis

The time complexity comparison for the proposed algorithm and the 39 benchmark algorithms at 10, 30, 50 and 100 dimensions is shown in Figure 21. The time complexity value TC is calculated based on the instructions provided in study [102], using MATLAB R2023b on a machine with an AMD Ryzen 9 3900XT 12-core processor.

Overall, E-AHPSO's time complexity was observed to be highly competitive. It was more efficient than 58%, 64%, 65% and 68% of the 39 benchmark algorithms for 10, 30, 50 and 100-dimensional problems, respectively. E-AHPSO's time complexity can be seen to be on a par with L-SHADE and BFLPSO on 10-dimensional, HCL-DMS-PSO and BFLPSO on 30-dimensional, MaPSO and BFLPSO on 50-dimensional, and BBPSO and FFQ-HIDMS-PSO on 100-dimensional problems. As expected, E-AHPSO's time complexity is similar (slightly more efficient) compared to AHPSO. The time complexity data for all benchmark algorithms is provided in the supplementary material for interested readers.

6.5. Substitution Experiment

In this work, we highlighted the strengths and limitations of the AHPSO algorithm and theorised that it would be an effective explorative mechanism if complemented with a set of simpler exploitative PSO variants.

However, it might be argued that the overall effectiveness of the proposed ensemble algorithms may be due to the complex dynamics of the constituent algorithms or other factors, and that the utilisation of an arbitrary explorative mechanism instead of AHPSO might yield comparable or better results. This is a well-founded question and worth further investigation. Hence, we investigated the performance of various *alternative* algorithms substituted for AHPSO in the ensemble model used in the main experiments detailed earlier. The individual and collaborative performances were compared to observe any changes in the performance of the substitute algorithms when utilised as part of the ensemble.

The experiment detailed in section 5.1 was replicated on the 100-dimensional CEC'14 problems using the $E_1^{O_6}$, $E_4^{O_4}$, $E_8^{O_6}$, $E_{12}^{O_6}$, $E_{14}^{O_6}$, $E_{21}^{O_5}$ and $E_{27}^{O_6}$ ensembles that were highlighted in section 4.4, where AHPSO was substituted with several comparable algorithms, namely, HCLPSO, DMS-PSO, HIDMS-PSO,

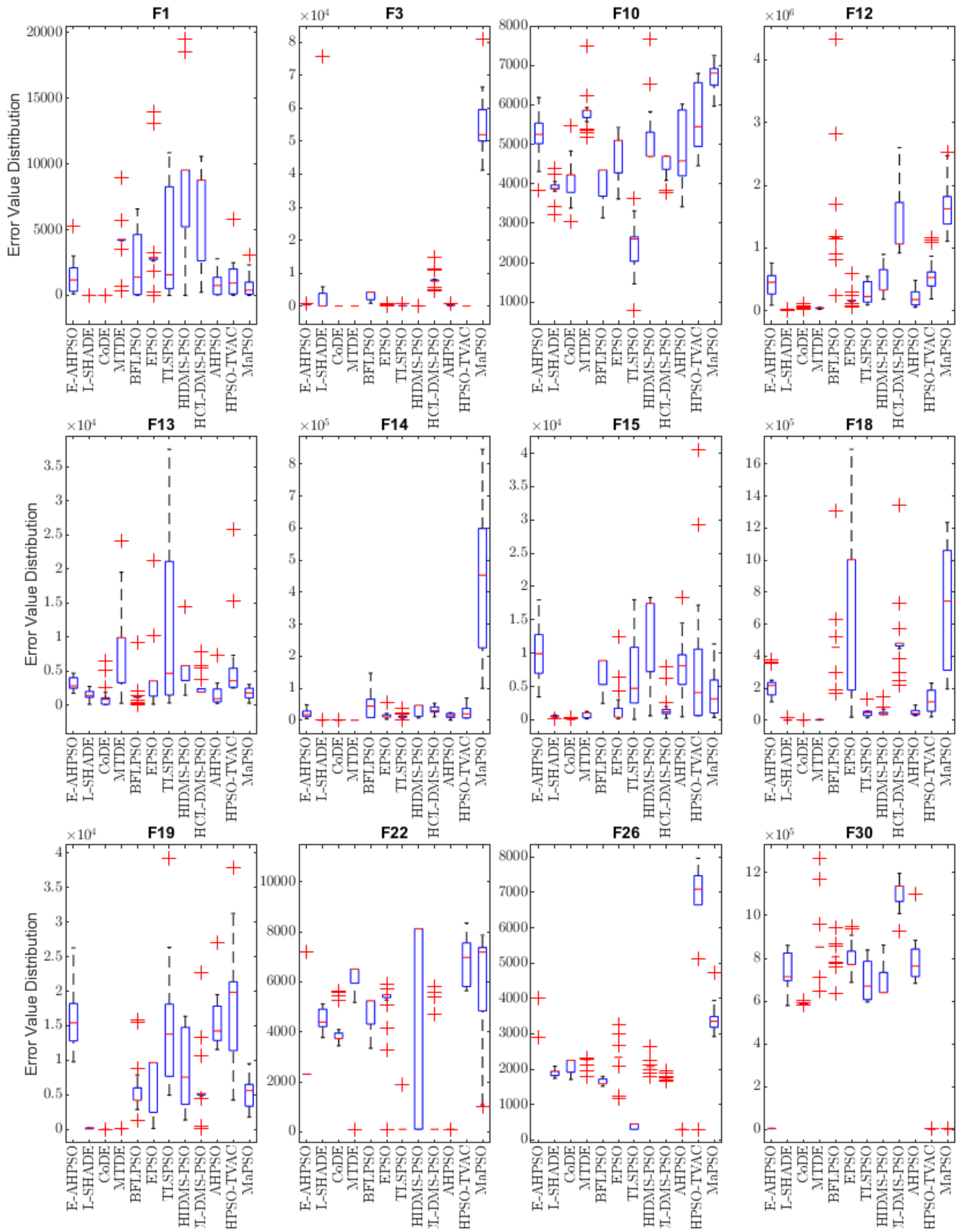

Figure 20: Distribution of error values on selected 50-dimensional CEC'17 problems.

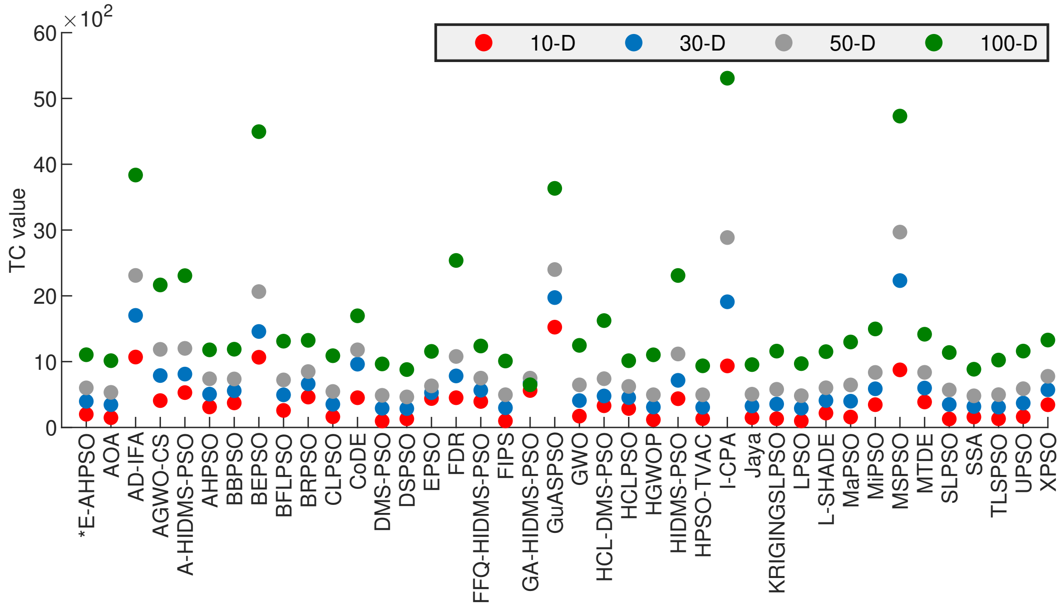

Figure 21: Time complexity comparison for 10, 30, 50 and 100-dimensional problems.

FFQ-HIDMSPSO, GA-HIDMS-PSO and A-HIDMS-PSO. The substitute algorithms were selected based on their behavioural heterogeneity, explorative capabilities and their overall comparable performance to AHPSO. Figure 22 presents a summary of the statistical results at $\alpha = 0.05$ for the 7 ensemble algorithms with substitute algorithms instead of AHPSO, along with the performance of the substitute algorithms on their own. The ensembles are denoted by the substitute algorithm's name followed by the ensemble key and ordering, e.g. DMS-PSO $^{O_6}_{1}$ which executes in the following sequence: DMS-PSO → HPSO-TVAC → BBPSO (see Table 1).

As shown in Figure 22, overall, we do not observe an appreciable difference between the individual performances of DMSPSO, HCLPSO and A-HIDMS-PSO, and the performances of the ensembles created when these algorithms are substituted for AHPSO. However, ensembles with GA-HIDMS-PSO exhibit a slight performance deterioration compared to the individual performance of GA-HIDMS-PSO, and, in contrast, ensembles with FFQ-HIDMS-PSO perform noticeably better than the individual performance of FFQ-HIDMS-PSO, especially in the case of the $E_1^{O_6}$ and $E_{21}^{O_5}$ ensembles.

In addition, the results indicate that FFQ-HIDMS-PSO is the most responsive substitute algorithm to the ensemble configuration among the 6

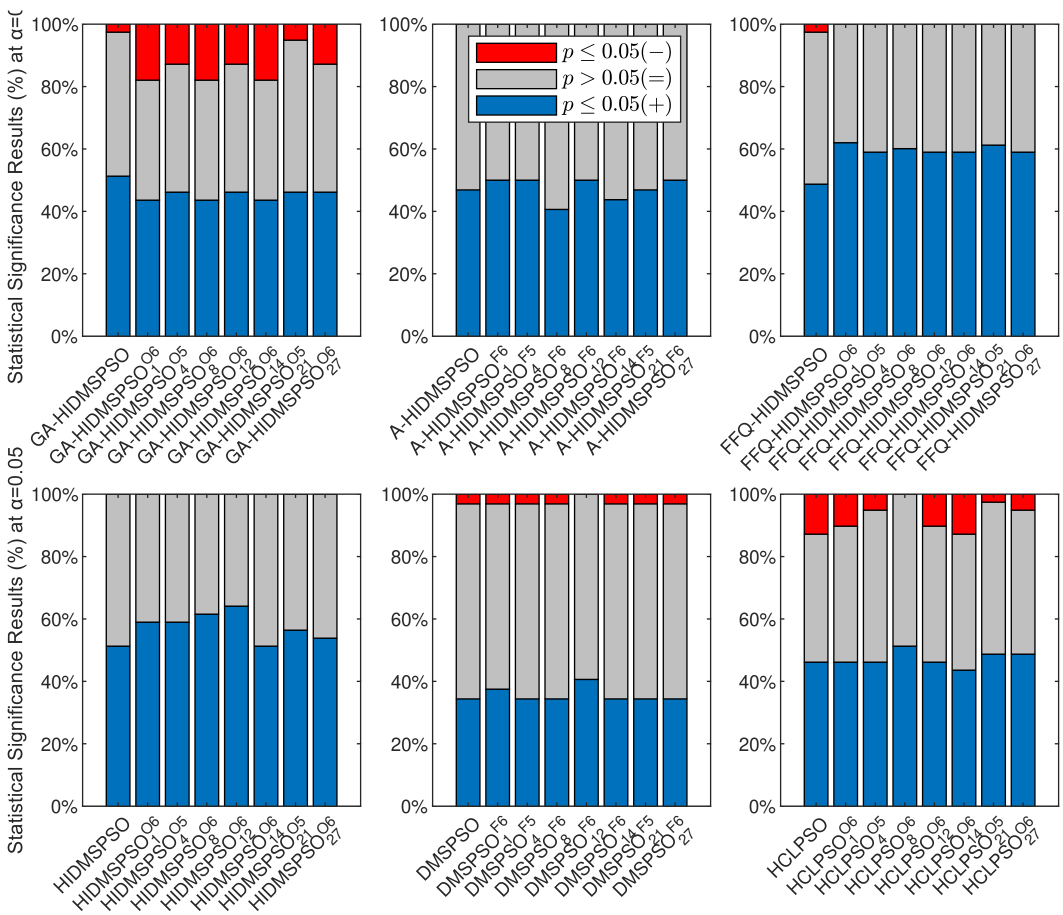

Figure 22: Summary of the Wilcoxon signed-rank test results for ensemble configurations using alternative exploratory mechanisms in place of AHPSO on the 100-dimensional CEC'14 problems. The symbols +, =, and − indicate statistically significantly better performance, no statistically significant difference, and statistically significantly worse performance, respectively, at $\alpha = 0.05$.

algorithms tested. Importantly, overall, none of the substitute algorithms yielded a performance comparable to our proposed ensemble algorithm, E-AHPSO. The argument as to *"why"* some algorithms are improved, or weakened, or are simply unresponsive within an ensemble, requires a specific and thorough assessment that is beyond the scope of this paper. However, in addition to various dynamics, we hypothesize that effective exploration capability is a major contributing factor that is essential in the discovery of promising regions in the early phase of the search, something that AHPSO is particularly good at. In relation to that, as shown in section 4.3, the results strongly favoured particular orderings in which AHPSO was executed as the first algorithm in the ensemble configuration. Further investigations, experiments and analysis verified this finding and we theorised that the primary underlying factor of superior performance attained when AHPSO is executed as the initial algorithm was closely related to its heterogeneous search dynamics that allow AHPSO to discover the promising regions during the initial phase of the search.

6.6. Contour Plot Analysis

The exploratory behaviour of AHPSO was previously studied in [15],[3]. Here, we utilise contour plots to discern the exploration capability and search space coverage of the constituent algorithms (AHPSO, HPSO-TVAC and MaPSO) of the proposed E-AHPSO algorithm.

Figures 23,24,25,26 display the contour plots for the Holder Table, McMormick, Schwefel and Griewank functions with highly complex landscapes and multiple local and global minima. The particle positions at 1^{st}, 100^{th}, 500^{th} and 1000^{th} iterations are shown for each constituent algorithm of E-AHPSO. The same initial population distribution was used for each algorithm to accurately observe the search behaviour and coverage of the search space.

In all cases shown in Figures 23,24, 25,26 AHPSO consistently exhibits much broader search space coverage. In addition, we observe that, despite converging to global optima, AHPSO resumes exploring nearby regions to find potentially better solutions.

In section 4.3, it was highlighted and documented that ensemble combinations (see Table 1) with identical constituent algorithms performed distinctly across different orderings (see Table 2), where superior performance was consistently observed with AHPSO executed as the initial algorithm in ensemble configurations.

Contour plots presented in this section complement our previous findings and verify that AHPSO can sustain superior exploratory capability throughout the optimisation process. Hence, when executed as the "non-initial" algorithm in an ensemble configuration, AHPSO is ineffective at reviving solutions inherited from the constituent algorithms. However, when executed as the "initial" algorithm, AHPSO is able to discover promising solutions through sustained exploration and subsequently convey the discovered solutions, which are further exploited by the two constituent algorithms. This is also verified by the convergence analysis shown in section 6.6, where a significant part of the convergence was observed during the phase executed by AHPSO, followed by HPSO-TVAC and MaPSO, which are reflected as subtle changes to the convergence pattern.

The "heterogeneous behaviour" observed and highlighted in this section facilitates the sustainability of AHPSO's exploratory behaviour, and it is the key component and the tenable rationale for the proposed algorithm's (E-AHPSO) superior performance, achieved through initial extensive exploration by AHPSO, followed by the exploitation of HPSO-TVAC and MaPSO.

6.7. Velocity Magnitude Analysis at Handover

Figures 27,28,29 show the velocity magnitudes of all particles in the swarm at handover (captured at the end of the final iteration of each of the ensemble's constituent algorithms), representing the swarm's state immediately before the population is handed over to the next algorithm in the ensemble's sequence. The magnitudes are displayed for different dimensions of CEC'13, CEC'14 and CEC'17 problems, and the velocity magnitude is calculated as:

$$\|\mathbf{v}_i(t)\| = \sqrt{\sum_{d=1}^{D} v_{i,d}(t)^2} \tag{26}$$

where $\|\mathbf{v}_i(t)\|$ represents the velocity magnitude of particle i at iteration t, D is the problem dimensionality, and $(v_{i,d}(t)$ is the velocity component of particle i in dimension d.

As shown in Figures 27-29, overall, AHPSO exhibits the highest velocity magnitudes across most of the 10, 30, 50, and 100-dimensional CEC'13, CEC'14 and CEC'17 functions, which indicates that at the point of handover to the subsequent algorithm, AHPSO hands over a swarm that's still strongly mobile through the search space. We also observe a relatively large interquartile range, suggesting highly diverse particle movement when AHPSO reaches

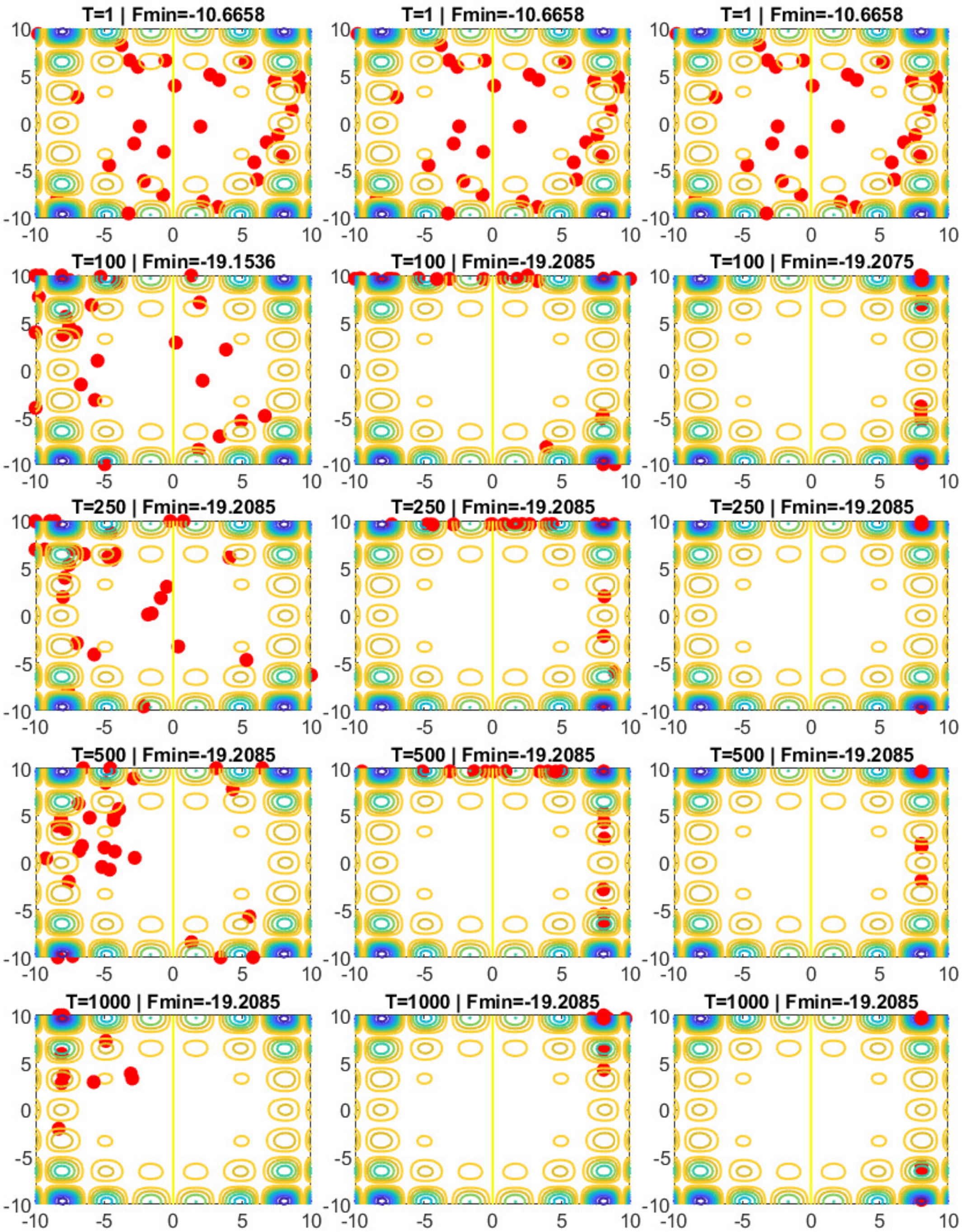

Figure 23: Search behaviour of AHPSO (left column), HPSO-TVAC (middle column), MaPSO (right column) on 2-dimensional Holder Table function.

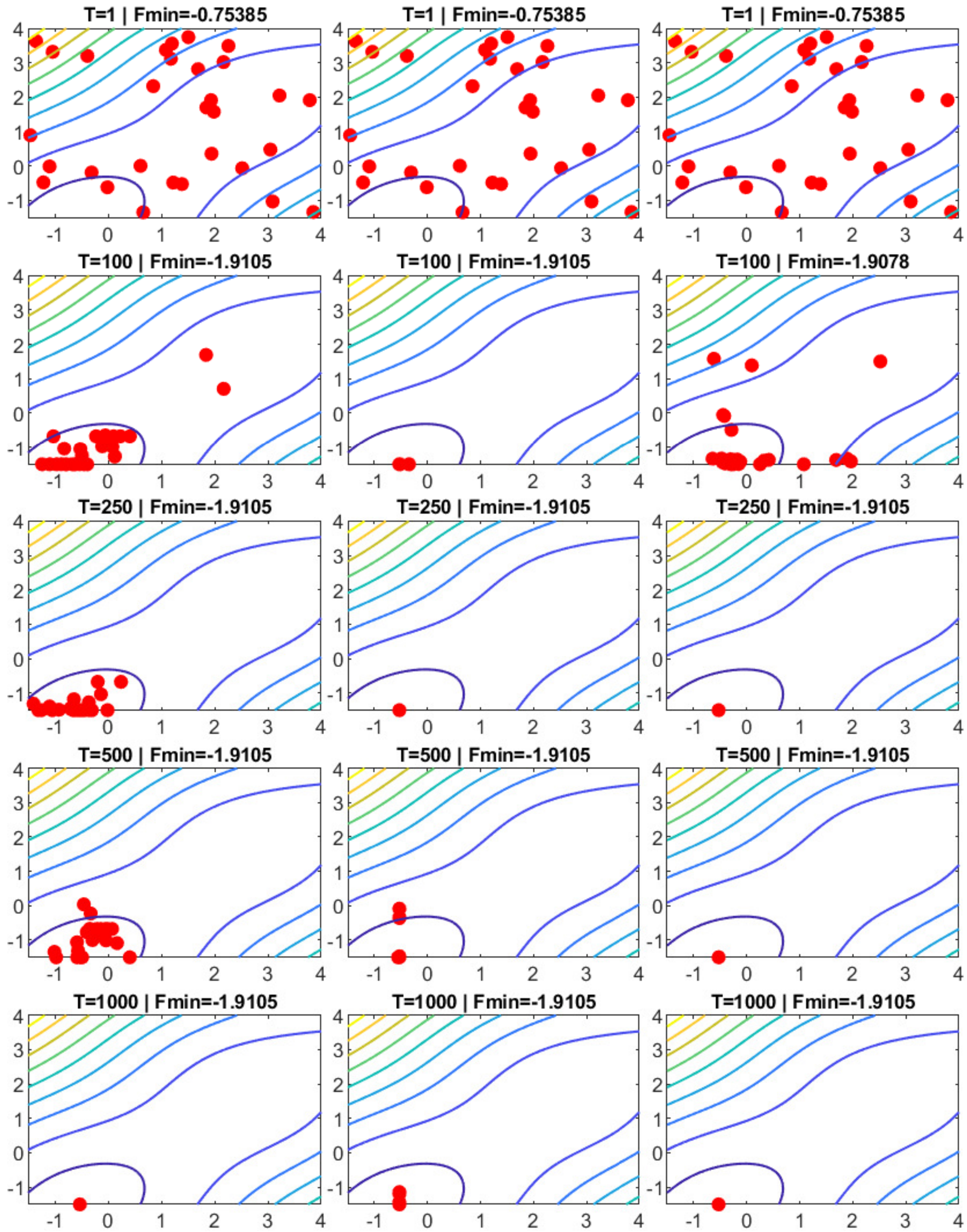

Figure 24: Search behaviour of AHPSO (left column), HPSO-TVAC (middle column), MaPSO (right column) on 2-dimensional McCormick function.

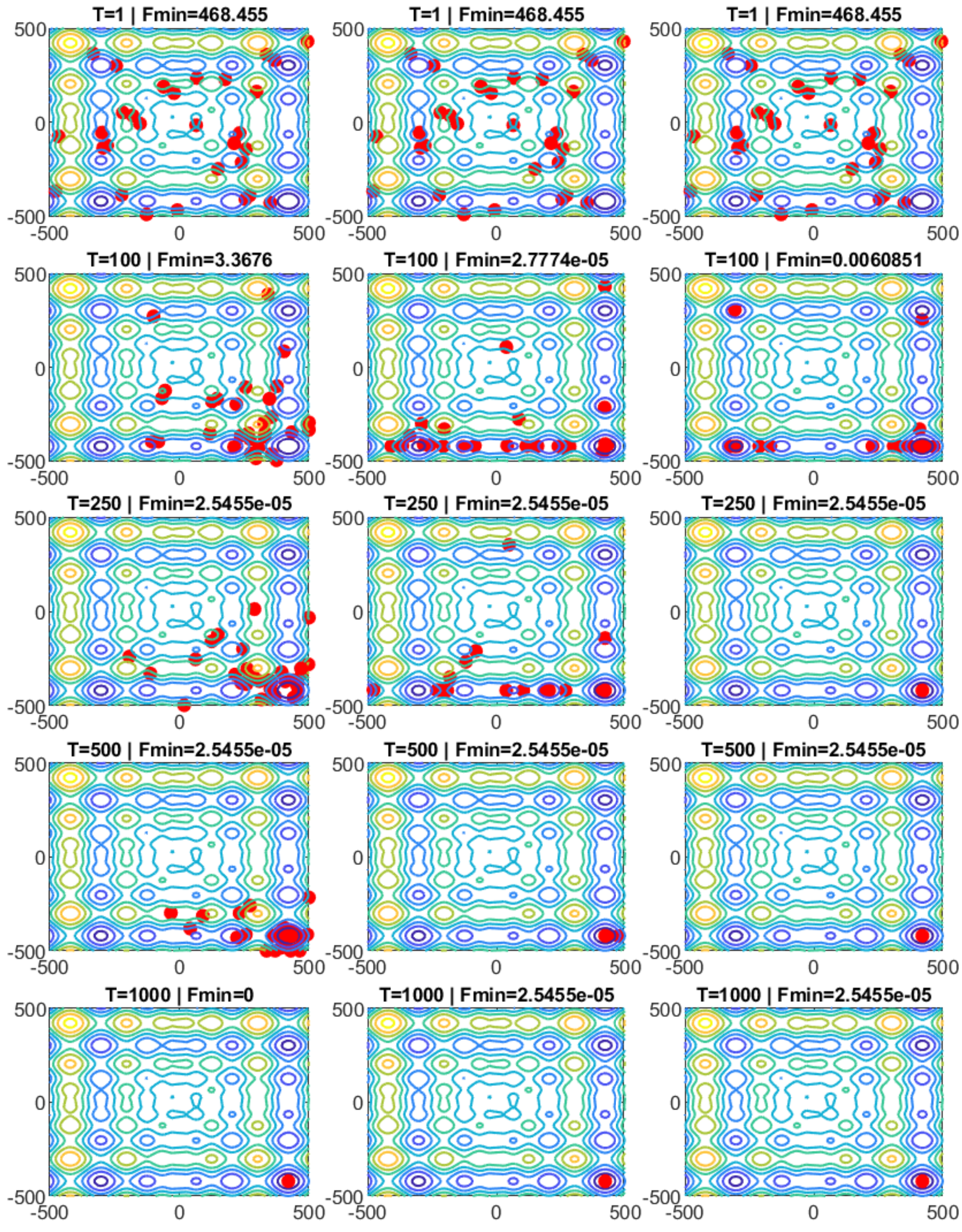

Figure 25: Search behaviour of AHPSO (left column), HPSO-TVAC (middle column), MaPSO (right column) on 2-dimensional Schwefel function.

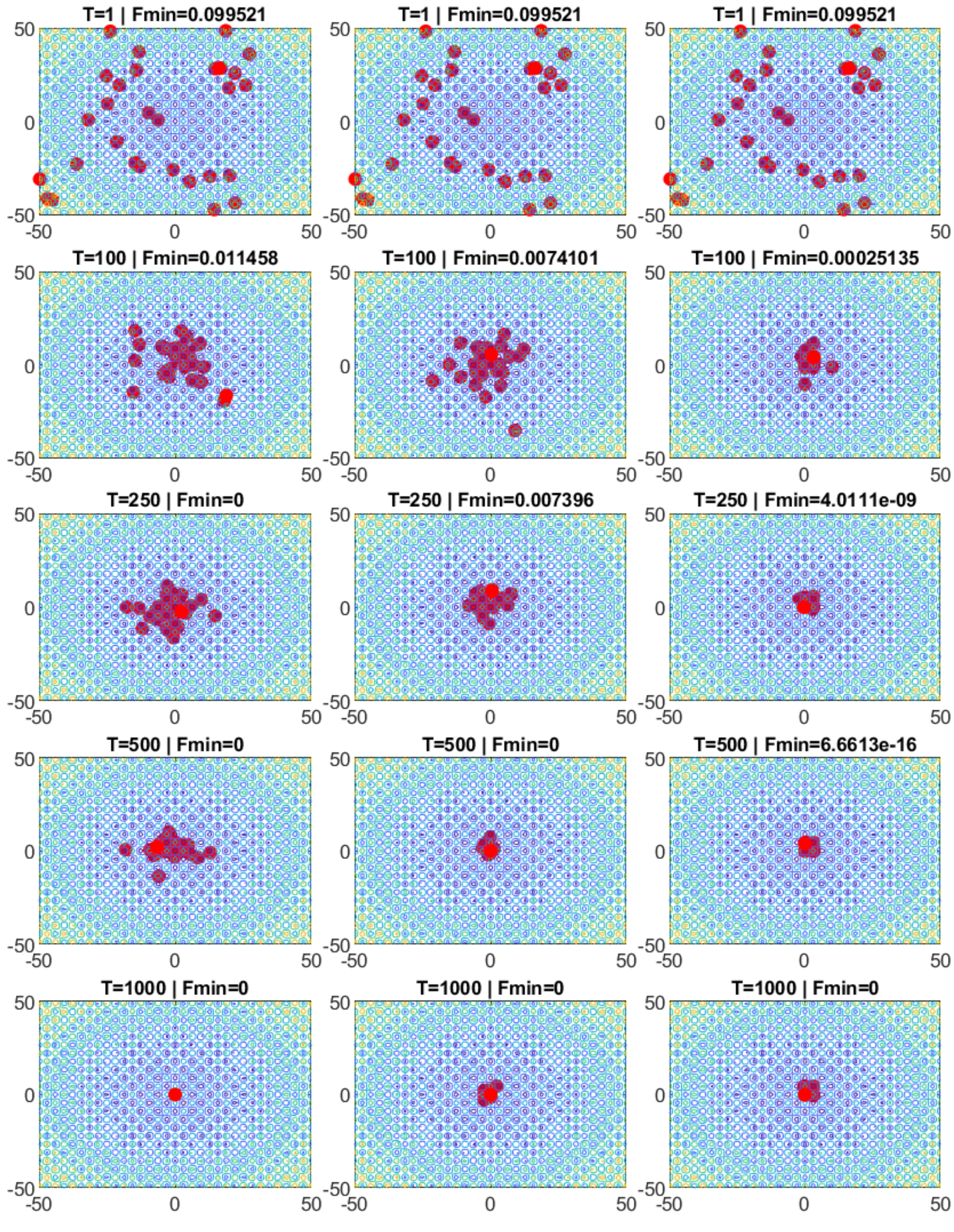

Figure 26: Search behaviour of AHPSO (left column), HPSO-TVAC (middle column), MaPSO (right column) on 2-dimensional Griewank function.

the end of the allocated iterations. On the other hand, HPSO-TVAC yields a more controlled, yet still partially active population, whereas MaPSO typically exhibits a trend toward a nearly converged swarm with low-mobility particles. The captured particle state at the handover point further justifies why certain algorithm combinations and execution orders perform better, supporting our initial hypothesis.

6.8. AHPSO Re-Entry for Diversity Revival

Further experiments were conducted on E-AHPSO to examine whether AHPSO could revive the search process after control had been transferred to either HPSO-TVAC or MaPSO. The purpose of these experiments was to investigate whether reintroducing AHPSO at a later stage could restore population diversity after the swarm had already been influenced by the search behaviour of another algorithm within the ensemble. To investigate this, the ensemble ordering of E-AHPSO was modified to reinvoke AHPSO as the late-stage optimiser, yielding two loop-based variants: AHPSO→MaPSO→AHPSO and AHPSO→HPSO-TVAC→AHPSO.

Experiments conducted on 10, 30, 50, and 100-dimensional problems using the CEC'13, CEC'14, and CEC'17 benchmark suites exhibited a consistent trend. As shown in Figures 30-31, in both variants where AHPSO is reinvoked after HPSO-TVAC and MaPSO, AHPSO noticeably restored population diversity during the final phase. This indicates its ability to reintroduce exploratory behaviour after the swarm had passed through a different search phase, particularly following a more exploitative algorithm. This effect was observed in both AHPSO → HPSO-TVAC → AHPSO and AHPSO → MaPSO → AHPSO configurations, suggesting that AHPSO retains a strong capacity to increase diversity even when it is not used only as the initial search component. In this sense, the results support AHPSO's role as a population revival mechanism, capable of re-expanding the search process when the swarm begins to lose diversity or become concentrated in narrower regions of the search space.

However, although both loop-based mechanisms demonstrated the ability to revive population diversity, reinvoking AHPSO in the final phase did not provide the same level of generalisation capability or overall optimisation performance as the proposed E-AHPSO algorithm. While the final AHPSO phase helped recover diversity, this did not consistently translate into superior convergence behaviour or stronger final solution quality across the different benchmark suites and dimensional settings. Therefore, while AHPSO

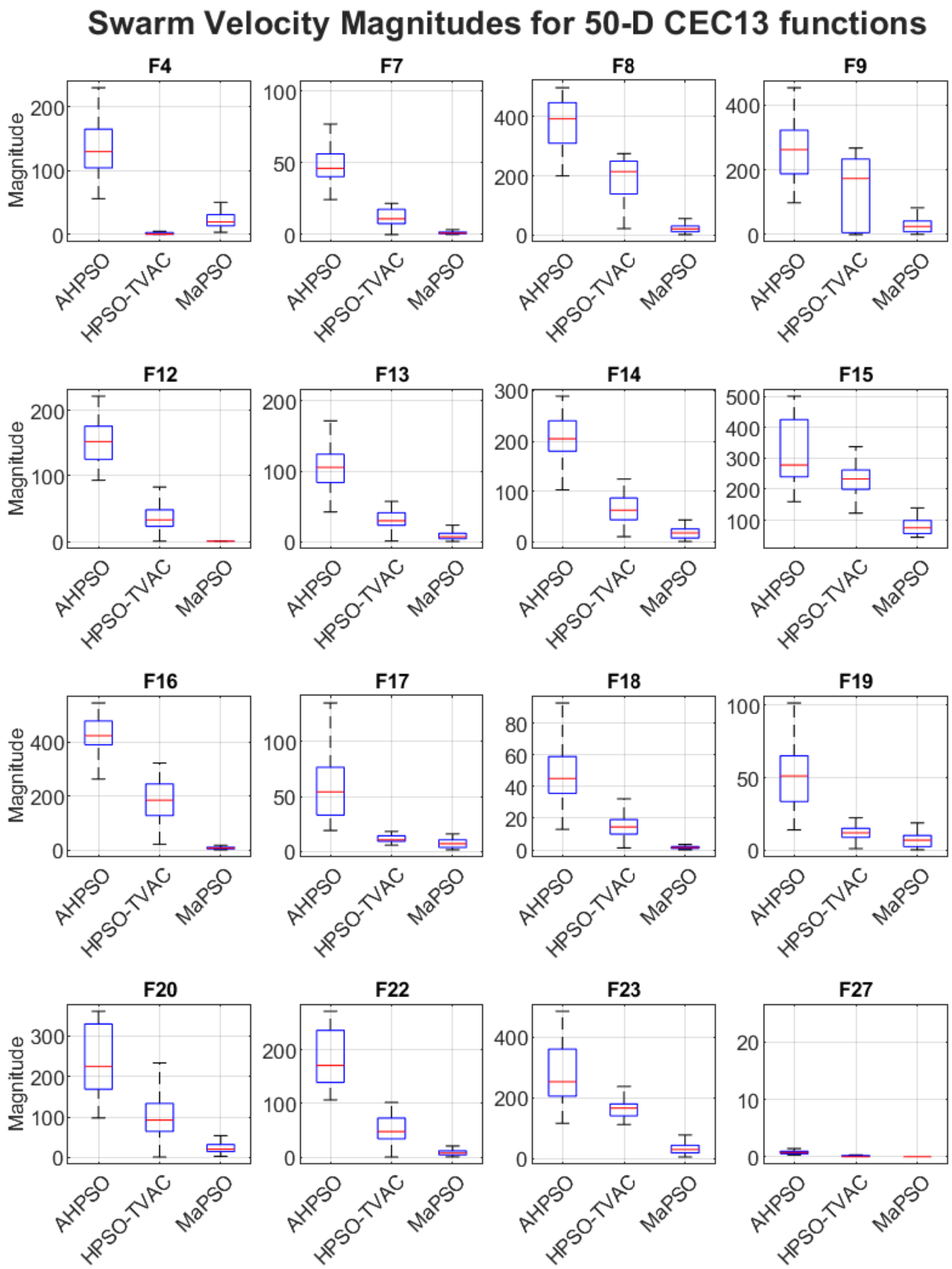

Figure 27: Velocity magnitude distributions of AHPSO, HPSO-TVAC and MaPSO at the handover point for 50-dimensional CEC'13 benchmark functions, illustrating the swarm's particle mobility and exploration state before transition to the next algorithmic phase.

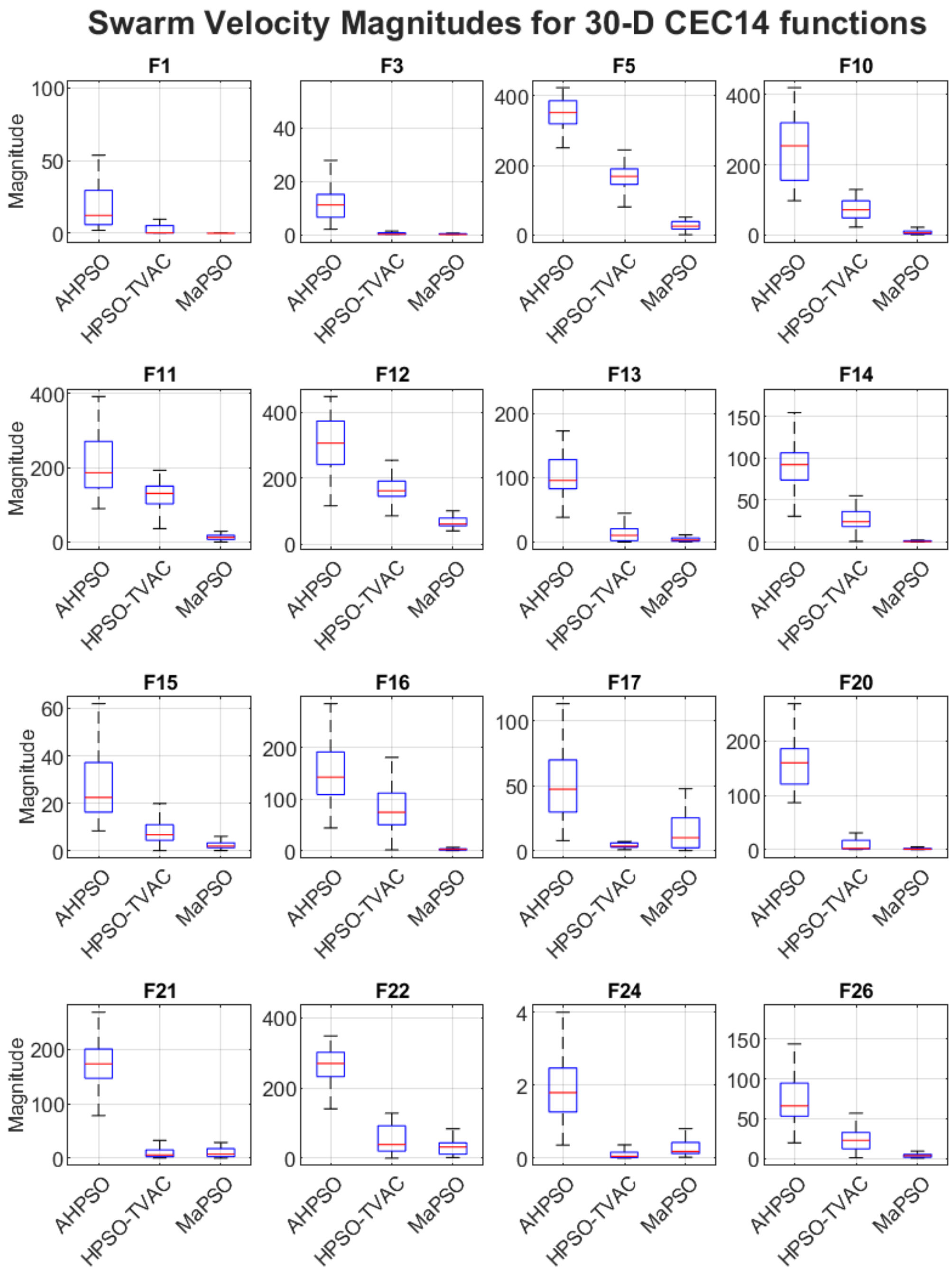

Figure 28: Velocity magnitude distributions of AHPSO, HPSO-TVAC and MaPSO at the handover point for 30-dimensional CEC'14 benchmark functions, illustrating the swarm's particle mobility and exploration state before transition to the next algorithmic phase.

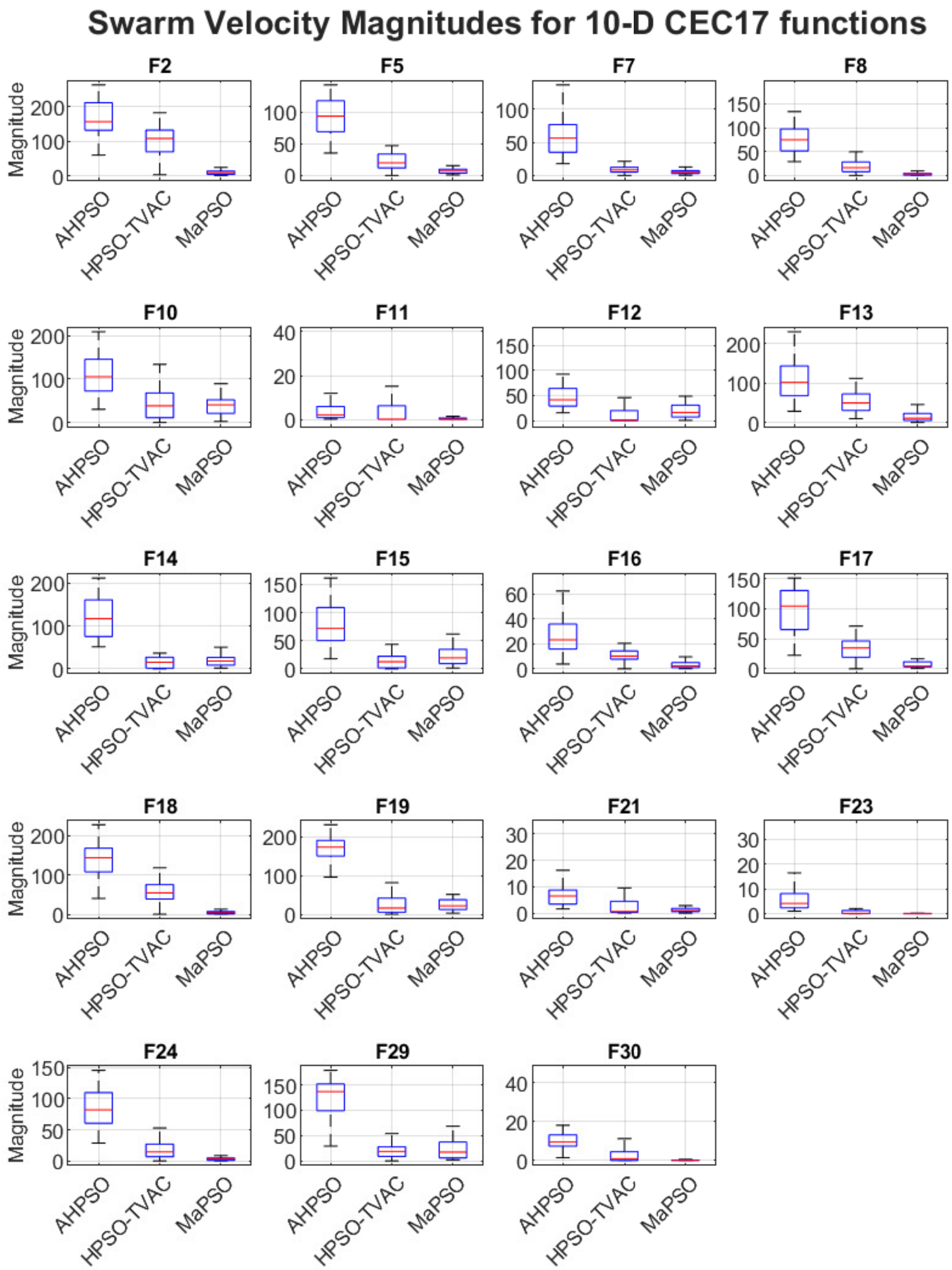

Figure 29: Velocity magnitude distributions of AHPSO, HPSO-TVAC and MaPSO at the handover point for 10-dimensional CEC'17 benchmark functions, illustrating the swarm's particle mobility and exploration state before transition to the next algorithmic phase.

was discovered to be effective as a late-stage diversity revival mechanism, the loop-based variants appear less robust and less broadly effective than the proposed E-AHPSO algorithm. This suggests that diversity recovery alone is insufficient to explain the effectiveness of E-AHPSO; rather, the specific ordering and complementary interactions among AHPSO, HPSO-TVAC, and MaPSO are critical to achieving stronger and more generalisable optimisation performance.

6.9. Success Rates Analysis

Despite its superior performance on higher-dimensional problems, our experiments revealed that the proposed E-AHPSO algorithm performs moderately on low-dimensional problems compared to its high-dimensional performance. We have calculated success rates to better understand E-AHPSO's performance compared to other algorithms on 10-dimensional problems.

A run is considered successful if it meets the following criteria:

$$|f_{best} - f_{optimum}| \leq \epsilon \tag{27}$$

where f_{best} is the best objective value discovered, $f_{optimum}$ is the global optimum, and ϵ is the tolerance threshold (10^{-8}) as specified in test suite reports[128].

Each algorithm is run 51 times on each CEC'13 and CEC'17 10-dimensional problems, and the success rate is calculated as:

$$SuccessRate = (\frac{NumberOfSuccessfulRuns}{NumberOfRuns}) * 100 \tag{28}$$

Figures 32-33 display the success rates for each of the 10-dimensional CEC'13 and CEC'17 problems across 40 algorithms, including the proposed E-AHPSO.

The results reveal that E-AHPSO successfully converged to the global optima for a few functions, including the unimodal functions (F1, F5; 100% success rate) and the multimodal function (F11; 50% success rate). We also observe a similar pattern with many of the comparator algorithms, where the success rate is concentrated on selected functions rather than being distributed across the entire set of 10-dimensional CEC'13 problems. In addition, we generally observe a higher success rate for the DE variants MTDE, CoDE, and L-SHADE across unimodal and multimodal functions.

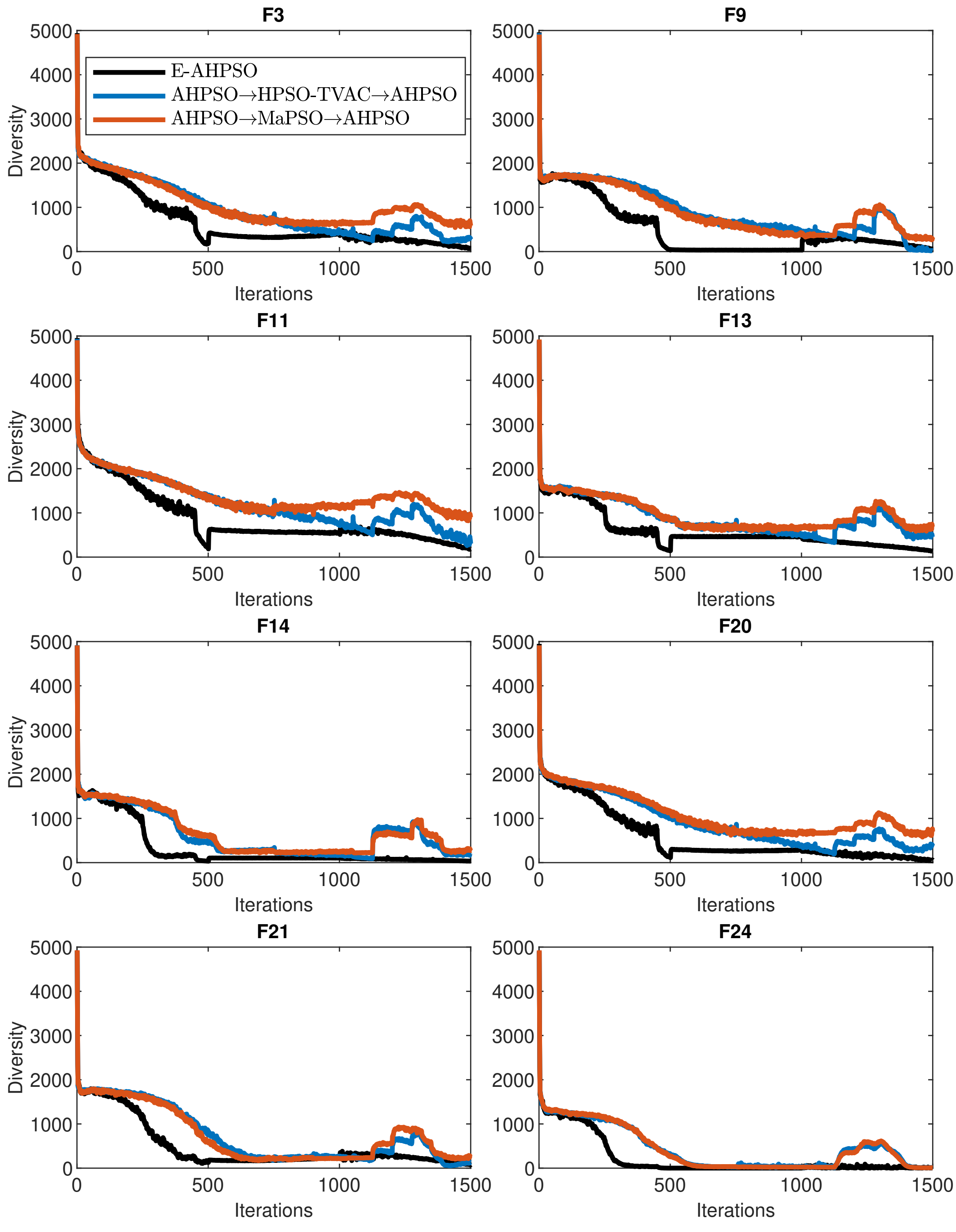

Figure 30: Population diversity trajectories of E-AHPSO and the two AHPSO re-entry variants, AHPSO → HPSO-TVAC → AHPSO and AHPSO → MaPSO → AHPSO, illustrating how late-stage AHPSO reinvocation restores swarm diversity relative to the original ensemble execution across selected 100-dimensional CEC'14 benchmark problems.

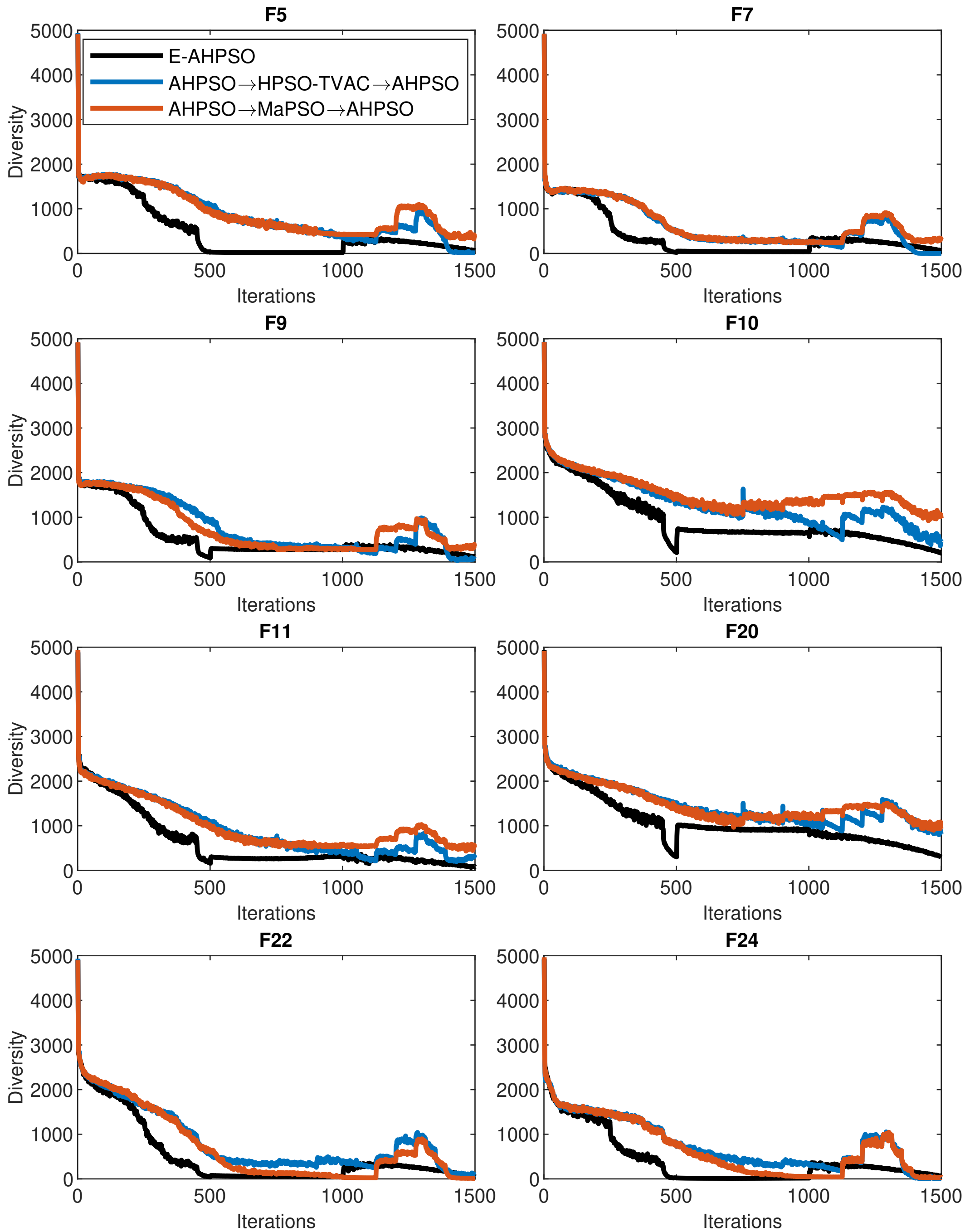

Figure 31: Population diversity trajectories of E-AHPSO and the two AHPSO re-entry variants, AHPSO → HPSO-TVAC → AHPSO and AHPSO → MaPSO → AHPSO, illustrating how late-stage AHPSO reinvocation restores swarm diversity relative to the original ensemble execution across selected 100-dimensional CEC'17 benchmark problems.

The success rate results for the 10-dimensional CEC'17 functions are shown in Figure 33. We observe a similar pattern of higher success rates for CoDE, L-SHADE, and MTDE across a wider range of functions within the 10-dimensional CEC'17 test suites. It may be worth noting that E-AHPSO has better coverage of the 10-dimensional CEC'17 functions (compared to CEC'13) and has converged to global optima on unimodal, multimodal, hybrid, and composition functions.

Furthermore, the average success rates presented in Figures 34-35 place E-AHPSO among the competitive algorithms rather than among the lower-performing group. Nevertheless, the results confirm that E-AHPSO is not consistently superior on 10-dimensional problems, particularly on relatively simple unimodal landscapes, where several simpler and more exploitation-focused algorithms achieve higher success rates. On such landscapes, a direct search strategy can converge more rapidly because less extensive exploration and behavioural coordination are required.

This result highlights a trade-off inherent in the ensemble design of E-AHPSO. Its multi-phase search process, diversity-maintenance mechanisms, and complementary exploration–exploitation behaviours provide greater robustness as the dimensionality and structural complexity of the problem increase. However, these mechanisms may introduce unnecessary search and coordination overhead on low-dimensional unimodal landscapes, where simpler PSO or DE variants can exploit the search space more directly and converge faster. Consequently, E-AHPSO's comparatively moderate performance on some 10-dimensional problems should not be interpreted as a general weakness, but as the cost of achieving broader robustness and generalisation across multimodal, hybrid, composition, and higher-dimensional problems.

It should also be noted that a higher success rate does not necessarily correspond to better average optimisation performance. For example, E-AHPSO achieved better average performance on the 10-dimensional CEC'13 problems than on the corresponding CEC'14 and CEC'17 problems (see section 5), but it attained global-optimum coverage across a wider range of function types in the 10-dimensional CEC'17 suite. Taken together, these findings indicate that the principal advantage of E-AHPSO lies in its robust performance across diverse and complex search landscapes, whereas simpler algorithms may retain an advantage in convergence speed on certain low-dimensional unimodal problems.

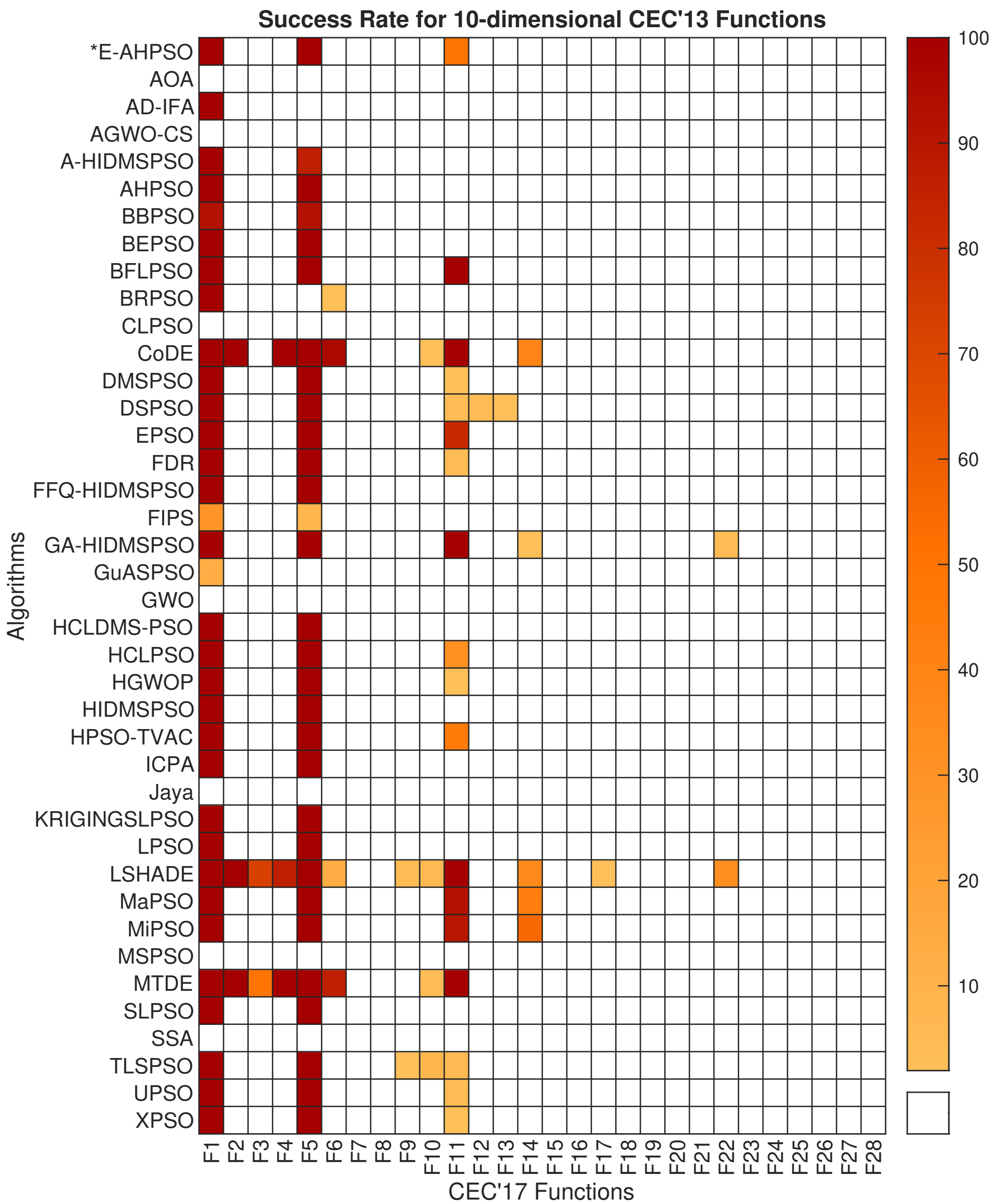

Figure 32: Comparison of the Success Rates for 10-dimensional CEC’13 functions

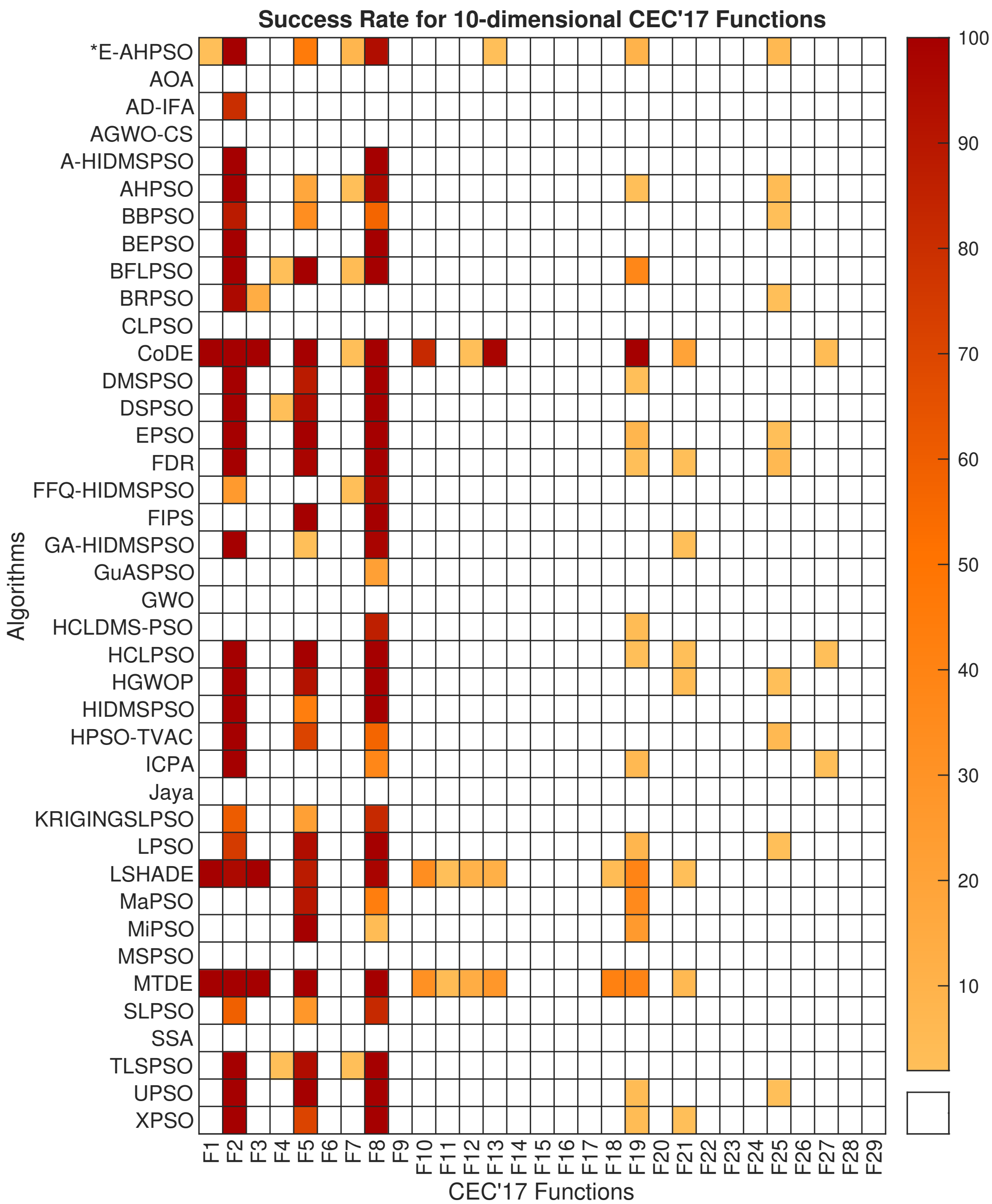

Figure 33: Comparison of the Success Rates for 10-dimensional CEC'17 functions

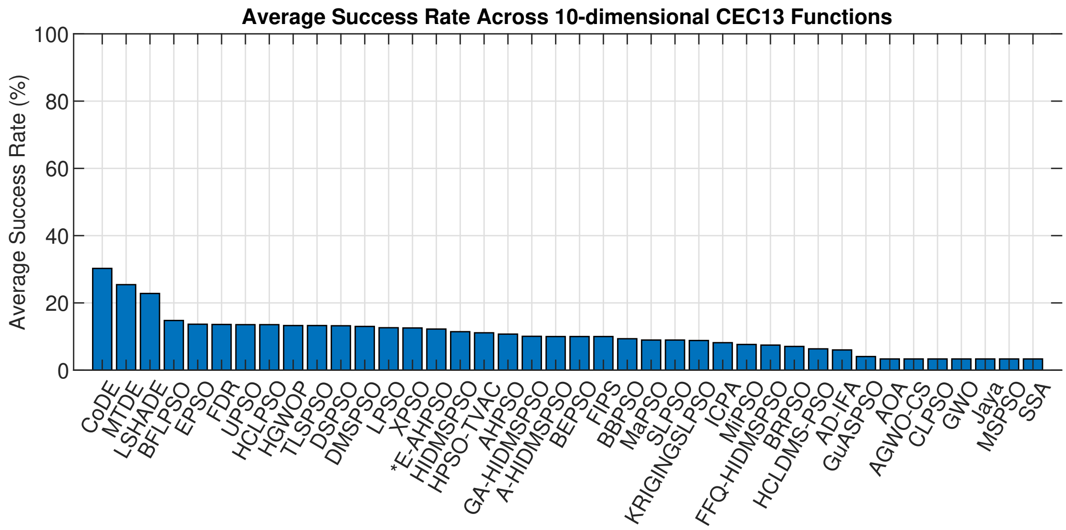

Figure 34: Average Success Rate Across 10-dimensional CEC'13 Functions

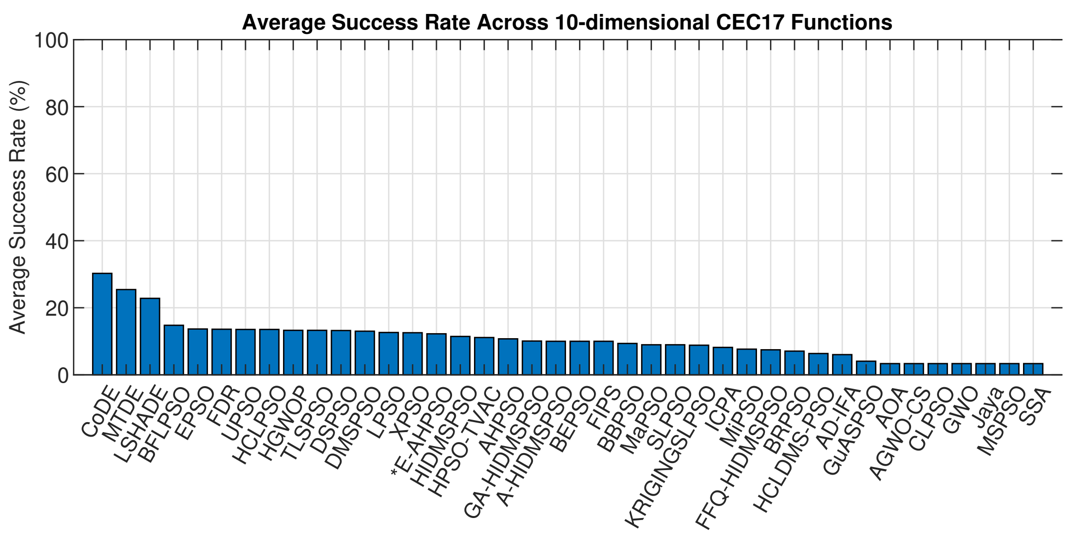

Figure 35: Average Success Rate Across 10-dimensional CEC'17 Functions

7. Conclusions

This paper highlighted the strengths and weaknesses of a recently published PSO variant: the Altruistic Heterogeneous PSO (AHPSO) algorithm. AHPSO is highly explorative and has proven to be effective on high-dimensional and complex problems. However, it can exhibit mediocre performance on relatively simpler problems. The source of this limitation can be traced to the heterogeneous properties of AHPSO, which enable the discovery of promising regions but can result in less efficient exploitation and convergence. Considering this, we investigated the potential for collaboration between the AHPSO algorithm and a pool of well-known, simpler, more exploitative PSO variants to build a powerful, general-purpose ensemble algorithm that would perform well across many different problem types and dimensions. The ultimate objective of this work was to design a PSO-based ensemble algorithm that used AHPSO as a central explorative mechanism, assisted by a set of complementary exploitative algorithms. Detailed preliminary experiments, using the CEC'17 test suite, revealed a promising candidate, namely E-AHPSO. Subsequently, E-AHPSO was further tested against 39 benchmark algorithms, comprising recent and mature PSO variants, non-PSO algorithms including powerful differential evolution variants, L-SHADE, CoDE, and MTDE, on 10, 30, 50 and 100-dimensional CEC'13, CEC'14, CEC'17, and CEC'20 (10 and 20-dimensional) test suite problems. As part of our primary objective, the work also aimed to develop a general-purpose algorithm that is highly competitive across a very wide range of continuous-variable problems and problem dimensions, and that can be used 'out of the box' without any need for parameter tuning. In the very thorough comparative testing, E-AHPSO achieved a highly competitive average final rank across all test suites and dimensions. It was statistically significantly (at a significance level of $\alpha = 0.05$) better than 64%, 71% and 78% of the large and challenging set of benchmark comparator algorithms on the CEC'13, CEC'14 and CEC'17 test suites, respectively, averaged over 10, 30, 50 and 100 dimensions. None of the comparator algorithms was statistically significantly better than E-AHPSO on the CEC'14 test suite at any dimension, while a maximum of one of the comparator algorithms was significantly better on some of the lower dimensions for the CEC'17 and CEC'13 test suites. Moreover, the statistical results on the recent CEC'20 test suite also indicated that the E-AHPSO was significantly better than (at a significance level of $\alpha = 0.05$) 56% and 64% of the 39 benchmark algorithms at dimensions 10 and 20, respectively, whilst none of the 39 benchmark algorithms was statistically significantly better than E-AHPSO. Analysis revealed that E-AHPSO performed consistently very well across all problem types and dimensions in the test suites. Its convergence and population diversity dynamics were comparable to or better than those of the top-performing comparator algorithms. All of this is strong evidence for E-AHPSO being a good candidate for a powerful general-purpose optimisation algorithm. In the case of comprehensive test suites such as CEC'13, CEC'14, and CEC'17, 100-dimensional problems constitute the most complex instances due to dimensionality. Hence, we particularly emphasise the performance of E-AHPSO on the 100-dimensional CEC'14 and CEC'17 problems, where it ranked 1^{st} among a very competitive set of benchmark algorithms. Considering this, it may be worth highlighting that the proposed algorithm was statistically significantly better than 61%, 79%, and 87% of the 39 benchmark algorithms at $\alpha = 0.05$ on the 100-dimensional CEC'13, CEC'14, and CEC'17 problems, respectively. As mentioned previously, the central hypothesis underpinning our approach was that AHPSO, with its powerful explorative and diversity-maintaining properties, would discover multiple very promising areas of the search space in the initial phase of the search (see section 6.6); these would then be efficiently exploited by the simpler exploitative members of the ensemble to converge to optimal or near-optimal solutions. The very detailed empirical investigations of E-AHPSO described in this paper, including the execution order of the ensemble algorithms (see section 4.3), validate that hypothesis. The algorithm exhibits a very good balance between exploration and exploitation. In a substitution experiment presented in section 6.5, we assigned the role of AHPSO to six comparable PSO variants (DMS-PSO, HIDMS-PSO, HCLPSO, FFQ-HIDMS-PSO, GA-HIDMS-PSO and HIDMS-PSO) in each of the seven ensembles that were shortlisted in the preliminary experiments (see section 4.4). All the AHPSO replacement algorithms used in this experiment failed to yield comparable performance; some were simply unresponsive to such collaboration, and others exhibited performance deterioration. These results further verified the central role of AHPSO in an ensemble configuration.

Several limitations and performance trade-offs of E-AHPSO were identified during this study. Although E-AHPSO performed competitively on lower-dimensional problems and outperformed many of the benchmark algorithms, including AHPSO, its relative advantage became more pronounced as problem dimensionality and structural complexity increased. This pattern reflects a trade-off inherent in its ensemble design. On comparatively simple, low-dimensional unimodal landscapes, simpler and more exploitation-focused PSO variants may converge faster because they can exploit the search space directly, without requiring the multi-phase search behaviour and coordination employed by E-AHPSO. In contrast, E-AHPSO's complementary exploration–exploitation mechanisms, diversity maintenance, and sequential transfer of the evolving population become increasingly beneficial on multimodal, hybrid, composition, and higher-dimensional problems. Therefore, its moderate performance on certain 10-dimensional problems should not be interpreted as a general weakness, but as the cost of achieving greater robustness and generalisation across a wider range of complex optimisation landscapes. Future work could investigate adaptive mechanisms for shortening or bypassing exploratory phases when the search landscape appears relatively simple, thereby improving convergence on low-dimensional problems while preserving the robustness demonstrated on more complex problems. Further improvements to convergence speed and diversity maintenance may also enhance the algorithm's overall performance. In addition, the analysis in section 6.8 showed that reintroducing AHPSO as a late-stage optimiser can visibly restore population diversity, although this recovery alone did not consistently produce better final optimisation performance.

Besides our efforts to design a competitive ensemble algorithm, we also intended to minimise the algorithm's time complexity. Often, algorithms that perform well on high-dimensional and more complex problems compromise on time complexity due to their complex algorithmic dynamics. As highlighted in section 6.4, overall, E-AHPSO exhibited lower time complexity compared to 58%, 64%, 65% and 68% of the 39 benchmark algorithms for 10, 30, 50 and 100-dimensional problems, respectively. Considering the existing applications of the standard AHPSO algorithm on various real-world problems [15][3], including parameter optimisation of Hybrid Active Power Filters (HAPF) [16], Transmission Network Expansion Planning, Planetary Gear Train Design, and the Robot Gripper Design problem [129]. Hence, as a much more efficient incremental version, the E-AHPSO algorithm presented in this paper also has significant potential for real-world applications, which is an interesting avenue for future work.

Finally, the novel contribution of this paper can be summarised as the systematic methodology utilised to construct a PSO-based ensemble algorithm and the resulting sequential ensemble algorithm, E-AHPSO. In summary, the work conducted in this paper empirically identifies both the optimal combination of constituent algorithms and their most effective execution order as opposed to assembling an ensemble via ad hoc selection, adaptive switching, or equal treatment of all constituent algorithms. Algorithmically, the proposed (E-AHPSO) is novel because AHPSO is deliberately positioned as the central exploratory mechanism, responsible for discovering promising regions of the search space and maintaining population diversity, before passing the evolved population to MaPSO and HPSO-TVAC for further refinement and exploitation. Hence, the resulting work is not merely a combination of existing PSO variants but a purposefully engineered collaboration in which constituent algorithms have distinct roles. By exploiting AHPSO's strong exploration capability and combining it with the convergence strengths of simpler PSO variants, the proposed E-AHPSO achieves stronger overall performance and broader generalisation capability compared to its constituent algorithms as demonstrated across multiple CEC benchmark suites, problem types and dimensions without any parameter tuning.

Further theoretical analysis of the algorithm and investigations of its efficacy in other domains, such as combinatorial problems or mixed continuous-discrete variable problems, are other fruitful directions for interested researchers. The empirical methodology used in this work to develop ensemble algorithms could serve as a reference for creating new, more advanced collaborative algorithms.

References

[1] D. H. Wolpert and W. G. Macready. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[2] Fevzi Tugrul Varna and Phil Husbands. AHPSO: Altruistic Heterogeneous Particle Swarm Optimisation Algorithm for Global Optimisation. In *2021 IEEE Symposium Series on Computational Intelligence, SSCI 2021 - Proceedings*, pages 1–8. IEEE, dec 2021.

[3] Fevzi Tugrul Varna and Phil Husbands. Two New Bio-Inspired Particle Swarm Optimisation Algorithms for Single-Objective Continuous Variable Problems Based on Eavesdropping and Altruistic Animal Behaviours. *Biomimetics*, 9(9):538, 2024.

[4] Fuqing Zhao, Fei Ji, Tianpeng Xu, Ningning Zhu, and Jonrinaldi. Hierarchical parallel search with automatic parameter configuration for

particle swarm optimization. *Applied Soft Computing*, 151:111126, jan 2024.

[5] Dikshit Chauhan and Anupam Yadav. An archive-based self-adaptive artificial electric field algorithm with orthogonal initialization for real-parameter optimization problems. *Applied Soft Computing*, 150:111109, jan 2024.

[6] Hui Chen, Li-Yong Shen, Chunmei Wang, Lei Tian, and Sanguo Zhang. Multi Actors-Critic based particle swarm optimization algorithm. *Neurocomputing*, 624:129460, apr 2025.

[7] Chan Huang, Jinhao Yu, and Junhui Yang. Top-level dual exploitation particle swarm optimization. *Memetic Computing*, 16(1):1–28, mar 2024.

[8] Xufen Cai, Long Ye, and Qin Zhang. Ensemble learning particle swarm optimization for real-time UWB indoor localization. *Eurasip Journal on Wireless Communications and Networking*, 2018(1):125, dec 2018.

[9] E. K. Burke, G. Kendall, and E. Soubeiga. A Tabu-Search Hyperheuristic for Timetabling and Rostering. *Journal of Heuristics*, 9(6):451–470, dec 2003.

[10] Guohua Wu, Rammohan Mallipeddi, and Ponnuthurai Nagaratnam Suganthan. Ensemble strategies for population-based optimization algorithms – A survey. *Swarm and Evolutionary Computation*, 44:695–711, feb 2019.

[11] Andries P. Engelbrecht. Heterogeneous particle swarm optimization. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6234 LNCS, pages 191–202. Springer, 2010.

[12] Filipe V. Nepomuceno and Andries P. Engelbrecht. A self-adaptive heterogeneous PSO for real-parameter optimization. In *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, pages 361–368. IEEE, jun 2013.

[13] Guohua Wu, Rammohan Mallipeddi, P.N. Suganthan, Rui Wang, and Huangke Chen. Differential evolution with multi-population based ensemble of mutation strategies. *Information Sciences*, 329:329–345, feb 2016.

[14] Yu Xue, Shuiming Zhong, Yi Zhuang, and Bin Xu. An ensemble algorithm with self-adaptive learning techniques for high-dimensional numerical optimization. *Applied Mathematics and Computation*, 231:329–346, mar 2014.

[15] Fevzi Tugrul Varna. *Design and implementation of bio-inspired heterogeneous particle swarm optimisation algorithms for unconstrained and constrained problems*. Phd, University of Sussex, 2023.

[16] Anupam Yadav Chauhan, Dikshit. Optimizing the parameters of hybrid active power filters through a comprehensive and dynamic multi-swarm gravitational search algorithm. *Engineering Applications of Artificial Intelligence*, 123, 2023.

[17] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren. Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing Journal*, 11(2):1679–1696, mar 2011.

[18] J. Kennedy. Bare bones particle swarms. In *2003 IEEE Swarm Intelligence Symposium, SIS 2003 - Proceedings*, pages 80–87. IEEE, 2003.

[19] Yuhui Shi and Russell C. Eberhart. Parameter selection in particle swarm optimization. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 1447, pages 591–600. Springer, 1998.

[20] Asanga Ratnaweera, Saman K. Halgamuge, and Harry C. Watson. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation*, 8(3):240–255, jun 2004.

[21] R. Mendes, J. Kennedy, and J. Neves. The Fully Informed Particle Swarm: Simpler, Maybe Better. *IEEE Transactions on Evolutionary Computation*, 8(3):204–210, jun 2004.

[22] K.E. Parsopoulos and M.N. Vrahatis. UPSO: A Unified Particle Swarm Optimization Scheme. In *International Conference of Computational Methods in Sciences and Engineering 2004 (ICCMSE 2004)*, pages 868–873. CRC Press, apr 2019.

[23] Mohammad Hasanzadeh, Mohammad Reza Meybodi, and Mohammad Mehdi Ebadzadeh. Adaptive parameter selection in comprehensive learning particle swarm optimizer. In *Communications in Computer and Information Science*, volume 427, pages 267–276. Springer, 2014.

[24] Ryoji Tanabe and Alex S. Fukunaga. Improving the search performance of SHADE using linear population size reduction. In *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, pages 1658–1665. IEEE, jul 2014.

[25] Yong Wang, Zixing Cai, and Qingfu Zhang. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation*, 15(1):55–66, feb 2011.

[26] Mohammad H. Nadimi-Shahraki, Shokooh Taghian, Seyedali Mirjalili, and Hossam Faris. MTDE: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems. *Applied Soft Computing Journal*, 97:106761, dec 2020.

[27] Yu Zhou, Junhao Kang, and Hainan Guo. Many-objective optimization of feature selection based on two-level particle cooperation. *Information Sciences*, 532:91–109, sep 2020.

[28] Kexin Hu, Zhongjing Ma, Suli Zou, Jian Li, and Jinhui Zhang. Game-Based Social Learning Particle Swarm Optimizer for Inverse Kinematics of Robotic Arms. *IEEE Robotics and Automation Letters*, 10(7):7078–7085, jul 2025.

[29] Madhusmita Mishra and D. P. Acharjya. A review on swarm optimization, hybridization and extent of applications. *International Journal of Data Science and Analytics*, may 2025.

[30] Emir Turajlic. Particle Swarm Optimization with Time-Varying Acceleration Coefficients in Multilevel Image Thresholding. In *2024 IEEE*

14th Symposium on Computer Applications and Industrial Electronics (ISCAIE), pages 413–417. IEEE, may 2024.

[31] Yin Bo, Xiaogang Guo, Quansheng Liu, Yucong Pan, Lei Zhang, and You Lu. Prediction of tunnel deformation using PSO variant integrated with XGBoost and its TBM jamming application. *Tunnelling and Underground Space Technology*, 150:105842, aug 2024.

[32] J. Kennedy and R. Eberhart. Particle swarm optimization. In *ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995.

[33] Y. Shi and 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)*, pages 69–73. IEEE, 1998.

[34] Yuhui Shi and Russell C. Eberhart. Empirical study of particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, volume 3, pages 1945–1950. IEEE, 1999.

[35] Li Li, Bing Xue, Ben Niu, Yujuan Chai, and Jianhuang Wu. The novel non-linear strategy of inertia weight in particle swarm optimization. In *2009 Fourth International on Conference on Bio-Inspired Computing*, pages 1–5. IEEE, oct 2009.

[36] W. Al-Hassan, M. Fayek, and S. Shaheen. PSOSA: An Optimized Particle Swarm Technique for Solving the Urban Planning Problem. In *2006 International Conference on Computer Engineering and Systems*, pages 401–405. IEEE, nov 2006.

[37] Gao Yue-lin and Duan Yu-hong. A New Particle Swarm Optimization Algorithm with Random Inertia Weight and Evolution Strategy. In *2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007)*, pages 199–203. IEEE, dec 2007.

[38] Jianxin Wu, XiangXin He, WeiGuo Zhao, and Rui Wang. Exponential inertia weight particle swarm algorithm for dynamics optimization of electromechanical coupling system. In *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, pages 479–483. IEEE, nov 2009.

[39] Zhi-Hui Zhan, Jun Zhang, Yun Li, and H.S.-H. Chung. Adaptive Particle Swarm Optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(6):1362–1381, dec 2009.

[40] M.R. Tanweer, S. Suresh, and N. Sundararajan. Self regulating particle swarm optimization algorithm. *Information Sciences*, 294:182–202, feb 2015.

[41] Millie Pant, T. Radha, and V. P. Singh. Particle Swarm Optimization Using Gaussian Inertia Weight. In *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, pages 97–102. IEEE, dec 2007.

[42] Marco S. Nobile, Paolo Cazzaniga, Daniela Besozzi, Riccardo Colombo, Giancarlo Mauri, and Gabriella Pasi. Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization. *Swarm and Evolutionary Computation*, 39:70–85, apr 2018.

[43] B. Y. Qu, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das. A distance-based locally informed particle swarm model for multimodal optimization. *IEEE Transactions on Evolutionary Computation*, 17(3):387–402, jun 2013.

[44] James Kennedy and Rui Mendes. Population structure and particle swarm performance. In *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, volume 2, pages 1671–1676. IEEE, 2002.

[45] Stefan Janson and Martin Middendorf. A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(6):1272–1282, dec 2005.

[46] Daniel Parrott and Xiaodong Li. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Transactions on Evolutionary Computation*, 10(4):440–458, aug 2006.

[47] Hui Wang, Zhijian Wu, Shahryar Rahnamayan, Changhe Li, Sanyou Zeng, and Dazhi Jiang. Particle swarm optimisation with simple and efficient neighbourhood search strategies. *International Journal of Innovative Computing and Applications*, 3(2):97–104, 2011.

[48] Rui Mendes. *Population Topologies and Their Influence in Particle Swarm Performance.* PhD thesis, Universidade do Minho, 2004.

[49] Mahdi Setayesh, Mengjie Zhang, and Mark Johnston. Effects of static and dynamic topologies in particle swarm optimisation for edge detection in noisy images. In *2012 IEEE Congress on Evolutionary Computation, CEC 2012*, pages 1–8. IEEE, jun 2012.

[50] P. N. Suganthan. Particle swarm optimiser with neighbourhood operator. In *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, volume 3, pages 1958–1962. IEEE, 1999.

[51] Lin Wang, Bo Yang, and Jeff Orchard. Particle swarm optimization using dynamic tournament topology. *Applied Soft Computing Journal*, 48:584–596, nov 2016.

[52] J. J. Liang and P. N. Suganthan. Dynamic multi-swarm particle swarm optimizer with a novel constraint-handling mechanism. In *2006 IEEE Congress on Evolutionary Computation, CEC 2006*, pages 9–16. IEEE, 2006.

[53] Xuewen Xia, Ling Gui, and Zhi Hui Zhan. A multi-swarm particle swarm optimization algorithm based on dynamical topology and purposeful detecting. *Applied Soft Computing Journal*, 67:126–140, jun 2018.

[54] S. Z. Zhao, J. J. Liang, P. N. Suganthan, and M. F. Tasgetiren. Dynamic multi-swarm particle swarm optimizer with local search for Large Scale Global Optimization. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 3845–3852. IEEE, jun 2008.

[55] Wenxing Ye, Weiying Feng, and Suohai Fan. A novel multi-swarm particle swarm optimization with dynamic learning strategy. *Applied Soft Computing*, 61:832–843, dec 2017.

[56] J.J. Liang, A.K. Qin, P.N. Suganthan, and S. Baskar. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10(3):281–295, jun 2006.

[57] Nandar Lynn and Ponnuthurai Nagaratnam Suganthan. Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. *Swarm and Evolutionary Computation*, 24:11–24, oct 2015.

[58] Changhe Li, Shengxiang Yang, and Trung Thanh Nguyen. A self-learning particle swarm optimizer for global optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(3):627–646, jun 2012.

[59] Hui Wang, Zhijian Wu, Shahryar Rahnamayan, Yong Liu, and Mario Ventresca. Enhancing particle swarm optimization using generalized opposition-based learning. *Information Sciences*, 181(20):4699–4714, oct 2011.

[60] Wei Hong Lim and Nor Ashidi Mat Isa. An adaptive two-layer particle swarm optimization with elitist learning strategy. *Information Sciences*, 273:49–72, jul 2014.

[61] Guiping Xu, Quanlong Cui, Xiaohu Shi, Hongwei Ge, Zhi Hui Zhan, Heow Pueh Lee, Yanchun Liang, Ran Tai, and Chunguo Wu. Particle swarm optimization based on dimensional learning strategy. *Swarm and Evolutionary Computation*, 45:33–51, mar 2019.

[62] Ran Cheng and Yaochu Jin. A social learning particle swarm optimization algorithm for scalable optimization. *Information Sciences*, 291(C):43–60, jan 2015.

[63] Yunfeng Zhang, Xinxin Liu, Fangxun Bao, Jing Chi, Caiming Zhang, and Peide Liu. Particle swarm optimization with adaptive learning strategy. *Knowledge-Based Systems*, 196:105789, may 2020.

[64] Bo Wei, Xuewen Xia, Fei Yu, Yinglong Zhang, Xing Xu, Hongrun Wu, Ling Gui, and Guoliang He. Multiple adaptive strategies based particle swarm optimization algorithm. *Swarm and Evolutionary Computation*, 57:100731, sep 2020.

[65] Baoxian Liang, Yunlong Zhao, and Yang Li. A hybrid particle swarm optimization with crisscross learning strategy. *Engineering Applications of Artificial Intelligence*, 105:104418, oct 2021.

[66] Xinmin Tao, Xiangke Li, Wei Chen, Tian Liang, Yetong Li, Jie Guo, and Lin Qi. Self-Adaptive two roles hybrid learning strategies-based particle swarm optimization. *Information Sciences*, 578:457–481, nov 2021.

[67] Jiawei Lu, Jian Zhang, and Jianan Sheng. Enhanced multi-swarm cooperative particle swarm optimizer. *Swarm and Evolutionary Computation*, 69:100989, mar 2022.

[68] Han Li, Juan Li, Peishu Wu, Yancheng You, and Nianyin Zeng. A ranking-system-based switching particle swarm optimizer with dynamic learning strategies. *Neurocomputing*, 494:356–367, jul 2022.

[69] Fan Li, Yingli Li, Xiwen Cai, and Liang Gao. A surrogate-assisted hybrid swarm optimization algorithm for high-dimensional computationally expensive problems. *Swarm and Evolutionary Computation*, 72:101096, jul 2022.

[70] Xinming Zhang and Qiuying Lin. Three-learning strategy particle swarm algorithm for global optimization problems. *Information Sciences*, 593:289–313, may 2022.

[71] Weilin Du and Bin Li. Multi-strategy ensemble particle swarm optimization for dynamic optimization. *Information Sciences*, 178(15):3096–3109, aug 2008.

[72] Jasper A. Vrugt, Bruce A. Robinson, and James M. Hyman. Self-adaptive multimethod search for global optimization in real-parameter spaces. *IEEE Transactions on Evolutionary Computation*, 13(2):243–259, apr 2009.

[73] Jacomine Grobler, Andries P. Engelbrecht, Graham Kendall, and V. S.S. Yadavalli. Alternative hyper-heuristic strategies for multi-method global optimization. In *2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010*, pages 1–8. IEEE, jul 2010.

[74] Dawei Zheng, Chao Qin, and Peipei Liu. Adaptive Particle Swarm Optimization Algorithm Ensemble Model Applied to Classification of Unbalanced Data. *Scientific Programming*, 2021:1–13, oct 2021.

[75] Guohua Wu, Rammohan Mallipeddi, and Ponnuthurai Nagaratnam Suganthan. Ensemble strategies for population-based optimization algorithms – A survey. *Swarm and Evolutionary Computation*, 44:695–711, feb 2019.

[76] Xianshun Chen, Yew Soon Ong, Meng Hiot Lim, and Kay Chen Tan. A multi-facet survey on memetic computation. *IEEE Transactions on Evolutionary Computation*, 15(5):591–607, oct 2011.

[77] Crina Grosan and Ajith Abraham. Hybrid evolutionary algorithms: Methodologies, architectures, and reviews. In *Studies in Computational Intelligence*, volume 75, pages 1–17. Springer, 2007.

[78] A. K. Qin, V. L. Huang, and P. N. Suganthan. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 13(2):398–417, apr 2009.

[79] Haiping Ma, Shufei Su, Dan Simon, and Minrui Fei. Ensemble multi-objective biogeography-based optimization with application to automated warehouse scheduling. *Engineering Applications of Artificial Intelligence*, 44:79–90, sep 2015.

[80] Qinqin Fan and Xuefeng Yan. Differential evolution algorithm with self-adaptive strategy and control parameters for P-xylene oxidation process optimization. *Soft Computing*, 19(5):1363–1391, may 2015.

[81] Tetsuyuki Takahama and Setsuko Sakai. Differential evolution with dynamic strategy and parameter selection by detecting landscape modality. In *2012 IEEE Congress on Evolutionary Computation, CEC 2012*, pages 1–8. IEEE, jun 2012.

[82] Mostafa Z. Ali, Noor H. Awad, and Ponnuthurai N. Suganthan. Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization. *Applied Soft Computing*, 33:304–327, aug 2015.

[83] Giovanni Iacca, Fabio Caraffini, and Ferrante Neri. Multi-strategy co-evolving aging particle optimization. *International Journal of Neural Systems*, 24(1):1450008, feb 2014.

[84] Kezong Tang, Zuoyong Li, Limin Luo, and Bingxiang Liu. Multi-strategy adaptive particle swarm optimization for numerical optimization. *Engineering Applications of Artificial Intelligence*, 37:9–19, jan 2015.

[85] Guohan Lin, Kuiyin Zhao, and Qin Wan. Takagi-sugeno fuzzy model identification using coevolution particle swarm optimization with multi-strategy. *Applied Intelligence*, 45(1):187–197, jul 2016.

[86] Chuan Wang, Yancheng Liu, Yang Chen, and Yi Wei. Self-adapting hybrid strategy particle swarm optimization algorithm. *Soft Computing*, 20(12):4933–4963, dec 2016.

[87] Yu Wang, Bin Li, Thomas Weise, Jianyu Wang, Bo Yuan, and Qiongjie Tian. Self-adaptive learning based particle swarm optimization. *Information Sciences*, 181(20):4515–4538, oct 2011.

[88] Hussein Samma, Chee Peng Lim, and Junita Mohamad Saleh. A new Reinforcement Learning-based Memetic Particle Swarm Optimizer. *Applied Soft Computing Journal*, 43:276–297, jun 2016.

[89] Shiu Yin Yuen and Xin Zhang. On composing an algorithm portfolio. *Memetic Computing*, 7(3):203–214, sep 2015.

[90] Sheng Xin Zhang, Shao Yong Zheng, and Li Ming Zheng. An Efficient Multiple Variants Coordination Framework for Differential Evolution. *IEEE Transactions on Cybernetics*, 47(9):2780–2793, sep 2017.

[91] Genghui Li, Qiuzhen Lin, Laizhong Cui, Zhihua Du, Zhengping Liang, Jianyong Chen, Nan Lu, and Zhong Ming. A novel hybrid differential evolution algorithm with modified CoDE and JADE. *Applied Soft Computing Journal*, 47:577–599, oct 2016.

[92] S. Thangavelu and C. Shunmuga Velayutham. An investigation on mixing heterogeneous differential evolution variants in a distributed framework. *International Journal of Bio-Inspired Computation*, 7(5):307–320, 2015.

[93] Guohua Wu, Xin Shen, Haifeng Li, Huangke Chen, Anping Lin, and P. N. Suganthan. Ensemble of differential evolution variants. *Information Sciences*, 423:172–186, jan 2018.

[94] Nandar Lynn and Ponnuthurai Nagaratnam Suganthan. Ensemble particle swarm optimizer. *Applied Soft Computing Journal*, 55:533–548, jun 2017.

[95] W.D. Hamilton. The genetical evolution of social behaviour. I. *Journal of Theoretical Biology*, 7(1):1–16, jul 1964.

[96] Gilbert Roberts. Cooperation: How Vampire Bats Build Reciprocal Relationships. *Current Biology*, 30(7):R307–R309, apr 2020.

[97] Richard C. Connor. Pseudo-reciprocity: Investing in mutualism. *Animal Behaviour*, 34(5):1562–1566, oct 1986.

[98] T. Peram, K. Veeramachaneni, and C. K. Mohan. Fitness-distance-ratio based particle swarm optimization. In *2003 IEEE Swarm Intelligence Symposium, SIS 2003 - Proceedings*, pages 174–181. IEEE, 2003.

[99] Jianbin Xin, Guimin Chen, and Yubao Hai. A Particle Swarm Optimizer with Multi-stage Linearly-Decreasing Inertia Weight. In *2009 International Joint Conference on Computational Sciences and Optimization*, pages 505–508. IEEE, apr 2009.

[100] Yong Feng, Gui-Fa Teng, Ai-Xin Wang, and Yong-Mei Yao. Chaotic Inertia Weight in Particle Swarm Optimization. In *Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007)*, pages 475–475. IEEE, sep 2007.

[101] Ahmad Nickabadi, Mohammad Mehdi Ebadzadeh, and Reza Safabakhsh. A novel particle swarm optimization algorithm with adaptive inertia weight. *Applied Soft Computing*, 11(4):3658–3670, jun 2011.

[102] Ponnuthurai N Suganthan, Nikolaus Hansen, Jing J Liang, Kalyanmoy Deb, Y P Chen, Anne Auger, and S Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *Technical Report, Nanyang Technological University, Singapore, May 2005 AND KanGAL Report 2005005, IIT Kanpur, India*, 2005.

[103] Liang Jing J and Ponnuthurai N. Suganthan Bo Y. Qu. Problem definitions and evaluation criteria for the CEC 2014 special session and

competition on single objective real-parameter numerical optimization. Technical report, Zhengzhou University,Zhongyuan University of Technology,Nanyang Technological University, 2013.

[104] CT Yue, Kenneth V Price, Ponnuthurai N Suganthan, JJ Liang, Mostafa Z Ali, BY Qu, Noor H Awad, and Partha P Biswas. Problem definitions and evaluation criteria for the cec 2020 special session and competition on single objective bound constrained numerical optimization. *Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Tech. Rep*, 201911, 2019.

[105] Jinran Wu, You-Gan Wang, Kevin Burrage, Yu-Chu Tian, Brodie Lawson, and Zhe Ding. An improved firefly algorithm for global continuous optimization problems. *Expert Systems with Applications*, 149:113340, 7 2020.

[106] Satyam Sharma, Ridhi Kapoor, and Sanjeev Dhiman. A novel hybrid metaheuristic based on augmented grey wolf optimizer and cuckoo search for global optimization. In *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*, pages 376–381. IEEE, 5 2021.

[107] Fevzi Tugrul Varna and Phil Husbands. HIDMS-PSO Algorithm with an Adaptive Topological Structure. In *2021 IEEE Symposium Series on Computational Intelligence, SSCI 2021 - Proceedings*, pages 1–8. IEEE, dec 2021.

[108] Laith Abualigah, Ali Diabat, Seyedali Mirjalili, Mohamed Abd Elaziz, and Amir H. Gandomi. The arithmetic optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*, 376:113609, 4 2021.

[109] Fevzi Tugrul Varna and Phil Husbands. Biased Eavesdropping Particles: A Novel Bio-inspired Heterogeneous Particle Swarm Optimisation Algorithm. In *2021 IEEE Symposium Series on Computational Intelligence, SSCI 2021 - Proceedings*, pages 1–8. IEEE, dec 2021.

[110] Xu Chen, Hugo Tianfield, and Wenli Du. Bee-foraging learning particle swarm optimization. *Applied Soft Computing*, 102:107134, apr 2021.

[111] Matthew Settles and Terence Soule. Breeding swarms: A GA/PSO hybrid. In *GECCO 2005 - Genetic and Evolutionary Computation Conference*, pages 161–168, New York, New York, USA, 2005. ACM Press.

[112] Xinming Zhang, Xia Wang, Qiang Kang, and Jinfeng Cheng. Differential mutation and novel social learning particle swarm optimization algorithm. *Information Sciences*, 480:109–129, apr 2019.

[113] Fevzi Tugrul Varna and Phil Husbands. HIDMS-PSO with Bio-inspired Fission-Fusion Behaviour and a Quorum Decision Mechanism. In *2021 IEEE Congress on Evolutionary Computation, CEC 2021 - Proceedings*, pages 1398–1405. IEEE, jun 2021.

[114] Fevzi Tugrul Varna and Phil Husbands. Genetic Algorithm Assisted HIDMS-PSO: A New Hybrid Algorithm for Global Optimisation. In *2021 IEEE Congress on Evolutionary Computation, CEC 2021 - Proceedings*, pages 1304–1311. IEEE, jun 2021.

[115] Farshad Rezaei and Hamid R. Safavi. GuASPSO: a new approach to hold a better exploration–exploitation balance in PSO algorithm. *Soft Computing*, 24(7):4855–4875, apr 2020.

[116] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 3 2014.

[117] Shengliang Wang, Genyou Liu, Ming Gao, Shilong Cao, Aizhi Guo, and Jiachen Wang. Heterogeneous comprehensive learning and dynamic multi-swarm particle swarm optimizer with two mutation operators. *Information Sciences*, 540:175–201, nov 2020.

[118] Xinming Zhang, Qiuying Lin, Wentao Mao, Shangwang Liu, Zhi Dou, and Guoqi Liu. Hybrid Particle Swarm and Grey Wolf Optimizer and its application to clustering optimization. *Applied Soft Computing*, 101:107061, mar 2021.

[119] Fevzi Tugrul Varna and Phil Husbands. HIDMS-PSO: A New Heterogeneous Improved Dynamic Multi-Swarm PSO Algorithm. In *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020*, pages 473–480. IEEE, dec 2020.

[120] Jie Tian, Chaoli Sun, Ying Tan, and Jianchao Zeng. Granularity-based surrogate-assisted particle swarm optimization for high-dimensional expensive optimization. *Knowledge-Based Systems*, 187:104815, jan 2020.

[121] Ayşe Beşkirli and İdiris Dağ. I-CPA: An Improved Carnivorous Plant Algorithm for Solar Photovoltaic Parameter Identification Problem. *Biomimetics*, 8(8):569, nov 2023.

[122] R. Venkata Rao. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, pages 19–34, 2016.

[123] Xinmin Tao, Wenjie Guo, Qing Li, Chao Ren, and Rui Liu. Multiple scale self-adaptive cooperation mutation strategy-based particle swarm optimization. *Applied Soft Computing*, 89:106124, apr 2020.

[124] Shiqiang Qin, Yazhou Zhang, Hongyou Cao, and Juntao Kang. Identification of multiple alternative solutions in structural model updating using multiple subswarm particle swarm optimization. *Engineering Optimization*, 55(2):271–290, feb 2023.

[125] Ran Cheng and Yaochu Jin. A social learning particle swarm optimization algorithm for scalable optimization. *Information Sciences*, 291:43–60, jan 2015.

[126] Mohit Jain, Vijander Singh, and Asha Rani. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. *Swarm and Evolutionary Computation*, 44:148–175, 2 2019.

[127] Xuewen Xia, Ling Gui, Guoliang He, Bo Wei, Yinglong Zhang, Fei Yu, Hongrun Wu, and Zhi-Hui Zhan. An expanded particle swarm optimization based on multi-exemplar and forgetting ability. *Information Sciences*, 508:105–120, jan 2020.

[128] N H Awad, M Z Ali, J J Liang, B Y Qu, and P N Suganthan. Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization. Technical report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 2016.

[129] Libin Hong, Zhantao Gu, Ruibin Bai, John Woodward, and Ender Özcan. An effective combination of mechanisms for particle swarm optimization-based ensemble strategy. *Swarm and Evolutionary Computation*, 99:102154, 12 2025.