biomimetics

MDPI

Article

Two New Bio-Inspired Particle Swarm Optimisation Algorithms for Single-Objective Continuous Variable Problems Based on Eavesdropping and Altruistic Animal Behaviours

Fevzi Tugrul Varna * and Phil Husbands *

AI Group, Department of Informatics, University of Sussex, Brighton BN1 9RH, UK
* Correspondence: f.varna@sussex.ac.uk (F.T.V.); philh@sussex.ac.uk (V.H.)

Abstract: This paper presents two novel bio-inspired particle swarm optimisation (PSO) variants, namely biased eavesdropping PSO (BEPSO) and altruistic heterogeneous PSO (AHPSO). These algorithms are inspired by types of group behaviour found in nature that have not previously been exploited in search algorithms. The primary search behaviour of the BEPSO algorithm is inspired by eavesdropping behaviour observed in nature coupled with a cognitive bias mechanism that enables particles to make decisions on cooperation. The second algorithm, AHPSO, conceptualises particles in the swarm as energy-driven agents with bio-inspired altruistic behaviour, which allows for the formation of lending–borrowing relationships. The mechanisms underlying these algorithms provide new approaches to maintaining swarm diversity, which contributes to the prevention of premature convergence. The new algorithms were tested on the 30, 50 and 100-dimensional CEC'13, CEC'14 and CEC'17 test suites and various constrained real-world optimisation problems, as well as against 13 well-known PSO variants, the CEC competition winner, differential evolution algorithm L-SHADE and the recent bio-inspired I-CPA metaheuristic. The experimental results show that both the BEPSO and AHPSO algorithms provide very competitive performance on the unconstrained test suites and the constrained real-world problems. On the CEC13 test suite, across all dimensions, both BEPSO and AHPSO performed statistically significantly better than 10 of the 15 comparator algorithms, while none of the remaining 5 algorithms performed significantly better than either BEPSO or AHPSO. On the CEC17 test suite, on the 50D and 100D problems, both BEPSO and AHPSO performed statistically significantly better than 11 of the 15 comparator algorithms, while none of the remaining 4 algorithms performed significantly better than either BEPSO or AHPSO. On the constrained problem set, in terms of mean rank across 30 runs on all problems, BEPSO was first, and AHPSO was third.

Keywords: bio-inspired search algorithm; optimisation; particle swarm optimisation; swarm intelligence; altruism; eavesdropping; group behaviour; metaheuristics

Citation: Varna, F.T.; Husbands, P. Two New Bio-Inspired Particle Swarm Optimisation Algorithms for Single-Objective Continuous Variable Problems Based on Eavesdropping and Altruistic Animal Behaviours. *Biomimetics* **2024**, *9*, 538. https://doi.org/10.3390/biomimetics9090538

Academic Editor: Chaoran Cui and Xiaohui Han

Received: 29 June 2024
Revised: 21 August 2024
Accepted: 28 August 2024
Published: 5 September 2024

1. Introduction

In recent years, swarm intelligence algorithms have become one of the most widely used class of optimisation methods [1–3]. Their effectiveness and convenience have led to many variants [4] and successful applications to diverse real-world problems [5–8]. One of the best known and most widely applied swarm algorithms is particle swarm optimisation (PSO) [9]. The PSO algorithm has been extensively investigated with regards to its search dynamics [10,11] and theoretical strengths and limitations [12,13], resulting in many recent extensions developed with the intention of improving the performance of the canonical PSO [14–19].

Despite the development of many variants with diverse inspirations, the core analogy at the heart of most PSO algorithms is biological—a simple model of flocking behaviour observed in many species. Indeed, bio-inspiration has become the dominant driving force for many new metaheuristic algorithms. However, the canonical PSO algorithm's model of

particle movement [9] is relatively simple compared to natural flocking behaviours. Hence, in most cases, the homogeneous nature of canonical PSO particles' behaviour moves them towards a common goal using two standard 'exemplars' (or guiding positions), namely their 'cognitive' (own best position) and 'social' (swarm best position) influences, which tends to trigger rapid loss of diversity, leading to premature convergence. Various approaches have been proposed to address this problem over the past decade or so, including hybridisation with other search algorithms [16,20,21], using extended learning strategies [22–24] and employing more sophisticated topologies to define the local population structure [25–28]. Another powerful contemporary way to potentially minimise this issue and to improve the balance between exploitation and exploration within the search process is to design efficient, heterogeneous agent behaviours to avoid the stagnation of particles and improve overall performance by avoiding premature convergence [14,17,29].

In light of this, in the current paper, we propose two novel PSO algorithms that take their inspiration from forms of dynamic animal group behaviour that, as far as we know, have not previously been used in search algorithms, namely eavesdropping and altruism. These analogies are used to develop algorithms that possess heterogeneous behavioural dynamics at the both agent and swarm levels. Through this heterogeneity, more efficient exploration and exploitation search dynamics are enabled while maintaining diversity and avoiding premature convergence. Such effective exploration and exploitation performance is especially important for efficient searches of high-dimensional and complex problem spaces, which feeds into our overall motivation, i.e., the development of powerful new general-purpose optimisers for unconstrained and constrained single-objective, real-valued problems.

The first of these novel algorithms, BEPSO (biased eavesdropping PSO), is inspired by the alert-signalling behaviour of animals used to attract conspecifics (of the same species or group) to a discovered resource location for potential exploitation and the way in which surrounding heterospecific (of a different species or group) eavesdropper animals try to exploit that information themselves to improve their own fitness.

Eavesdropping plays a significant role in animal communication and the evolution of such communication [30]. Briefly, eavesdropping occurs as a result of animals accessing communication signals transmitted by heterospecifics that were not intended for them. In nature, it is more common for signal interceptors to be intraspecific (of the same species) in order to perceive the call and extract the required information accurately. However, it is not uncommon to observe interspecific animals (competitors of a different species) intercept signals and use them as an advantage to increase their own fitness. Interspecific eavesdropping is particularly interesting, as different species may be proficient in distinct areas within the same habitat and capable of recognising different threats through their distinct sensory capabilities. A concrete example of interspecies eavesdropping is illustrated by the relationship between red squirrels and Eurasian jays [31]. In this case, truly astonishing evolutionary dynamics have resulted in communication between a mammal and a bird, which have become positively biased towards one another and are able to warn and guard each other within the same habitat.

BEPSO takes ideas from animal eavesdropping to develop a set of dynamic interacting mechanisms that underlie particle behaviours and generate movement exemplars that prove to be far more efficient than the simple personal best position and population best positions used by the canonical PSO.

The second of the novel algorithms, altruistic heterogeneous PSO (AHPSO), is inspired by a certain kind of altruistic animal group behaviour.

The role of altruism in evolutionary dynamics was first analysed in mathematical detail by W.D. Hamilton in 1964 [32]. He showed how altruism could arise and be maintained within the Darwinian framework, conferring overall benefit to the group if not to the individual. Traditionally, researchers tended to assess the benefit of altruism to an organism by examining the average number of offspring, with contributors exhibiting less reproductive success in comparison to beneficiaries. However, several different types of

altruistic behaviours have been discovered in various species. Some exemplary altruistic behaviours are observed in social insect colonies such as ants, wasps, bees and termites. In these colonies, the sterile workers are devoted to the queen by protecting the nest and foraging food. By doing so, sterile workers have no reproduction fitness, but they contribute to the queen's reproductive efforts. An example of a more complex organism exhibiting altruistic behaviour is the blood-regurgitating vampire bat, which feeds undernourished bats in their group to avoid starvation [33]. Velvet monkeys exhibit similar behaviour to their groups by giving alarm calls to warn of the presence of predators, putting themselves at risk in doing so [34]. The type, level and results of altruistic behaviour vary widely between organisms, as do the relationships evolved between helper and beneficiary actors.

The AHPSO algorithm incorporates a kind of conditional altruistic behaviour in which particles in a behaviourally heterogeneous population can lend and borrow 'energy', with such transactions depending on the lender's judgment of how 'credit-worthy' a borrower is. These mechanisms generate more effective exemplars than in the canonical PSO, delaying the loss of population diversity and preventing premature convergence, resulting in a highly efficient search algorithm.

The performance of the new BEPSO and AHPSO algorithms was verified over 30, 50 and 100 dimensions of the widely used CEC'13, CEC'14 and CEC'17 benchmark test suites, along with a set of 14 constrained real-world problems, compared against a demanding set of 13 well-known state-of-the-art PSO variants, the 2014 CEC competition winner, L-SHADE (a powerful differential evolution algorithm) and the recent bio-inspired I-CPA metaheuristic. The overall results of this very thorough comparative investigation show that both BEPSO and AHPSO are statistically significantly superior to a large majority of the comparator algorithms and highly competitive against all others, particularly on high-dimensional complex problems (none of the other algorithms were statistically superior to the new algorithms on such problems). In addition, they are shown to be very strong candidates for constrained real-world applications, with BEPSO having the highest mean rank of all the test algorithms on the suite of constrained problems. Both BEPSO and AHPSO achieved robust, high-quality performance across the entire range of test problems and suites *with a single set of parameter values*; they did not need tuning for each new type of problem. Both algorithms were shown to maintain diversity in the population without sacrificing efficient convergence to optimal solutions.

Section 2 discusses related work; then Section 3 describes the proposed algorithms in detail. Section 4 details the experimental method. This is followed by Section 5, in which we present the results of the extensive comparative experiments, and the paper closes with discussion and conclusions in Sections 6 and 7.

2. Related Work

2.1. Bio-inspired Search Algorithms

There has been a surge in the development of bio-inspired search algorithms in the last few decades due to the applicability of these methods across a wide variety of domains and problems. A part of this surging interest is due to the limitation of traditional gradient-based algorithms. In contrast to gradient-based algorithms, bio-inspired algorithms are not gradient-dependent and are significantly less sensitive to the initial solution.

Two of the main research topics in bio-inspired algorithms have been avoidance of premature convergence and sensitivity to/dependence on control parameters. Work in these areas has led to the proposal of many variants of the basic canonical bio-inspired algorithms to address these issues [35–41] .

Various recent and matured bio-inspired algorithms include the genetic algorithm [42], the wasp algorithm [43], the shark algorithm [44], ant colony optimisation [45], particle swarm optimisation [9], bacterial foraging optimisation [46], cuckoo search [47], artificial bee colony search [48], the firefly algorithm [49], the bat algorithm [50], flower pollination algorithms [51], artificial plant optimisation [52], the squirrel search algorithm [53] and the mayfly optimisation algorithm [54]. Whilst many of the newly published algorithms are

typically benchmarked solely on artificial test problems, various recently published and matured algorithms have successfully been applied to real-world problems [55–66].

Due to the vast and ever-accumulating number of newly published bio-inspired algorithms in the literature, Kar [67] divided bio-inspired algorithms into four quadrants, namely the zone of theory development, zone of applications, zone of rediscovery and zone of commercialisation. *Quadrant 1* includes algorithms that are most recently published and not sufficiently studied in all dimensions. Hence, the absence of literature on such work provides potential for researchers to further improve and investigate these algorithms. Algorithms within *quadrant 2* are more mature in relation to theoretical development and have significant potential for researchers to apply them to novel areas. *Quadrant 3* encapsulates algorithms that were introduced but were overlooked or failed to attract sufficient attention from researchers. Similarly, revisiting the theoretical foundations and practical aspects of these algorithms to improve, hybridise, ensemble or apply them to novel domains may be of interest to researchers. Finally, *quadrant 4* captures algorithms that have already been extensively applied to different domains. Although it may be more challenging for researchers to find an unexplored application of these algorithms, at the same time, these algorithms have greater potential to be adopted and applied to real-world applications, as they are proven. The work presented in this paper is probably best characterized as falling within *quadrant 1*.

Many of the most competitive bio-inspired algorithms also fall under the umbrella of swarm intelligence. PSO and its variants are probably the most widely used class of swarm intelligence algorithms.

2.2. Particle Swarm Optimisation

In the canonical PSO [9,68], particles represent a solution in a D-dimensional search space, and each particle possesses three attributes, namely its position, memory of its best position so far and a velocity, as denoted by vectors $\mathbf{x}_i$, $\mathbf{pbest}_i$ and $\mathbf{v}_i$, respectively. Initially, each particle's velocity and position are randomly assigned, and subsequently, at each time step, the fitness function is employed to guide particles towards a combination of two 'exemplars', namely $\mathbf{pbest}_i$ and **gbest**, the latter corresponding to the best position known to the swarm at time t. At each time step, the velocity and position of each particle are updated using the following two equations:

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

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \mathbf{v}_i^{(t)} \quad (2)$$

where ω is an inertia weight parameter that reflects the impact of the previous velocity on the new velocity, an important factor in achieving a balance between global exploration and local exploitation [68]. In addition, c_1 and c_2 are the 'cognitive' and 'social' acceleration coefficients, where the cognitive coefficient controls the local search (guided by exemplar $\mathbf{pbest}_i$), whilst the social component controls global exploration (guided by exemplar **gbest**) and represents a kind of cooperation between the particles. The control of these two coefficients is important in performing am efficient search, as excessive values of c_1 would lead to excessive wandering of the particles and, similarly, an excessive value of c_2 would lead to premature convergence of the swarm. $\mathbf{r}_1$ and $\mathbf{r}_2$ are random D-dimensional vectors, with each component generated in the range of $[0, 1]$, powering the stochastic element of the search.

The novel search algorithms introduced in the next section build upon this basic framework. They employ heterogeneous populations (not all members following the same rules as in the canonical version) and use more complex ways of calculating the dynamic exemplars, among other developments.

As an indication of their effectiveness, PSO algorithms have been successfully applied to various practical problems in the last few decades [69–77].

As mentioned in the previous section, in common with various other bio-inspired algorithms, one of the main limitations of the canonical PSO algorithm is premature convergence that originates from loss of population diversity. PSO variants that address premature convergence and related issues tend to propose mechanisms that periodically revamp or maintain population diversity throughout the search process [14,15,78,79]. However, the development of mechanisms to maintain high population diversity while concurrently enabling rapid convergence to optimal or near-optimal solutions is an area that has only been partially explored and remains an active field of research. It is exactly this area to which the new algorithms proposed in this paper belong; they use novel mechanisms that are shown to maintain diversity and drive rapid convergence to optimal solutions.

3. Proposed Algorithms

In this section, the two novel PSO search algorithms, BEPSO and AHPSO, are explained in detail. For both, the aim was to design heterogeneous particle behaviours that maintain diversity and provide an efficient search.

3.1. BEPSO: Biased Eavesdropping PSO

In BEPSO, the bio-inspired particle behaviour model comprises the following three components: recognition, communication and bias. The recognition component refers to the particle's ability to distinguish between conspecific and heterospecific particles. The communication component refers to the implicit signal-based communication particles perform when they discover a new and better position. The bias component enables particles to build a form of perception towards each other that evolves through social experiences. This perception is used to adopt different behaviours during the search process.

Initially, particles are divided into two groups. Particles of the same group recognise each other as conspecifics and those from the other group as heterospecifics. All particles are assigned an initial random bias towards the rest of the particles in the swarm in such a fashion that any two conspecific particles may be either negatively biased or unbiased towards each other, while any two heterospecifics may be positively biased towards each other. In the first cycle of the algorithm, particle search behaviour initiates by updating velocity and position using the canonical PSO algorithm's update equations (Equations (1) and (2)). Thereafter, the more complex update rules given below take over. After the positional update, if a particle discovers a better position, in an attempt to guide conspecifics to a potentially better location, the particle communicates with the surrounding conspecific particles by transmitting a signal indicating the new location. The signaller particle intends the communication signal recipients to be solely conspecifics, but surrounding heterospecific particles eavesdrop and exploit the information in the signal (as detailed later). The signal recipients (from both conspecific and heterospecific groups) either accept or reject the information provided by the signal, considering several factors, including their bias towards the signaller particle. Before transmitting the signal, the particle determines the transmission point ($\boldsymbol{\tau}$) for the intended communication signal. $\boldsymbol{\tau}$ is a position that lies between the previous and the current position of the signaller particle and is calculated using Equation (3).

$$\boldsymbol{\tau}_i^t = \boldsymbol{x}_{signaller}^{t-1} + rand * (\boldsymbol{x}_{signaller}^{t} - \boldsymbol{x}_{signaller}^{t-1}) \tag{3}$$

where $\boldsymbol{x}_{signaller}^{t}$ and $\boldsymbol{x}_{signaller}^{t-1}$ are the current (newly discovered) and previous position of the signaller particle, respectively, and *rand* is a uniformly distributed random number in the range of $(0, 1)$. The communication signal has a radius defined by the SR parameter with minimum and maximum bounds. $SR_{max} = 0.01 * (UB - LB) * d$, and $SR_{min} = 0.001 * (UB - LB) * d$, where UB and LB are the upper and lower bounds of the given problem, respectively, and d is the dimension of the problem. The radius of the communication signal is determined individually for each signal based on the particle's fitness compared to the average fitness in the swarm. This simulates the signal's loudness; hence, the range

of the signal extends or shrinks based on the quality of the discovered position. This behaviour mimics the confidence of the particle in the quality of the discovered location to attract more conspecifics. Hence, a confident signaller particle transmits a signal with a wider influence range, while particles with less confidence in the quality of the discovered position use lower SR with the intention of transmitting a signal to fewer conspecifics, thereby minimising the potential loss of fitness in the conspecific population. The SR parameter is calculated using Equation (4).

$$SR(\psi(\boldsymbol{x}), \overline{\psi}, SR_{min}, SR_{max}) = \begin{cases} rand(SR_{min}, \frac{SR_{min}+SR_{max}}{2}) & \psi(\boldsymbol{x}) < \overline{\psi} \\ rand(\frac{SR_{min}+SR_{max}}{2}, SR_{max}) & else, \end{cases} \quad (4)$$

where $\psi(x)$ is the fitness of a signaller particle (to be maximised, in the case of function minimisation, this is inversely proportional to the function evaluation value ($f(x)$)), $\overline{\psi}$ is the average fitness in the swarm at time t and $rand(n1, n2)$ is a random number in the range of $(n1, n2)$. This ensures that fitter particles shout louder (have higher SR). The communication signal is modelled using k signal layers to mimic environmental noise and the distortion of the signal as it travels out towards the boundary of the signal range. Hence, recipient particles located in different locations relative to the signal "hear" differently distorted variants of the original signal (newly located position). Figure 1 shows a visual depiction of the communication signal with intended conspecific recipients and eavesdroppers. We mutate the signal vector k times (for k signal layers) with a non-uniform Gaussian mutation operator, starting with a small mutation (p = 0.1), and as k increases, increasing the mutation probability to trigger larger mutations, with an upper bound of 1. This ensures that the further away a particle is, the more distorted the signal it receives. Any particle whose Euclidean distance from the transmission point (τ) is less than SR is a recipient of the signal (see the algorithm pseudocode). The upper and lower bounds for p were chosen after preliminary experiments, as values outside that range were found to be ineffective. Other researchers have found that p = 0.1 is a good lower bound for the non-uniform mutation operator when it is used to increase diversity [17].

Particles (both conspecific and eavesdroppers) can accept or ignore the information provided by the communication signal, depending on their bias and the signaller particle's confidence in the newly discovered position (detailed later). The recipient particles closest to the transmission point receive the least distorted signal, and those furthest away receive the most distorted signal. This set of stochastic mechanisms–distorting the signal as it travels across the search space and placing the transmission point between the current and last location–prevents multiple particles from clustering in exactly the same location while encouraging movement towards confidently signalled better regions, helping to avoid stagnation within recipient particles.

Among animals, survival and cooperation strongly depend on bias towards others, either through genetic influence, e.g., cooperating with a conspecific for the first time, regardless of any lack of previous experience, or through social learning, where positive association plays a role. In our algorithmic model, particles are either positively biased, negatively biased or neutral (unbiased) towards each other and use their bias to decide whether to accept or reject the information the signal provides as a guide. Negative bias can be thought of as the particle's defence mechanism built over time to avert potential negative social guidance and, thus, minimise the loss of fitness due to misleading communication signals. Positive bias, on the other hand, enables particles to form an implicit cooperative relationship and accept guidance through signal calls from established "social partners" formed over time in an attempt to improve fitness. The two mechanisms complement each other and allow particles to form decentralised communication that evolves based on particles' individual social experiences. Particles' biases form over time as a result of the accumulation of consecutive positive or negative experiences resulting from the use of signal information. An experience is positive when the signal improves the fitness of the recipient particles and negative when it reduces it. The accumulator decision model from the free-response paradigm [80] is employed to enable particles to accumulate evidence

through signaller–recipient relationships. When either of the two accumulated experience variables (α (positive) and β (negative); Equations (5) and (6), respectively) reaches a threshold, a decision response is triggered to accept or reject a signal. The following update equations are used to build the bias "evidence" between a recipient and a signaller particle (assuming minimisation of the fitness function ($f(\boldsymbol{x})$)):

$$\alpha_{ji}^{t} = \begin{cases} \alpha_{ji}^{t-1} + \lambda^{t} & f(\boldsymbol{x}_j)^{t} \leq f(\boldsymbol{x}_j)^{t-1} \\ \alpha_{ji}^{t-1} & f(\boldsymbol{x}_j)^{t} > f(\boldsymbol{x}_j)^{t-1} \end{cases} \tag{5}$$

$$\beta_{ji}^{t} = \begin{cases} \beta_{ji}^{t-1} & f(\boldsymbol{x}_j)^{t} \leq f(\boldsymbol{x}_j)^{t-1} \\ \beta_{ji}^{t-1} - \lambda^{t} & f(\boldsymbol{x}_j)^{t} > f(\boldsymbol{x}_j)^{t-1} \end{cases} \tag{6}$$

where α_{ji}^{t} is the positive bias variable and β_{ji}^{t} is the negative bias response variable at time t that the jth particle (recipient) has collected for the ith particle (signaller), and λ^{t} is the accumulating factor that contributes to the bias response variables as follows:

$$\lambda^{t} = (f(\boldsymbol{x})^{t} - f(\boldsymbol{x})^{t-1})^{2} \times 0.01 \tag{7}$$

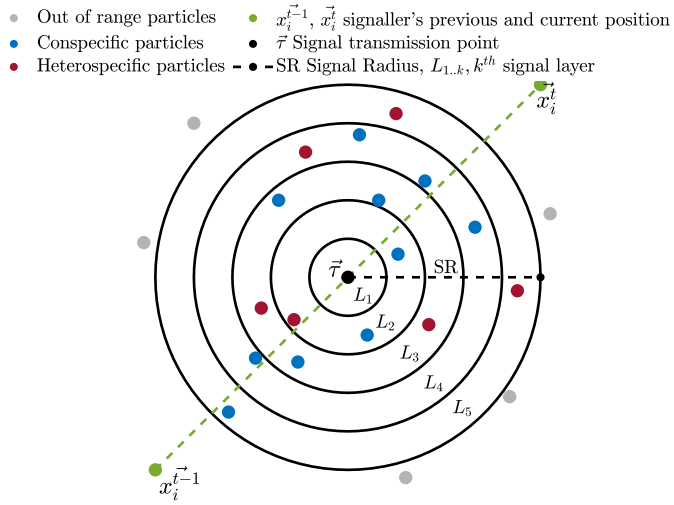

Figure 1. Visual representation of the communication signal sent by the signaller particle in the BEPSO algorithm.

If the experience is positive, α increases; if it is negative, β decreases. The multiplicative factor of 0.01 in Equation (7) was chosen after preliminary experiments conducted with a range of possible values.

Figure 2 shows a visual depiction of Equations (5) and (6) unfolding over time. Each time evidence is collected, Equation (8) is used to determine if either of the response variables (α or β) has reached the specified bias threshold value (the α threshold is positive, and the β threshold is negative).

$$Bias_{i}^{j} = \begin{cases} 1, & \alpha_{ji}^{t} \geq \eta \\ -1, & \beta_{ji}^{t} \leq -\eta \\ 0, & else \end{cases} \tag{8}$$

where $Bias_{i}^{j}$ is the jth particle's bias towards the ith particle, and the threshold (η) is an integer in the range [10, 100]. The value of η controls the pace at which particles become biased; hence, the value of η can have a direct impact on the behaviour of particles Particles tend to be rapidly biased when η is set in the lower range. On the contrary, particles can remain unbiased towards most other particles in the swarm for extended periods when η is in the higher range.

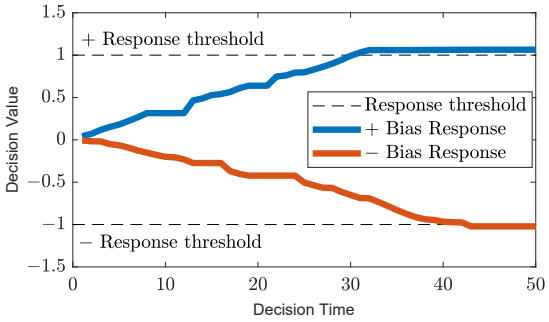

Figure 2. The accumulator decision model showing the *i*th particle's bias towards the *j*th particle at time *t*.

At the beginning of the search process, all particles are given random biases (positive (1), negative (−1) or neutral (0)) to allow for heterogeneity from the start of the search process. Since there would be no transmission of signals at $t = 1$, all particles initially update their velocity and position using the standard PSO update equations (Equations (1) and (2)). After the positional update, if the particle discovers a better position, it must transmit a communication signal to attract conspecifics to a potentially better location using the procedures described above.

To try and avoid costly 'mistakes' due to misleading information, receiving particles—both conspecifics and surrounding eavesdropping heterospecifics—decide to exploit or ignore the signal information according to a simple risk-versus-reward assessment. The following rules define the criteria both conspecifics and eavesdroppers use to exploit or ignore the signal information:

1. Conspecific recipient particles decide to exploit signal information only if the recipient particle is positively biased or unbiased towards the signaller and the signaller particle's confidence in the newly discovered position is high.
2. Eavesdropper particles decide to exploit signal information if the eavesdropper is positively or negatively biased towards the signaller but the signaller's confidence in the newly discovered position is high.

The signaller particle's confidence is high if $SR >= ((SR_{min} + SR_{max}))/2$ and low if $SR < ((SR_{min} + SR_{max}))/2)$.

In nature, animals adopt various strategies to deter eavesdroppers or make their signals less desirable or noticeable to heterospecifics. In this study, the signaller particle aims to evade eavesdroppers by adopting a probabilistic strategy whereby it occasionally deliberately uses a smaller SR value to attract fewer particles (see algorithm psuedocode for details). This behaviour enables signallers to appear less confident in the quality of the discovered position to make the signal less conspicuous for eavesdroppers. This evasion strategy mostly affects eavesdroppers, even whilst negatively biased towards the signaller particle, because they place weight on the signaller's confidence. However, it comes at a cost to conspecifics of the signaller, as it narrows the range, meaning fewer receive the signal.

Both conspecific and eavesdropper recipients that adopt the signal-based guidance use the following equation to update their velocity:

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

where $\boldsymbol{L}_k$ is the signal vector for the kth layer of the signal (the appropriate layer relative to the distance between the signaller and recipient), and the other symbols are as used in Equation (1).

Non-signal-based behaviour uses two repellent exemplars and a collaboration exemplar. The two repellent exemplars ($\boldsymbol{x}_{fur}^t$ and $\boldsymbol{x}_{OoR}^t$) are the particle located farthest from the signaller and a (randomly selected) non-recipient particle that is outside of the signal range. The two repellent exemplars are selected from the recipient's conspecific group.

The collaboration exemplar (x^t_{coll}) requires the random selection of four recipients from the other group (eavesdroppers). It is calculated as shown in Equation (10).

$$\begin{aligned} x^t_{coll} = \overline{R} = \sum_{i=1}^{2} \frac{x^{rnd}_i}{2}, \\ x^{rnd}_1 = x^{ed}_1 + rand * (x^{ed}_2 - x^{ed}_1), \\ x^{rnd}_2 = x^{ed}_3 + rand * (x^{ed}_4 - x^{ed}_3) \end{aligned} \tag{10}$$

where x^{ed}_1–x^{ed}_4 are the four randomly selected eavesdropper particles within the signal range. x^{rnd}_1 is a vector that lies between the positions of the first and second selected recipient particles, and, similarly, x^{rnd}_2 lies between the third and fourth selected recipients, as illustrated in Figure 3. x^t_{coll} is the average of the two vectors.

Particles that adopt the non-signal-based guidance update their velocities using the following equation:

$$\mathbf{v}^{t+1}_i = \omega \mathbf{v}^t_i + c^t_1 \mathbf{r}_1 (\mathbf{pbest}_i - \mathbf{x}^t_i) + c^t_2 \mathbf{r}_2 (\mathbf{S} - \mathbf{x}^t_i) \tag{11}$$

where **S** is an exemplar randomly selected as either x^t_{fur}, x^t_{OoR} or x^t_{coll}.

Imitation is one of the most common social learning behaviours that animals adopt. In our algorithmic model, the unbiased recipients imitate the dominating behaviour of their conspecifics. Hence, if p particles of the conspecifics or eavesdroppers adopt the signal-based guidance while q of them adopt the non-signal-based guidance and $p > q$, then the unbiased conspecifics/eavesdroppers imitate the behaviour dominantly adopted by their conspecifics. When $p = q$ or unbiased particles dominate one or both groups, signal-based or non-signal-based behaviour is randomly adopted by the unbiased recipient particles.

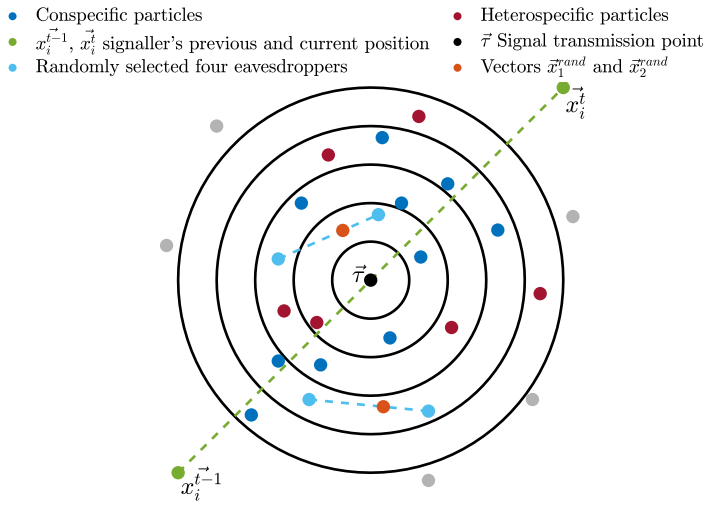

Figure 3. Visual depiction of the particles selected for the collaboration exemplar.

The heterogeneity in the swarm is formed by the mix of signal-based and non-signal-based behaviour adopted by recipient particles. The particles' biases formed over time maintain the balance of particles adopting these behaviours. The entitlement of particles as recipients depends on several factors, including the previous position, the position discovered position by the signaller, the SR value and the calculated transmission point. Hence, a small fluctuation in one of these factors significantly alters the list of potential recipients of both conspecifics and eavesdroppers at time t. Consequently, which set of particles become recipients is unpredictable for each transmitted signal. This unpredictable yet self-organising behaviour is a further support for population diversity, minimising the risk of particles being stuck at local optima.

In order to fully exploit existing potential solutions, the BEPSO model also incorporates the periodic use of multi-swarms, as introduced in [29]. Every so often, the swarm is split into multiple swarms, which supports another phase of search, after which they join back together, the standard BEPSO mechanisms resume and the cycle repeats.

The initiation of the multi-swarm mechanism requires the division of the swarm into N subswarms. Instead of randomly splitting the swarm into N equal subswarms, in our method, $N = 3$ subswarms are formed based on particles' dominating biases to enable each subswarm to possess an asymmetrical and self-regulating population. Hence, a particle is a member of $subswarm_1$ if it is predominantly positively biased towards most particles in the swarm. Similarly, if a particle is mostly negatively biased or unbiased, then the particle belongs to the corresponding groups ($subswarm_2$ and $subswarm_3$). Each member of a subswarm uses the following equation to update its velocity:

$$\mathbf{v}_i^{t+1} = \omega \mathbf{v}_i^t + c_1^t \mathbf{r}_1 (\mathbf{pbest}_i - \mathbf{x}_i^t) + c_2^t \mathbf{r}_2 (\mathbf{lbest}_k - \mathbf{x}_i^t) \quad (12)$$

where $\mathbf{lbest}_k$ is the position of the fittest particle in the k^{th} subswarm.

3.2. BEPSO Summary

The BEPSO algorithm uses a heterogeneous population that is split into two groups. Members of a group recognise each other as conspecifics and members of the other group as heterospecifics. When a particle finds a better position (solution), it generates a signal to guide conspecifics towards the new solution. The signal range is proportional to the quality of the solution, and the signal is distorted as it travels through space. The transmission point of the signal is randomly chosen between the last and the current position of the transmitting particle. Heterospecific particles in range can eavesdrop on the signal. Particles have biases towards each other, which determine whether or not a particle decides to adopt or ignore the signal information based on an accumulator decision model. These interacting stochastic mechanisms determine the social exemplars for particles in their movement update equations.

Particles' biases build over time, ensuring behavioural heterogeneity by forcing particles to dynamically adopt signal-based and non-signal-based guidance. This primary search strategy is supported by periodically activated subswarm searches to efficiently exploit the existing solutions. The asymmetrical populations in the subswarms exploit different local solutions with varying densities of particles in different search phases, resulting in efficient search behaviour with a good balance of exploration and exploitation. The various interacting mechanisms maintain diversity and prevent premature convergence while allowing for rapid, efficient movement towards optimal and near-optimal solutions. The overall BEPSO algorithm is shown in Algorithm 1.

BEPSO Parameters

The BEPSO algorithm involves a number of parameters, but our intention was to develop a technique that does not need tuning for each new problem it is applied to. Hence, after extensive preliminary parameter investigations involving a wide range of problems and problem types and sizes, the following set of parameters was found to be robust and effective and was adopted for all subsequent experiments reported in this paper. Note that, in common with many current PSO algorithms, the momentum term (ω) is adaptive and time-varying, as detailed in Algorithm 1. Full details of the extensive parameter investigations can be found in the Supplementary Material.

The population size = 40, the length of the two phases of the search (signalling and multi-swarm) = 10 iterations, SR_{max} = 0.01, SR_{min} = 0.001, and bias threshold = 20 (for other parameters, see Algorithm 1).

Algorithm 1: BEPSO

INPUT: swarm size, n, problem dimension, d, maximum number of iterations, T_{max}
OUTPUT: $gbest$
Define initial values for particles' velocity $\vec{v}$ and position $\vec{x}$;
initialise $\vec{guidance}_{1...n}$ `// guidance vector for particles`
$status = (0, 0, 0.., n)$; `// guidance status as 0 or 1`
$C = 0.15, \omega_{max} = 0.99, \omega_{min} = 0.2$;
$c_1 = 2.5 - (1 : T_{max}) * 2/T_{max}$
$c_2 = 0.5 - (1 : T_{max}) * 2/T_{max}$
$\omega_1 = \frac{\omega max + (\omega_{min} - \omega max)}{1 + exp(-5(\frac{2t}{T_{max}} - 1))}$;
$SR_{max} = 0.05 * (UB - LB) * d$;
$SR_{min} = 0.01 * (UB - LB) * d$;
$counter = 0$;
for *t=1:*T_{max} **do**
  if *mod(t,100)==0* **then**
    $phase_{multi-swarm} = true$;
  end
  if $phase_{multi-swarm} == true$ **then**
    $counter = counter + 1$
    if $counter == 100$ **then**
      $phase_{multi-swarm} = false$;
      $counter = 0$
    end
  end
  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 $phase_{subswarm} == true$ **then**
      update $\vec{v}_i$ and $\vec{x}_i$ using Equations (12) and (2)
    else
      if $status_i == 0$ **then**
        update $\vec{v}_i$ and $\vec{x}_i$ using Equations (1) and (2) `// at start, no signal info`
      else
        update $\vec{v}_i$ and $\vec{x}_i$ using Equations (1) and (2) (but use $\vec{guidance}_i$ instead of the $\vec{gbest}$ exemplar)
        Evaluate the fitness of $\vec{x}_i^t$
        Update particle's bias using Equations (5)–(8)
      end
      if $f(\vec{x}_i^{t+1}) < f(\vec{x}_i^t)$ **then**
        Calculate $\vec{\tau}$ using Equation (3)
        if *randi([0 1])==0* **then**
          `// 50/50 chance of disguising SR`
          $SR = \mathrm{U}(SR_{min}, \frac{SR_{min} + SRmax}{2})$
        else
          Calculate SR using Equation (4)
        end
        for *j=1:n* **do**
          if $norm(\vec{\tau} - \vec{x}_j^t) < SR$ **then**
            if j^{th} *particle is a conspecific* **then**
              if j^{th} *particle qualifies for signal-based guidance* **then**
                calculate $\vec{guidance}_j$ using Equations (9) and (2)
              else
                calculate $\vec{guidance}_j$ using Equations (11) and (2)
              end
              $status_j = 1$;
            else if j^{th} *particle is a heterospecific* **then**
              if j^{th} *particle qualifies for signal-based guidance* **then**
                calculate $\vec{guidance}_j$ using Equations (9) and (2)
              else
                calculate $\vec{guidance}_j$ using Equations (11) and (2)
              end
              $status_j = 1$;
            end
          end
        end
      end
    end
  end
end

3.3. AHPSO: Altruistic Heterogeneous PSO

In this section, we describe our second novel PSO algorithm inspired by a certain kind of altruistic animal behaviour, a direction that has not been explored before.

The AHPSO algorithm incorporates conditional altruistic behaviour and a heterogeneous particle model that delays the loss of population diversity and prevents premature convergence, resulting in a highly effective search algorithm. In our approach, particles are conceptualised as energy-driven entities with two possible states, namely active and inactive. Particles have a current energy level and an activation threshold and are inclined to be active by maintaining their current energy level above the threshold. The distinction in a particle's state is used to create a heterogeneous population, and particles' tendency to become active is aided by lending–borrowing relationships among them. Hence, conditional altruistic behaviour is exhibited by particles by lending energy to inactive particles to allow them to change their state. In doing so, helper/lender particles risk downgrading their own state from active to inactive. To minimise the risk of reducing their own fitness, a group of lenders assesses the situation of the beneficiary/borrower particle based on the level of altruistic behaviour it has exhibited before making a decision as to whether or not to lend. In our behavioural model, heterogeneity is attained through altruism, and better population diversity is achieved through heterogeneity. Hence, these two concepts depend on, feed and maintain each other.

In AHPSO, two behaviour models are used. The first is the altruistic particle model, which constitutes the particles' primary behavioural model. The second, the paired particle model, extends the former to boost population diversity further. They are applied one after the other in each overall cycle of the AHPSO algorithm.

3.4. Altruistic Particle Model (APM)

The activation status of particles is dependent on their current energy level ($E_{current}$) and activation threshold ($E_{activation}$). Initially, both values are randomly assigned. The concept of activation is employed to determine the type of movement strategy for particles at the individual level and, as a result, controls behavioural heterogeneity in the swarm.

Particles have an inherent tendency to be active; hence, particles in the inactive state are expected to borrow energy from other particles when their $E_{current} < E_{activation}$. The main factor influencing and maintaining swarm heterogeneity is the particles' altruistic behaviour. A particle that behaves altruistically by making significant energy contributions to other swarm members is highly unlikely to be rejected when in need of energy itself, and on the contrary, particles that exhibit lower altruistic behaviour are inclined to be rejected.

Persistent borrowing behaviour in a particle over prolonged periods results in a highly unstable lending–borrowing ratio and reduces the altruism value (A_i) of the particle (as the particle consumes a lot more resources than it contributes to the swarm). A_i is calculated according to Equation (13).

$$A_i = \frac{L_i^t}{B_i^t} \tag{13}$$

where L_i^t and B_i^t are the number of times the ith particles lent and borrowed energy, respectively, up to time t. When a particle is unable to activate, it attempts to borrow energy from randomly selected potential lenders, and in order to lend energy, potential lender particles expect the energy-requesting particle to meet an altruism criterion defined by ϕ (Equation (14)).

$$\phi = P_1 \times P_2 \tag{14}$$

This criterion is based on the independent probability of two events (P_1 and P_2). For P_1, the altruism value of the borrower particle (A_i) is used, and P_2 is calculated as

$$P_2 = \frac{\delta}{N} \tag{15}$$

where δ is the number of particles in the swarm with active status at time t, and N is the population size.

P_2 gives a rough measure of the probability that a lender particle will return the energy lent by the swarm. In addition to enforcing altruistic behaviour, the criterion (ϕ) provides a form of altruistic assessment of lender particles' probability of returning lent energy.

Potential lender particles use the γ value described by Equation (16) to inform the final decision to either lend energy or reject the request of the borrower particle.

$$\gamma(\phi, \beta) = \begin{cases} false, & \text{if } \phi < \beta \\ true, & \text{if } \phi \geq \beta \end{cases} \tag{16}$$

where β is the average altruism value in the swarm.

If the decision (γ) is in favour of the energy-requesting particle (i.e., true), an equal amount of energy is borrowed from each lender to compensate for the required energy of the borrower particle. This is calculated as

$$E_{required} = \frac{E_{activation} - E_{current}}{\alpha} \tag{17}$$

where $E_{required}$ is the amount of energy required from each lender and α is the number of selected lenders. The movement strategy adopted by particles in the altruistic behaviour model is based on the altruistic traits of particles. Particles that are active use the canonical PSO update equation shown in Equations (1) and (2), whereas inactive particles who do not meet the criterion (ϕ) and, therefore, cannot borrow use Equation (18) to update their velocity (and position via Equation (2)).

$$\boldsymbol{v}_i^{t+1} = \omega \boldsymbol{v}_i^{(t} + c_1 \boldsymbol{r}_1 (\boldsymbol{pbest}_i - \boldsymbol{x}_i^t) + c_2 \boldsymbol{r}_2 (\boldsymbol{pbest}_{minA}^t - \boldsymbol{x}_i^t) \tag{18}$$

where $\boldsymbol{pbest}_{minA}^{(t)}$ is the personal best position of the least altruistic particle at time t. In the AHPSO framework, particles that do not meet the criterion (ϕ) are less altruistic at time t and, hence, behave together with similarly less altruist particles. Considering the evolving dynamics of the altruistic model, the least and most altruistic particles fluctuate. Hence, guidance towards the least altruistic particle partially enables cooperation through altruism and supports heterogeneity. Energy sharing takes place between the lender particles and the borrower who meets the criterion (ϕ). As lenders are randomly selected without any criteria, there is a distinct possibility of some lenders not having excess energy to lend. Therefore, after borrowing energy, the borrower particle may still lack sufficient energy to activate. In this case, an exemplar for the particle is generated by the mean position of half of the lender particles, and their velocities are updated according to Equation (19).

$$\boldsymbol{v}_i^{t+1} = \omega \boldsymbol{v}_i^{(t} + c_1 \boldsymbol{r}_1 (\boldsymbol{pbest}_i - \boldsymbol{x}_i^t) + c_2 \boldsymbol{r}_2 (\boldsymbol{x}_{mean} - \boldsymbol{x}_i^t) \tag{19}$$

where $\boldsymbol{x}_{mean}$ is the mean position of the randomly selected $\lceil \frac{\alpha}{2} \rceil$ 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 establish implicit communication between lender and borrower particles.

If, however, the borrower particle succeeds in borrowing sufficient energy to activate, the particle's velocity is calculated using Equation (20).

$$\boldsymbol{v}_i^{t+1} = \omega \boldsymbol{v}_i^{(t} + c_1 \boldsymbol{r}_1 (\boldsymbol{pbest}_i - \boldsymbol{x}_i^t) + c_2 \boldsymbol{r}_2 (\boldsymbol{pbest}_{maxA}^t - \boldsymbol{x}_i^t) \tag{20}$$

where $\boldsymbol{pbest}_{maxA}^{(t)}$ is the personal best position of the most altruistic particle at time t. The ith particle is guided towards the most altruistic particle to establish partial cooperation and maintain heterogeneity. An additional stochastic element is introduced by randomly reinitiating $E_{current}$ and $E_{activation}$ for the entire swarm at specific intervals. The idea behind this is to fluctuate the altruism value of particles and allow less altruistic particles at time t to

cooperate, contribute and evolve as altruistic particles. In contrast, an altruistic particle could "devolve" and exhibit selfish behaviour. As a result, this model allows altruistic and selfish particles to adopt distinct movement strategies that change and adapt depending on the level of a particle's "evolution", leading to an adaptive and heterogeneous particle population.

3.5. Paired Particle Model (PPM)

The paired particle model is an extension of the altruistic behaviour model described in the previous section. The purpose of the PPM is to further boost the heterogeneity properties of the algorithm, leading to increased population diversity. The PPM is run after the APM in each overall iteration of AHPSO. A relatively small proportion of the population is used for the paired particle model (see Section 3.6 for values). This model employs two movement strategies for the selected particles, namely a coupling-based strategy and an opposition-based strategy; each pair randomly selects which to use (see Algorithm 2). The paired particle model enables particles to randomly form and maintain pair-style bonds similar to the mechanism employed in [81]. An altruistic particle may abandon its pair if the pair is less altruistic than the swarm's average.

3.5.1. Coupling-Based Strategy

The coupling-based strategy distinguishes pairs as tightly or loosely coupled or neutral, which determines the type of movement strategy. The following rules govern the type of coupling relationship paired particles adopt:

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

Tightly coupled paired particles tend to have more influence on each other than loosely coupled pairs. Tightly and loosely coupled particles update their velocities using Equation (21) and Equation (22), respectively.

$$\boldsymbol{v}_i^{t+1} = \omega \boldsymbol{v}_i^{(t} + c_1 \boldsymbol{r}_1((\boldsymbol{x}_{pair}^i \times E_{current}^i) - \boldsymbol{x}_i^t) + c_2 \boldsymbol{r}_2((\boldsymbol{pbest}_{pair}^i \times E_{current}^i) - \boldsymbol{x}_i^t) \quad (21)$$

where $\boldsymbol{x}_{pair}^i$ and $\boldsymbol{pbest}_{pair}^i$ are the ith particle's pair position and personal best position, respectively.

$$\boldsymbol{v}_i^{t+1} = \omega \boldsymbol{v}_i^{(t} + c_1 \boldsymbol{r}_1((\boldsymbol{pbest}_{pair}^i \times E_{current}^i) - \boldsymbol{x}_i^t) + c_2 \boldsymbol{r}_2((\boldsymbol{x}_{pair}^i \times E_{current}^i) - \boldsymbol{x}_i^t) \quad (22)$$

$E_{current}^i$ is used as a damping factor to prevent the possibility of particles rapidly oscillating, instead performing small movements in this secondary phase of the search. In essence, the coupling-based strategy empowers particles within the paired behaviour model to influence each other, regardless of any distance constraints between pairs. Therefore, clustered particles take small steps towards their pair, depending on the type of coupling relationship formed, causing perturbations in the current position without an explicit impact on $\boldsymbol{pbest}$. However, these fluctuations in particle position subsequently influence the next position of the particle, helping to escape local optima.

3.5.2. Opposition-Based Strategy

The opposition-based movement strategy guides paired particles towards exemplars with opposite features. By guiding both particles of a pair in potentially distinct directions, the strategy aims to maintain diversity within such pairs and, hence, within a proportion of the population. In a way, this movement strategy partly compensates for the limitations of previously described coupling-based strategy, where pairs influence each other. The opposition-based strategy aims to slow down learning between pairs without destroying it and delays the loss of diversity between pairs by guiding both in the direction of distinct

exemplars. The altruism value of the paired particles is used as the determining factor to distinguish the type of movement a particle performs. Exemplar selection for members of paired particles works as follows. If the ith particle is more altruistic than its coupled pair, its velocity is updated using Equation (23).

$$\boldsymbol{v}_i^{t+1} = \omega \boldsymbol{v}_i^t + c_1 \boldsymbol{r}_1 (\boldsymbol{pbest}_i - \boldsymbol{x}_i^t) + c_2 \boldsymbol{r}_2 (\boldsymbol{x}_{maxA}^{pair} - \boldsymbol{x}_i^t) \tag{23}$$

where $\boldsymbol{x}_{maxA}^{pair}$ is randomly selected as either $\boldsymbol{pbest}$ or the position of the most altruistic individual of the most altruistic pair at time t.

But if the ith particle is less altruistic than its pair, its velocity is updated using Equation (24).

$$\boldsymbol{v}_i^{t+1} = \omega \boldsymbol{v}_i^t + c_1 \boldsymbol{r}_1 (\boldsymbol{pbest}_i - \boldsymbol{x}_i^t) + c_2 \boldsymbol{r}_2 (\boldsymbol{x}_{minA}^{pair} - \boldsymbol{x}_i^t) \tag{24}$$

where x_{minA}^{pair} is randomly selected as either $\boldsymbol{pbest}$ or the position of the least altruistic individual of the least altruistic pair at time t.

Since both movement strategies in the paired particle model always result in particles moving, they act as a stabilising mechanism that enables particles to partially escape from local optima and continue the search process. In both coupling-based and opposition-based learning, the fitness of the exemplar particles is deliberately not considered; this helps to minimise particle clustering around local optima, aiming to maintain diversity and, hence, guard against premature convergence.

The overall AHPSO algorithm is shown in Algorithm 2; note that an adaptive, time-varying momentum term (ω) is employed.

3.6. AHPSO Summary

The AHPSO algorithm uses a heterogeneous population in which a dynamic energy-based ecosystem develops. Particles are either active or inactive, depending on their energy level. All particles have an inherent drive to become active and, when inactive, attempt to borrow energy from other (randomly selected) particles in order to reach the activation threshold. The altruism of a particle develops over time, depending on its lending and borrowing behaviour. Particles use a model based on the altruism level of a potential borrower in order to decide whether or not to lend—a form of conditional altruism. Particles' movement strategies depend on their levels of activation and altruism, which are controlled by social exemplars generated by interacting stochastic rules. The algorithm uses two search phases, in which different movement rules apply.

The two phases of AHPSO, which are executed consecutively in each cycle, namely the altruistic and paired particle models, complement each other. The search process is initiated with the APM, during which particles attempt to change their state from inactive to active. Behaviourally, active particles tend to be more focused on exploitation. On the contrary, inactive particles, attempting to borrow energy, are more focused on exploration and are mainly influenced by the most and least altruistic particles at time t. The level of altruistic behaviour exhibited by particles varies a great deal, and particles evolve frequently from more to less or less to more altruistic. Hence, the different types of particles (active, successful borrowers and unsuccessful borrowers) are guided by highly diverse and rapidly evolving exemplars, enabling efficient search behaviour while maintaining population diversity.

Next, the second behaviour model (PPM) takes over. Unlike the main APM, this is used for only a selected proportion of the population. The main purpose of the PPM is to further increase heterogeneity and prevent stagnation of particles for the selected proportion of the population. Because energy and altruism levels play a part in the movement strategies, changes in the values of $E_{current}$ and A_i shape particles' behavioural patterns and result in stochastic switching between different types of movements, leading to diverse behaviour and an evolution of the strategy. The result, as with BEPSO, is a highly effective balance between exploration and exploitation.

Algorithm 2: AHPSO

INPUT: swarm size, n, max number iteration, T_{max}
OUTPUT: *gbest*
$C = 0.15, \omega_{max} = 0.99, \omega_{min} = 0.2;$
$c_1 = 2.5 - (1 : T_{max}) * 2/T_{max}$
$c_2 = 0.5 - (1 : T_{max}) * 2/T_{max}$
$\omega_1^t = \frac{\omega_{max} + (\omega_{min} - \omega_{max})}{1 + exp(-5(\frac{2t}{T_{max}} - 1))}$
$E_{current}^{1..n} = U(0.1, 1, [1\ n])$
$E_{activation}^{1..n} = U(0.5, 1, [1\ n])$
$\alpha = \lfloor n * 0.2 \rfloor$
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**
      `// APM phase`
      update $\vec{v}_i$ and $\vec{x}_i$ using Equations (1) and (2)
    else
      calculate A_i^t, P_2 and ϕ using Equations (13)–(15)
      if $\phi < \beta$ **then**
        update $\vec{v}_i$ and $\vec{x}_i$ using Equations (18) and (2)
      else
        randomly select α potential lenders
        calculate $E_{required}$ using Equation (17)
        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 Equations (20) and (2)
        else
          update $\vec{v}_i$ and $\vec{x}_i$ using Equations (19) and (2)
        end
      end
    end
    if i^{th} *particle is paired* **then**
      `// PPM phase`
      if *randi([0 1])==0* **then**
        `// randomly (50/50) chose strategy`
        if *pair is tightly coupled* **then**
          `// coupling based`
          update $\vec{v}_i$ and $\vec{x}_i$ using Equations (21) and (2)
        else if *pair is loosely coupled* **then**
          update $\vec{v}_i$ and $\vec{x}_i$ using Equations (22) and (2)
        end
      else
        if $A_i > A_{couple}$ **then**
          `// opposition based`
          update $\vec{v}_i$ and $\vec{x}_i$ using Equations (23) and (2)
        else
          update $\vec{v}_i$ and $\vec{x}_i$ using Equations (24) 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

AHPSO Parameters

The AHPSO algorithm involves a number of parameters, but, like with BEPSO, our intention was to develop a technique that works very well across a wide range of problems and problem sizes with a single general set of parameters. Hence, after extensive preliminary parameter investigations, the following robust set of parameters was found to be highly effective and was adopted for all subsequent experiments reported in this paper. Full details of the extensive parameter investigations can be found in the Supplementary Material.

The population size = 60, α is randomly set in the range of $[10, 18]$ each time it is used, the period after which the lender and borrower profiles of the swarm are reset is set to LB_{rate} = 10 in order to avoid stagnation, the period after which energy and energy activation values are reinitialised is ER = 5 and the paired population size = 6 (see Algorithm 2 for other details). The preliminary investigations also established that employing the secondary PPM phase of the search had a significantly positive impact.

4. Experimental Method

The performance of the new BEPSO and AHPSO algorithms was verified across multiple dimensions (30, 50 and 100) of the widely used CEC'13 [82], CEC'14 [83] and CEC'17 [84] benchmark test suites, along with various constrained real-world problems. A thorough comparison was conducted against 13 well-known state-of-the-art PSO variants; a recent bio-inspired metaheuristic (I-CPA); and the 2014 CEC competition winner, L-SHADE (a powerful differential evolution algorithm). Each of the comparator algorithms used the best published general parameter set.

The CEC test suites comprise unconstrained single-objective benchmark problems of various classes, including unimodal, multimodal, hybrid and composition functions. The CEC'13 suite comprises a total of 28 functions, namely 5 unimodal, 15 multimodal and 8 composition functions. The CEC'14 suite comprises 30 functions, namely 3 unimodal, 13 multimodal, 6 hybrid and 8 composition functions. The CEC'17 suite comprises 29 functions, namely 1 unimodal, 7 multimodal, 10 hybrid and 11 composition functions. Overall, a total of 87 unconstrained benchmark functions were used to evaluate the performance of the algorithms, each at three different problem dimensions. These test suites are widely regarded as suitably challenging, enabling thorough evaluation of search algorithms. The evaluation process of each test suite was carried out according to the evaluation criteria set out by the official CEC competitions [84].

The algorithms were also tested on the 14 non-convex constrained real-world problems [85] listed in Table 1.

In order to produce statistically robust results, each algorithm was run 30 times on each test problem. For the CEC test suites, the maximum number of function evaluations per problem was $10^4 \times d$, where d is the problem dimension.

For the constrained problems, the maximum number of function evaluations for each problem (Max_{FEs}) was determined using the following criteria:

$$Max_{FEs} = \begin{cases} 1 \times 10^5, & D \leq 10 \\ 2 \times 10^5, & 10 < D \leq 30 \\ 4 \times 10^5, & 30 < D \leq 50 \end{cases} \quad (25)$$

where D is the dimension of the problem. A penalty method, as defined in [86,87], was used to convert the constrained evaluation functions to unconstrained evaluation functions (adding penalties proportional to constraint violations). The method is defined by Equations (26) and (27), assuming function minimisation.

$$F(x) = f(x) + H(x) \quad (26)$$

$$H(x) = \omega_1 \delta + \omega_2 \sigma \quad (27)$$

where ω_1 and ω_2 are static weights (ω_1, ω_1 = 100), δ is the number of violated constraints and σ is the sum of all violated constraints.

The full set of 15 comparator algorithms and their key parameters, as used in this study, are shown in Table 2. The same set of general algorithm parameters was used for both the unconstrained and constrained test suites.

Table 1. Details of the 14 constrained real-world problems. D is the number of decision variables; g and h are the numbers of inequality and equality constraints, respectively; and $f(\mathbf{x}^*)$ is the best known objective function value.

	Problem	D	g	h	$f(x^*)$
	Process Synthesis and Design Problems				
RC01	Process flow sheeting problem	3	3	0	1.0765430833
RC02	Process synthesis problem	7	9	0	2.9248305537
	Mechanical Engineering Problems				
RC03	Weight minimisation of a speed reducer	7	11	0	2.9944244658×10^3
RC04	Pressure vessel design	4	4	0	5.8853327736×10^3
RC05	Three-bar truss design problem	2	3	0	2.6389584338×10^2
RC06	Step-cone pulley problem	5	8	3	16.069868725
RC07	10-bar truss design	10	3	0	$5.2445076066E \times 10^2$
RC08	Rolling element bearing	10	9	0	1.4614135715×10^4
RC09	Gas transmission compressor design	4	1	0	2.9648954173×10^6
RC10	Gear train design	4	1	1	0.0000000000
	Power Electronic Problems				
RC11	SOPWM for 7-level inverters	25	24	1	$1.5164538375 \times 10^{-2}$
RC12	SOPWM for 8-level inverters	30	29	1	$1.6787535766 \times 10^{-2}$
RC13	SOPWM for 11-level inverters	30	29	1	$9.3118741800 \times 10^{-3}$
RC14	SOPWM for 13-level inverters	30	29	1	$1.5096451396 \times 10^{-2}$

Table 2. The comparator algorithms used in the detailed investigations of BEPSO and AHPSO, along with their key parameter values.

Key	Algorithm	Parameters
L-Shade [88]	SHADE with linear population reduction	$N^{init} = round(D \times r^{N^{init}}), \lvert A \rvert = round(N^{init} \times r^{arch})$, $r^{arch} = 2.6, p = 0.11, H = 6$
BBPSO [89]	Bare-bones PSO	$\phi = 4.1, \lambda = 0.729, c_1, c_2 = 2.05, r_1, r_2\ U(0,1)$
BreedingPSO [90]	A GA/PSO hybrid	w = 0.8–0.6, $c_1, c_2 = 1.49445$ $V_{max} = 0.15 * Range$
HCLPSO [23]	Heterogeneous comprehensive learning PSO	w = 0.99–0.29, $c_1 = 2.5 - 0.5, c_2 = 0.5 - 2.5, K : 3 - 1.5, V_{max} = 0.5 * Range$
CLPSO [22]	Comprehensive learning PSO	w = 0.9–0.2; $c_1, c_2 = 1.49445, V_{max} = 0.2 * Range$
FIPS [91]	Fully informed PSO	$\chi = 0.729, \sum c_i = 4.1$
FDR-PSO [92]	Fitness distance ratio PSO	w = 0.9–0.4, $c_1 = c_2 = 1, c = 2, V_{max} = 0.2 * Range$
UPSO [93]	Unified PSO	$\chi = 0.729, c_1, c_2 = 2.05, NR = 1$
EPSO [94]	Ensemble PSO	$w = 0.9 \rightarrow 0.2, w_1 = 0.9 \rightarrow 0.4, c_1 = 3 \rightarrow 1.5$, $c_{2_1} = 2.5 \rightarrow 0.5, c_{2_2} = 0.5 \rightarrow 2.5, c_{3_1} = 2.5 \rightarrow 0.5$, $c_{3_2} = 0.5 \rightarrow 2.5, c_{4_1} = 2.5 \rightarrow 0.5, c_{4_2} = 0.5 \rightarrow 2.5$, $Pc = 0.5, n_{size} = 3$
DMS-PSO [29]	Dynamic multi-swarm PSO	$w = 0.729, c_1 = c_2 = 1.49445, m = 3; R = 15$; $V_{max} = 0.5 * Range$
HPSO-TVAC [95]	Hierarchical PSO with time-varying coefficients	$c_1 = 2.5 - 0.5, c_2 = 0.5 - 2.5, V_{max} = 0.5 * Range$
LPSO [96]	Linearly decreasing inertia weight PSO	$w = 0.9 - 0.4; c_1, c_2 = 1.49445$
maPSO [97]	Macroscopic PSO	$w = 0.9 - 0.4, c = 1.49445$

Table 2. *Cont.*

Key	Algorithm	Parameters
miPSO [97]	Microscopic PSO	$w = 0.9 - 0.4, c = 1.49445$
I-CPA [98]	Improved carnivorous plant alg	$nCPlant = 2, nPrey = 8, n = nCPlant + nPrey$

Computational Complexity

The metric proposed in [82] was used to calculate computational complexity (see Section 6) using the following steps (originally designed for the CEC13 suite) for each required dimension:

Step 1—Calculate the given code (according to the methodology proposed in [82]) and record the computation time as T_0.
Step 2—Calculate the computation time just for F_{14} (CEC13 test suite) for 20×10^4 function evaluations on dimension D and record the results as T_1.
Step 3—Calculate the complete algorithm computation time for F_{14} with 20×10^4 function evaluations on the same dimension as T_2.
Step 4—Repeat step 3 5 times and attain 5 individual T_2 values ($\overline{T_2} = mean(T_2)$).

Finally, the time complexity (T_c) is calculated as $T_c = \overline{T_2} - (T_1/T_0)$.

5. Results

This section presents the results of detailed comparative investigations of the efficacy of BEPSO and AHPSO using the methodology outlined in the previous section. All results are based on the mean of 30 runs. See the Data Availability Statement at the end of this paper for details of access to raw results data, including all convergence graphs.

5.1. BEPSO: Performance

Figures 4 and 5 illustrate the performance of BEPSO relative to the comparator algorithms on first test suite (CEC'13) at dimensions of 30, 50 and 100. The height of the bars show how many test functions from the suite the algorithm found the best solution to (averaged over 30 runs), that is, the best solution found among all algorithms. Sometimes, multiple algorithms find the same best solution for a test function, and in other cases, they fin just one. Figure 4 compares all algorithms (including L-SHADE), and Figure 5 compares all PSO variants.

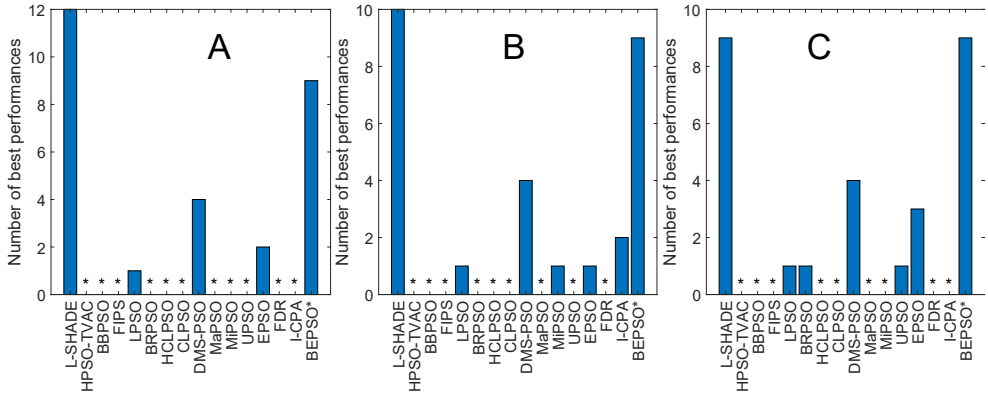

Figure 4. BEPSO comparison. The total number of best performances achieved by each algorithm with respect to mean error values (relative to best known/found function values) on the **CEC'13** problems. (**A**) 30 dimensions; (**B**) 50 dimensions; (**C**) 100 dimensions. * No best performances achieved.

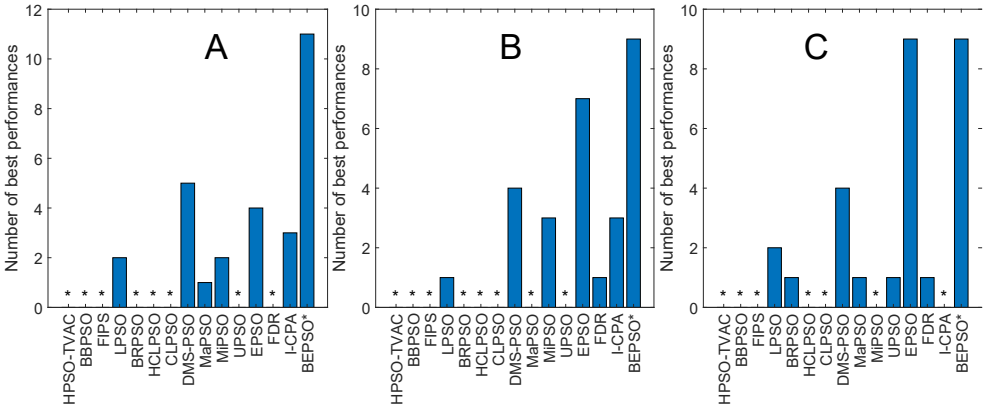

Figure 5. BEPSO comparison. The total number of best performances achieved by each PSO algorithm with respect to mean error values on the **CEC'13** problems. (**A**) 30 dimensions; (**B**) 50 dimensions; (**C**) 100 dimensions. * No best performances achieved.

Two things are clear from these bar charts of performance on the CEC'13 test suite. BEPSO is highly competitive relative to all other comparator PSO algorithms, dominating all of them at 30-D and 50-D in terms of the number of best solutions found and all but one (EPSO, which is equal) at 100-D; it is also highly competitive in comparison to the powerful differential evolution L-SHADE algorithm, with its performance relative to L-SHADE increasing as the dimension of the problem increases (equal at 100-D).

Table 3 summarises a detailed statistical analysis of the comparative experiments across all runs at each of the three dimensions in terms of the quality of the found solutions. Pairwise statistical difference tests between BEPSO and the comparison algorithms were carried out using the non-parametric Wilcoxon signed-rank test with a significance level of 5% and appropriate adjustments for multiple comparisons [99]. The (+) symbol is used to denote the algorithms over which BEPSO exhibited statistically significantly better performance, (=) indicates no statistically significant difference in the mean performance and (−) marks the comparison algorithms whose performance is statistically significantly better than that of BEPSO. The table shows us that BEPSO's performance on the CEC'13 test suite is significantly better than that of 10 of the 15 comparator algorithms across all dimensions, with no significant difference for the other 5. This means that none of the comparator algorithms were statistically significantly better than BEPSO on any the dimensions.

Table 3. BEPSO statistical tests. Wilcoxon signed-rank test results with a significance level of $p = 0.05$ for **CEC'13** problems at 3 different dimensions. +: statistically significantly better; =: no significant difference; −: statistically significantly worse.

BEPSO versus (CEC13)																
Dimension	**L-SHADE**	**HPSO-TVAC**	**BBPSO**	**FIPS**	**LPSO**	**BRPSO**	**HCLPSO**	**CLPSO**	**DMS-PSO**	**MaPSO**	**MiPSO**	**UPSO**	**EPSO**	**FDR**	**I-CPA**	**+/=/−**
30	=	+	+	+	+	+	=	+	=	+	+	+	=	=	+	
p	0.19	6.3×10^{-4}	3.7×10^{-6}	1.6×10^{-5}	0.015	9.2×10^{-6}	0.6	4.8×10^{-6}	0.45	0.001	0.0027	9.3×10^{-6}	0.6	0.09	0.014	10/5/0
50	=	+	+	+	+	+	=	+	=	+	+	+	=	=	+	
p	0.19	6.3×10^{-4}	3.7×10^{-6}	1.6×10^{-5}	0.015	9.2×10^{-6}	0.6	4.8×10^{-6}	0.45	0.001	0.0027	9.3×10^{-6}	0.6	0.09	0.014	10/5/0
100	=	+	+	+	+	+	=	+	=	+	+	+	=	=	+	
p	0.19	6.3×10^{-4}	3.7×10^{-6}	1.6×10^{-5}	0.015	9.2×10^{-6}	0.6	4.8×10^{-6}	0.45	0.001	0.0027	9.3×10^{-6}	0.6	0.09	0.014	10/5/0

BEPSO performed particularly well on the multimodal and composition test functions of this suite, which are generally regarded as the hardest problems.

Figures 6 and 7 and Table 4 illustrate the corresponding results and analysis for the CEC'14 test suite. Again, we see that BEPSO compares very well with the comparator algorithms across all dimensions, if not as strongly as for CEC'13. L-SHADE is statistically significantly better, on average, than BEPSO on each dimension, and EPSO is statistically significantly better at 30 dimensions and 100 dimensions. BEPSO is statistically significantly better or equal to all other algorithms on all dimensions. Again, BEPSO performed particularly well on the multimodal and composition problems and strongly on the hybrid functions.

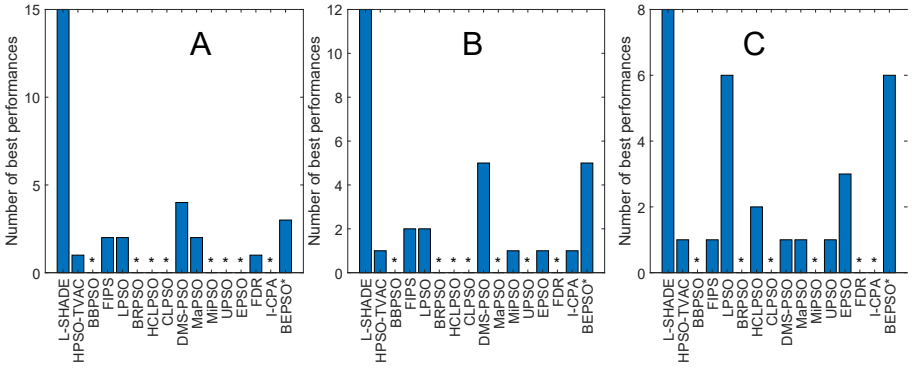

Figure 6. BEPSO comparison. The total number of best performances achieved by each algorithm with respect to mean error values on the **CEC'14** problems. (**A**) 30 dimensions; (**B**) 50 dimensions; (**C**) 100 dimensions. * No best performances achieved.

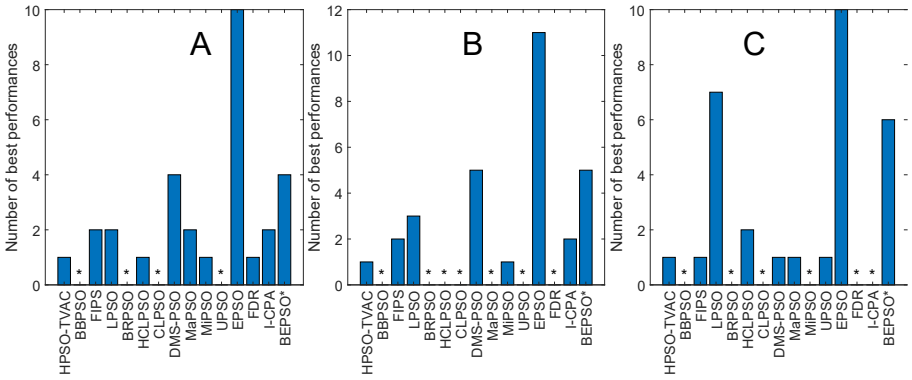

Figure 7. BEPSO comparison. The total number of best performances achieved by each PSO algorithm with respect to mean error values on the **CEC'14** problems. (**A**) 30 dimensions; (**B**) 50 dimensions; (**C**) 100 dimensions. * No best performances achieved.

Table 4. BEPSO statistical tests. Wilcoxon signed-rank test results with a significance level of $p = 0.05$ for **CEC'14** problems at 3 different dimensions.

BEPSO versus (CEC14)																
Dimension	**L-SHADE**	**HPSO-TVAC**	**BBPSO**	**FIPS**	**LPSO**	**BRPSO**	**HCLPSO**	**CLPSO**	**DMS-PSO**	**MaPSO**	**MiPSO**	**UPSO**	**EPSO**	**FDR**	**I-CPA**	**+/=/−**
30	−	=	+	+	=	+	−	+	=	=	=	+	−	=	=	
p	7.7×10^{-4}	0.23	6.8×10^{-5}	0.009	0.64	2.0×10^{-5}	0.009	3.6×10^{-6}	0.24	0.39	0.52	0.003	0.005	0.83	0.07	5/7/3
50	−	=	+	+	=	+	=	+	=	=	=	+	−	=	=	
p	7.5×10^{-4}	0.36	2.1×10^{-5}	6.2×10^{-3}	0.89	1.1×10^{-5}	0.23	3.1×10^{-6}	0.52	0.14	0.25	0.003	0.047	0.49	0.07	5/8/2
100	−	=	+	+	=	+	=	+	=	=	=	+	−	=	=	
p	0.019	0.89	6.6×10^{-6}	3.4×10^{-3}	0.56	1.1×10^{-5}	0.82	2.0×10^{-5}	0.3	0.15	0.1	2.9×10^{-3}	9.6×10^{-3}	0.7	0.15	5/8/2

Figures 8 and 9 and Table 5 give the corresponding results and analysis for the CEC'17 test suite. Here, we, again, see very strong performance from BEPSO across all dimensions, becoming particularly dominant as the dimensions increases, finding more best solutions than any other algorithm at 50-D and 100-D. Table 5 shows that BEPSO is statistically significantly better than the majority of other comparator algorithms at all dimensions, beating 11 out of 15 on the higher dimensions. L-Shade is statistically significantly better at 30 dimensions, but none of the comparator algorithms is statistically significantly better than BEPSO at the higher dimensions (50-D and 100-D). Again, BEPSO performed particularly well on the multimodal and composition problems and strongly on the hybrid problems.

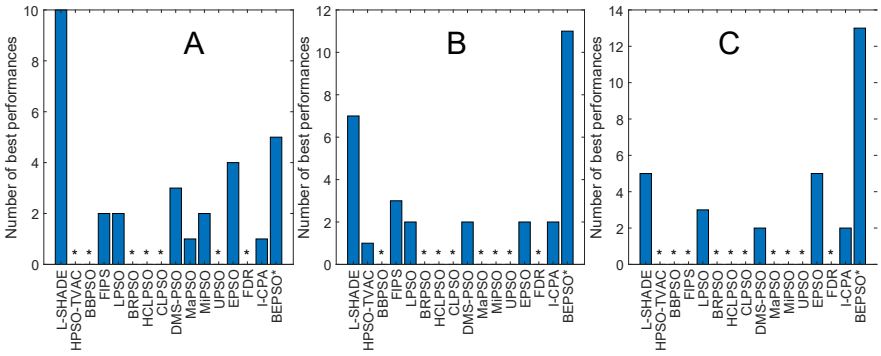

Figure 8. BEPSO comparison. The total number of best performances achieved by each algorithm with respect to mean error values on the **CEC'17** problems. (**A**) 30 dimensions; (**B**) 50 dimensions; (**C**) 100 dimensions. * No best performances achieved.

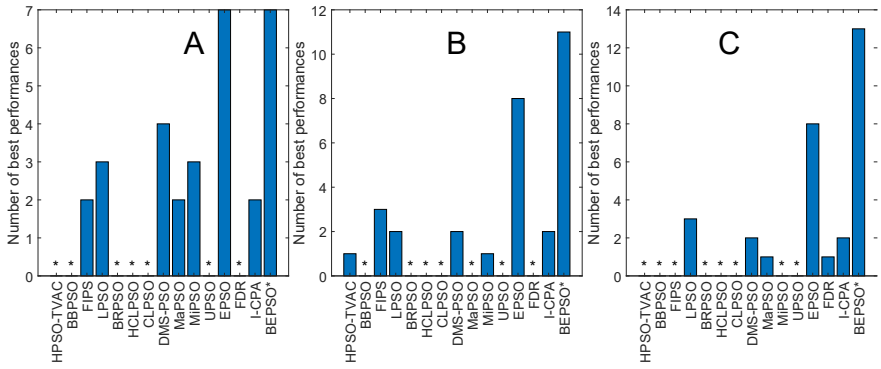

Figure 9. BEPSO comparison. The total number of best performances achieved by each PSO algorithm with respect to mean error values on the **CEC'17** problems. (**A**) 30 dimensions; (**B**) 50 dimensions; (**C**) 100 dimensions. * No best performances achieved.

Table 5. BEPSO statistical tests. Wilcoxon signed-rank test results with a significance level of $p = 0.05$ for **CEC'17** problems at 3 different dimensions.

BEPSO versus (CEC17)																
Dimension	**L-SHADE**	**HPSO-TVAC**	**BBPSO**	**FIPS**	**LPSO**	**BRPSO**	**HCLPSO**	**CLPSO**	**DMS-PSO**	**MaPSO**	**MiPSO**	**UPSO**	**EPSO**	**FDR**	**I-CPA**	**+/=/−**
30	−	+	+	+	=	+	=	+	=	+	=	+	=	+	+	9/5/1
p	0.019	2.1×10^{-3}	2.5×10^{-6}	1.9×10^{-4}	0.46	2.5×10^{-6}	0.58	2.5×10^{-6}	0.74	0.01	0.053	5.7×10^{-6}	0.74	0.01	5.1×10^{-3}	
50	=	+	+	+	=	+	+	+	=	+	+	+	=	+	+	11/4/0
p	0.54	5.8×10^{-3}	2.4×10^{-6}	8.9×10^{-5}	0.21	2.4×10^{-6}	0.03	2.4×10^{-6}	0.62	3.8×10^{-3}	9.0×10^{-3}	2.5×10^{-6}	0.74	4.8×10^{-3}	1.7×10^{-3}	
100	=	+	+	+	=	+	+	+	=	+	+	+	=	+	+	11/4/0
p	0.41	3.8×10^{-3}	2.4×10^{-6}	4.5×10^{-5}	0.12	2.5×10^{-6}	0.03	2.5×10^{-6}	0.29	1.3×10^{-3}	1.2×10^{-3}	2.5×10^{-6}	0.64	4.0×10^{-3}	7.1×10^{-3}	

5.2. BEPSO: Convergence

The analysis of the data presented in the previous subsection showed that the five consistently best-performing algorithms were (in alphabetical order) BEPSO, DMS-PSO, EPSO, HCLPSO and L-SHADE. In Figures 10–12, we compare the average convergence rates towards the best solution of these algorithms across a range of representative 100-D problems for each test suite. It is clear from these figures that BEPSO's convergence rates compare very favourably with the other top-performing algorithms. BEPSO's rates are consistently better than EPSO and HCLPSO and very similar to those of L-SHADE and DMS-PSO.

5.3. AHPSO: Performance

Figures 13–18 and Tables 6–8 show AHPSO's performance on the same set of test suites against the same collection of comparator algorithms. It can be readily seen that for the CEC'13 test suite, just like BEPSO, AHPSO is highly competitive relative to all other comparator PSO algorithms, dominating all of them in terms of number of best solutions found at all dimensions, with performance also increasing as the dimensions of the problem increase (Figure 14). Its performance is also highly competitive relative to the powerful L-SHADE differential evolution algorithm at all dimensions, achieving more best solutions at 100-D (Figure 13). Table 6 shows us that AHPSO's performance on the CEC'13 test suite is statistically significantly better than 10 of the 15 comparator algorithms across all dimensions, with no significant difference for the other 5. AHPSO performed particularly well on the composition test functions.

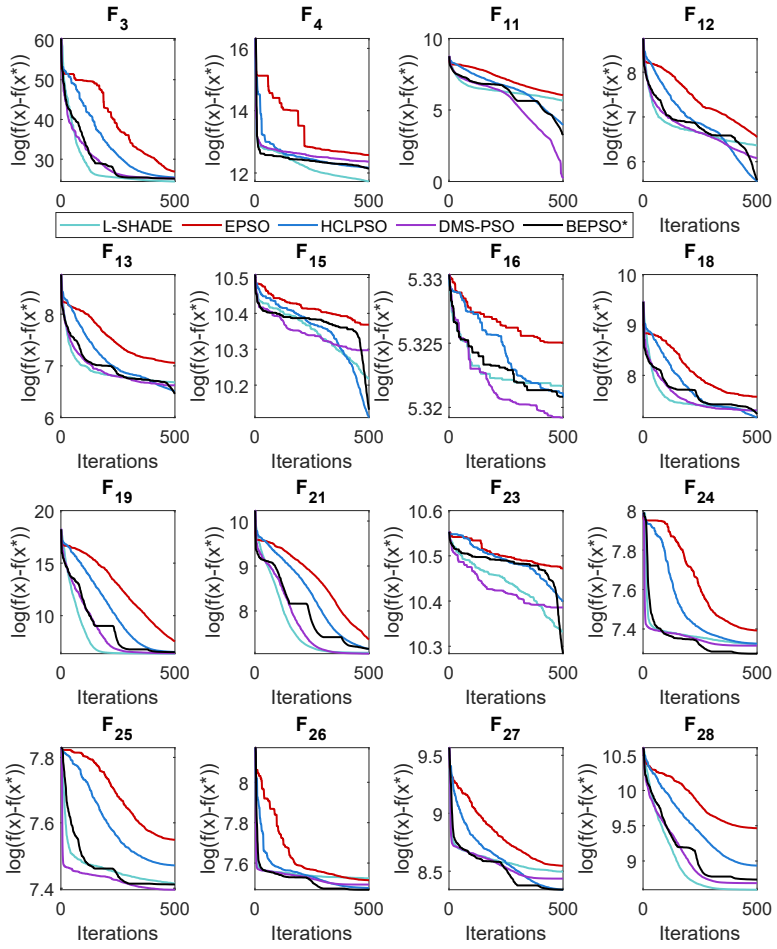

Figure 10. Average convergence rate comparison of BEPSO with L-SHADE, EPSO, HCLPSO and DMS-PSO on various 100-dimensional CEC'13 problems.

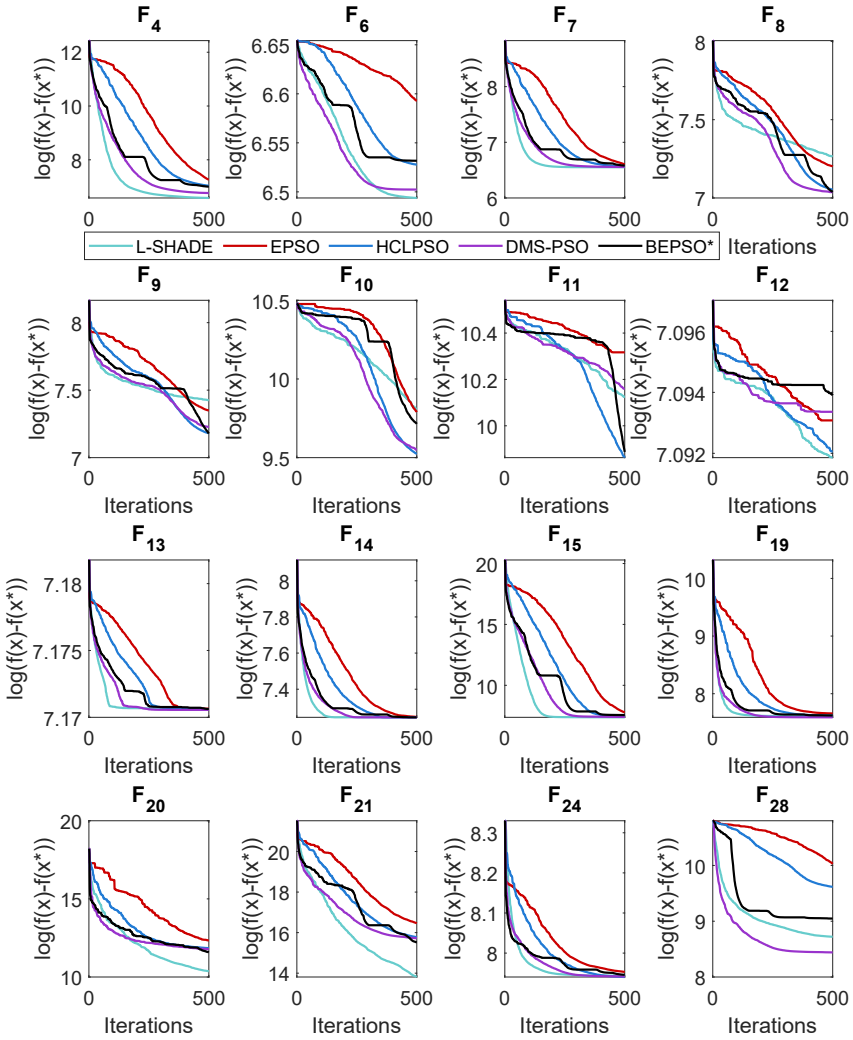

Figure 11. Average convergence rate comparison of BEPSO with L-SHADE, EPSO, HCLPSO and DMS-PSO on various 100-dimensional CEC'14 problems.

Figures 15 and 16 and Table 7 give the results for AHPSO on the CEC'14 test suite. Relative performance is strong but not as good as for CEC'13. But we see that the increase in performance with dimensions is more marked, with AHPSO achieving more best solutions than L-SHADE at 100-D. The statistical analysis reveals that L-SHADE and EPSO were statistically significantly better at 30-D and 50-D, but AHPSO was statistically significantly better or equal to all 15 comparator algorithms at 100-D. AHPSO was superior to all other algorithms on the composition test functions.

Figures 17 and 18 and Table 8 give the results for the CEC'17 test suite. Again, AHPSO's relative performance is very strong, being statistically significantly better than 11 of the 15 comparator algorithms at the higher dimensions (50-D and 100-D), while no other algorithms were statistically better than AHPSO at these dimensions. At 30-D, only L-SHADE is statistically significantly better. AHPSO performed particularly well on multimodal and composition test functions. At 100-D, AHPSO exhibited the best performance on more functions (11 out of 29) than all other comparison algorithms (including L-SHADE).

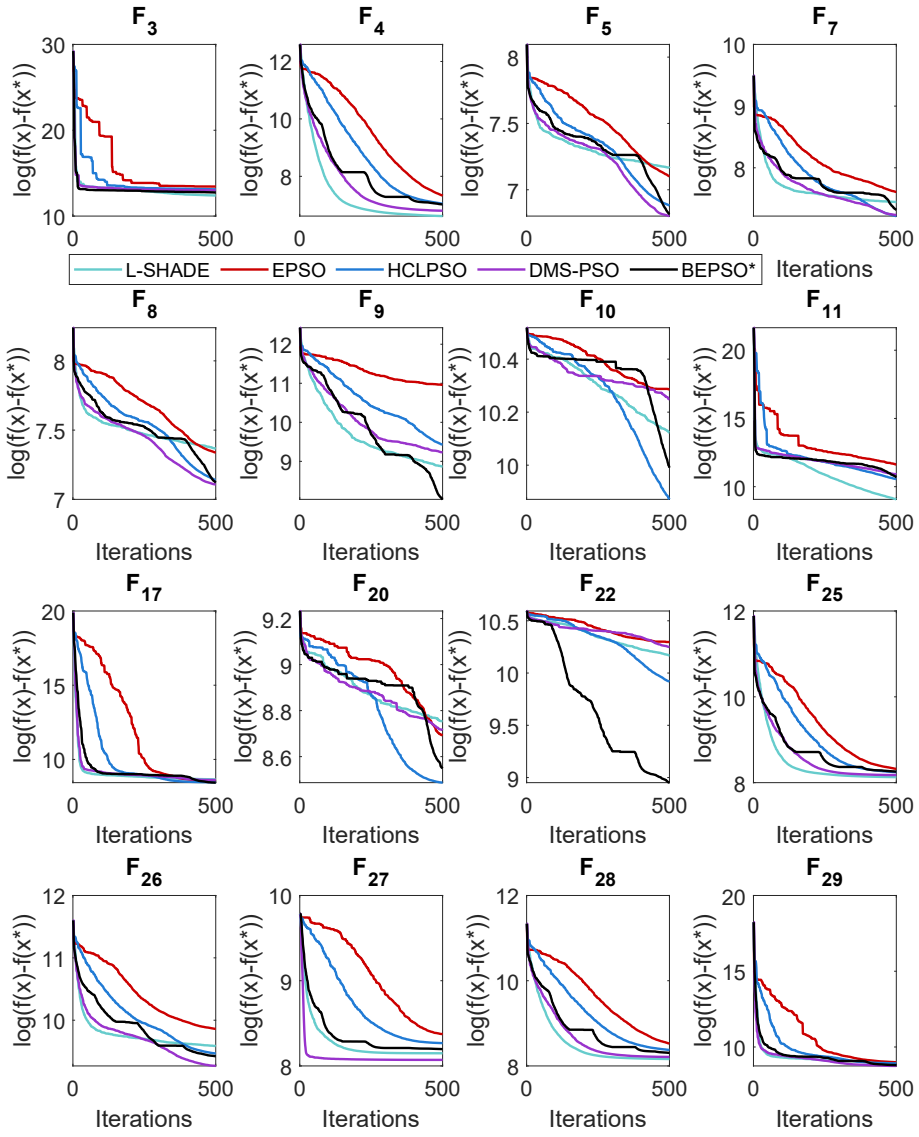

Figure 12. Average convergence rate comparison of BEPSO with L-SHADE, EPSO, HCLPSO and DMS-PSO on various 100-dimensional CEC'17 problems.

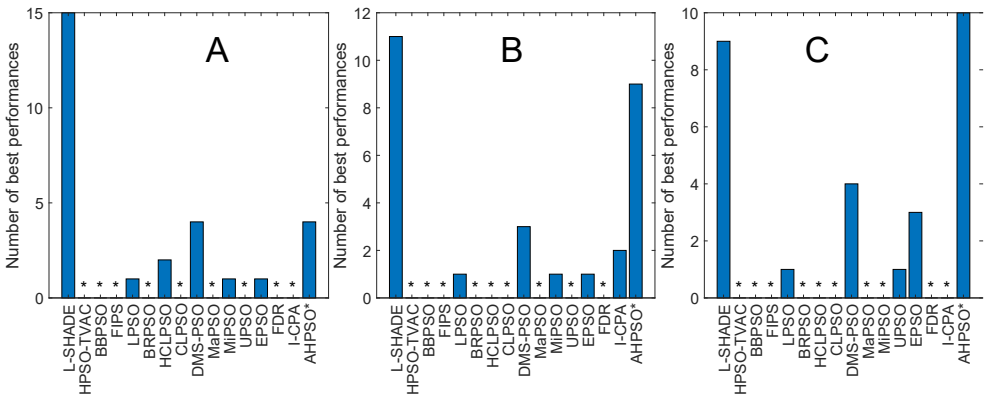

Figure 13. AHPSO comparison. The total number of best performances achieved by each algorithm with respect to mean error values (relative to best known/found function values) on the **CEC'13** problems. (**A**) 30 dimensions; (**B**) 50 dimensions; (**C**) = 100 dimensions. * No best performances achieved.

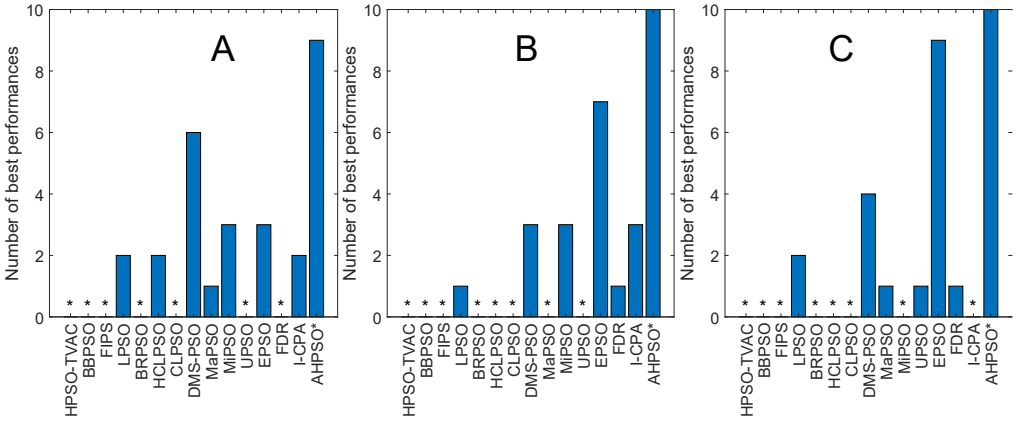

Figure 14. AHPSO comparison. The total number of best performances achieved by each PSO algorithm with respect to mean error values on the **CEC'13** problems. (**A**) 30 dimensions; (**B**) 50 dimensions; (**C**) 100 dimensions. * No best performances achieved.

Table 6. AHPSO statistical tests. Wilcoxon signed-rank test results with a significance level of $p = 0.05$ for **CEC'13** problems at 3 different dimensions. +: statistically significantly better; =: no significant difference; − : statistically significantly worse.

AHPSO versus (CEC13)																
Dimension	**L-SHADE**	**HPSO-TVAC**	**BBPSO**	**FIPS**	**LPSO**	**BRPSO**	**HCLPSO**	**CLPSO**	**DMS-PSO**	**MaPSO**	**MiPSO**	**UPSO**	**EPSO**	**FDR**	**I-CPA**	**+/=/−**
30	=	+	+	+	+	+	=	+	=	+	+	+	=	=	+	
p	0.09	1.8×10^{-3}	6.0×10^{-6}	3.0×10^{-5}	0.048	1.1×10^{-5}	0.95	5.3×10^{-6}	0.7	2.7×10^{-3}	3.7×10^{-3}	1.3×10^{-5}	0.49	0.21	0.018	10/5/0
50	=	+	+	+	+	+	=	+	=	+	+	+	=	=	+	
p	0.10	7.4×10^{-3}	6.3×10^{-6}	2.5×10^{-5}	0.043	1.5×10^{-4}	0.23	5.2×10^{-6}	0.93	0.004	0.0058	2.3×10^{-5}	0.27	0.34	0.03	10/5/0
100	=	+	+	+	+	+	=	+	=	+	+	+	=	=	+	
p	0.25	3.1×10^{-2}	6.6×10^{-6}	1.3×10^{-5}	0.011	9.0×10^{-6}	0.54	5.6×10^{-6}	0.95	0.003	0.003	8.4×10^{-5}	0.33	0.14	4.3×10^{-3}	10/5/0

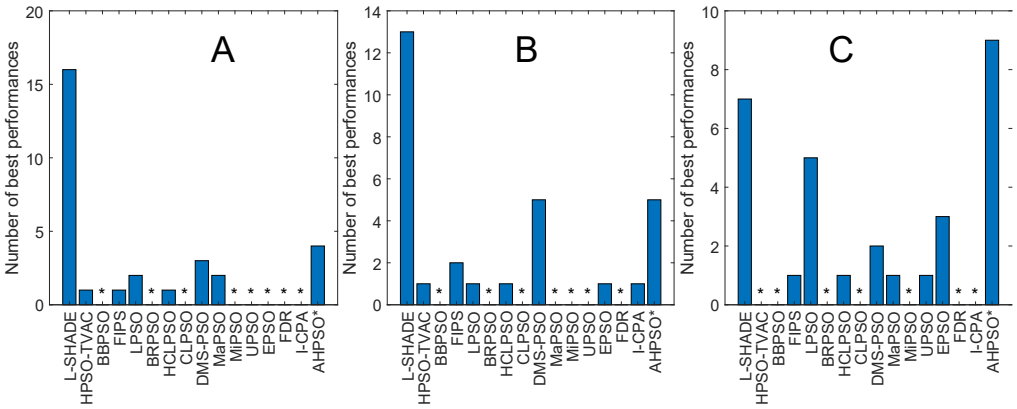

Figure 15. AHPSO comparison. The total number of best performances achieved by each algorithm with respect to mean error values on the **CEC'14** problems. (**A**) 30 dimensions; (**B**) 50 dimensions; (**C**) 100 dimensions. * No best performances achieved.

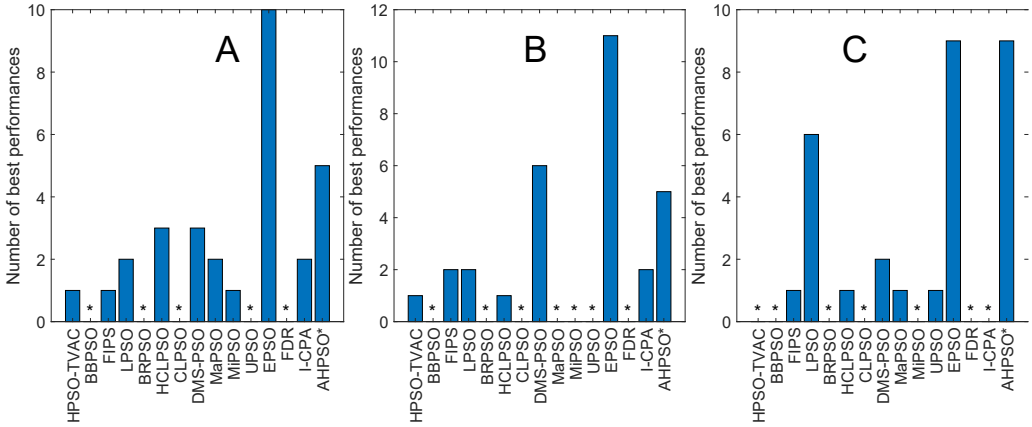

Figure 16. AHPSO comparison. The total number of best performances achieved by each PSO algorithm with respect to mean error values on the **CEC'14** problems. (**A**) 30 dimensions; (**B**) 50 dimensions; (**C**) 100 dimensions. * No best performances achieved.

Table 7. AHPSO statistical tests. Wilcoxon Signed Rank Test Results with a Significance Level of $p = 0.05$ for **CEC'14** problems at 3 different dimensions.

AHPSO versus (CEC14)																
Dimension	**L-SHADE**	**HPSO-TVAC**	**BBPSO**	**FIPS**	**LPSO**	**BRPSO**	**HCLPSO**	**CLPSO**	**DMS-PSO**	**MaPSO**	**MiPSO**	**UPSO**	**EPSO**	**FDR**	**I-CPA**	**+/=/−**
30	−	=	+	+	=	+	−	+	=	=	=	+	−	=	=	
p	1.9×10^{-3}	0.26	6.0×10^{-5}	0.014	0.93	1.8×10^{-5}	0.034	3.2×10^{-6}	0.16	0.50	0.55	0.002	0.001	0.57	0.08	5/7/3
50	−	=	+	+	=	+	=	+	=	=	=	+	−	=	=	
p	1.3×10^{-3}	0.43	2.7×10^{-5}	6.9×10^{-3}	0.89	8.7×10^{-6}	0.15	2.5×10^{-6}	0.26	0.17	0.26	0.001	0.026	0.61	0.07	5/8/2
100	=	=	+	+	=	+	=	+	=	+	+	+	=	=	+	
p	0.23	0.28	6.1×10^{-6}	7.6E-4	0.82	7.2×10^{-6}	0.45	3.4×10^{-6}	0.68	0.03	0.016	7.6×10^{-4}	0.11	0.14	0.02	8/7/0

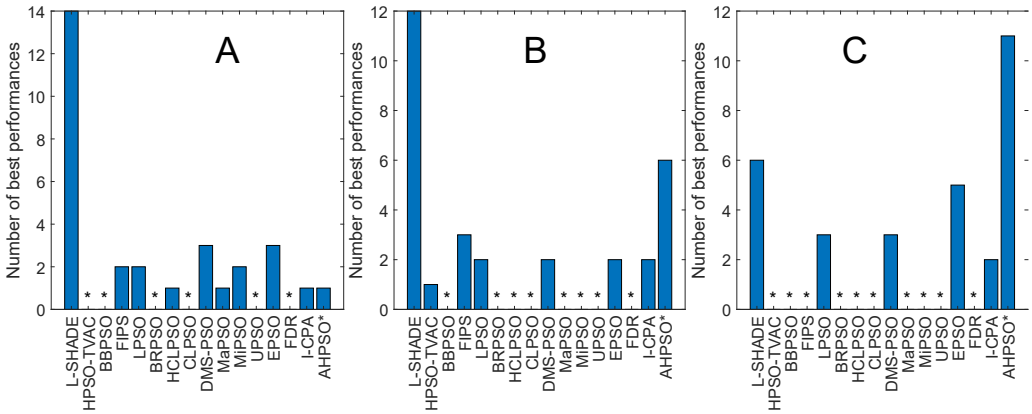

Figure 17. AHPSO comparison. The total number of best performances achieved by each algorithm with respect to mean error values on the **CEC'17** problems. (**A**) 30 dimensions; (**B**) 50 dimensions; (**C**) 100 dimensions. * No best performances achieved.

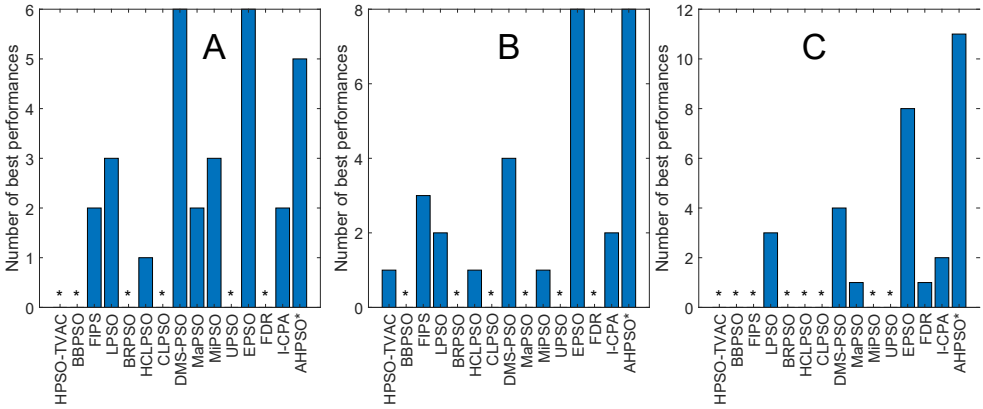

Figure 18. AHPSO comparison. The total number of best performances achieved by each PSO algorithm with respect to mean error values on the **CEC'17** problems. (**A**) 30 dimensions; (**B**) 50 dimensions; (**C**) 100 dimensions. * No best performances achieved.

Table 8. AHPSO statistical tests. Wilcoxon signed-rank test results with a significance level of $p = 0.05$ for **CEC'17** problems at 3 different dimensions.

AHPSO versus (CEC17)																
Dimension	**L-SHADE**	**HPSO-TVAC**	**BBPSO**	**FIPS**	**LPSO**	**BRPSO**	**HCLPSO**	**CLPSO**	**DMS-PSO**	**MaPSO**	**MiPSO**	**UPSO**	**EPSO**	**FDR**	**I-CPA**	**+/=/−**
30	−	+	+	+	=	+	=	+	=	+	=	+	=	+	+	
p	7.7×10^{-4}	1.8×10^{-3}	2.4×10^{-6}	3.7×10^{-4}	0.7	2.9×10^{-6}	0.96	2.5×10^{-6}	0.69	0.043	0.13	8.5×10^{-6}	0.39	0.028	4.0×10^{-3}	9/5/1
50	=	+	+	+	=	+	+	+	=	+	+	+	=	+	+	
p	0.17	2.1×10^{-3}	2.4×10^{-6}	9.4×10^{-5}	0.09	2.5×10^{-6}	0.013	2.4×10^{-6}	0.83	2.2×10^{-3}	4.2×10^{-3}	2.7×-6	0.74	9.8×10^{-4}	3.2×10^{-4}	11/4/0
100	=	+	+	+	=	+	+	+	=	+	+	+	=	+	+	
p	0.48	2.0×10^{-3}	2.5×10^{-6}	5.6×10^{-5}	0.14	2.5×10^{-6}	0.04	2.5×10^{-6}	0.52	1.3×10^{-3}	1.8×10^{-3}	2.7×10^{-6}	0.65	2.4×10^{-3}	7.1×10^{-3}	11/4/0

5.4. AHPSO: Convergence

Like BEPSO, AHPSO was one of the consistently best-performing algorithms across all the test suites. Figures 19–21 show its convergence rates relative to the other top-performing algorithms. AHPSO's converge rates are very similar to the best rates achieved; they are consistently better than those achieved by EPSO and HCLPSO.

The comparative experiments for the two algorithms across the CEC test suites were run independently. Perhaps unsurprisingly, given the nature of the results presented above, a third set of independent runs across the CEC'13, CEC'14 and CEC'17 test suites revealed that there was no statistically significant difference in performance of BEPSO and AHPSO at 30 dimensions, 50 dimensions and 100 dimensions [100].

5.5. Constrained Optimisation Problems

Besides verifying the performance of the two novel algorithms on the CEC'13, CEC'14 and CEC'17 benchmark test suites, we examined the performance of BEPSO and AHPSO on 14 constrained real-world problems [101] comprising process synthesis and design, mechanical engineering and power system problems compared against L-SHADE and the 10 best PSO variants employed in the previous experiments. The complete list of constrained real-world problems is displayed in Table 1 in Section 4. Each problem was tested 30 times.

Figure 22 shows the number of best solutions found by the algorithms across all constrained problems. By this measure, AHPSO and L-Shade perform best. AHPSO performed particularly well on the difficult power electronic problems (solving optimal pulse-width modulation problems with relatively large numbers of variables and constraints), finding

the best known solution for three of the four problems in this class, with L-SHADE being the only other algorithm able to find a (single) best solution in this class. Tables 9 and 10 show the pairwise statistical significance analysis (with appropriate reductions for multiple comparisons). It can be seen that none of the comparator algorithms is statistically significantly better than either BEPSO or AHPSO, and AHPSO is better than L-SHADE. In a detailed ranking analysis across all runs on all problems of all algorithms, the best mean rank (2.36) was achieved by BEPSO, the second best (2.5) by L-SHADE and the third best (2.57) by AHPSO.

Hence, on constrained real-world problems the two new BEPSO and AHPSO algorithms performed extremely strongly in comparison with all other comparator algorithms.

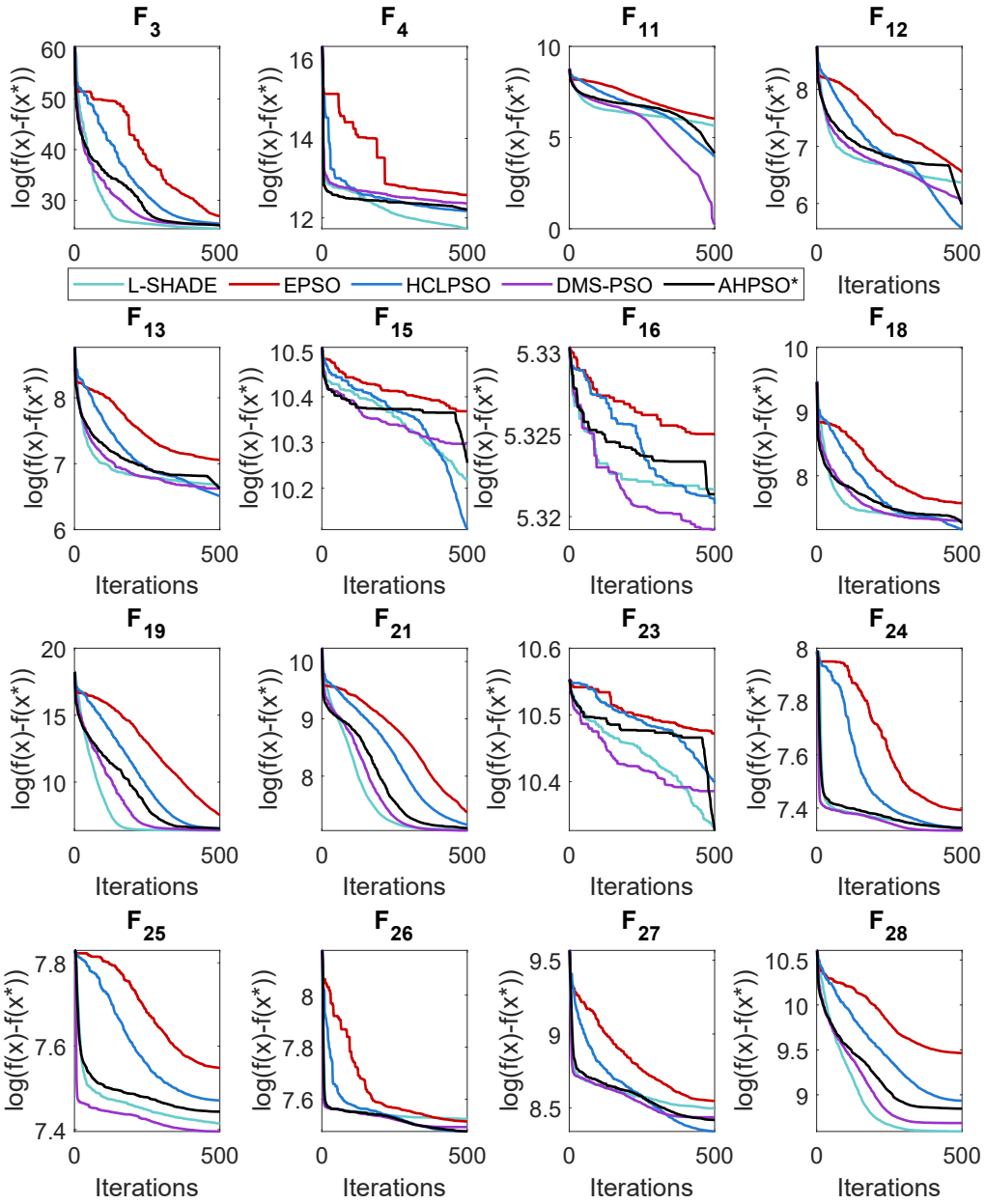

Figure 19. Average convergence rate comparison of AHPSO with L-SHADE, EPSO, HCLPSO and DMS-PSO on various 100-dimensional CEC'13 problems.

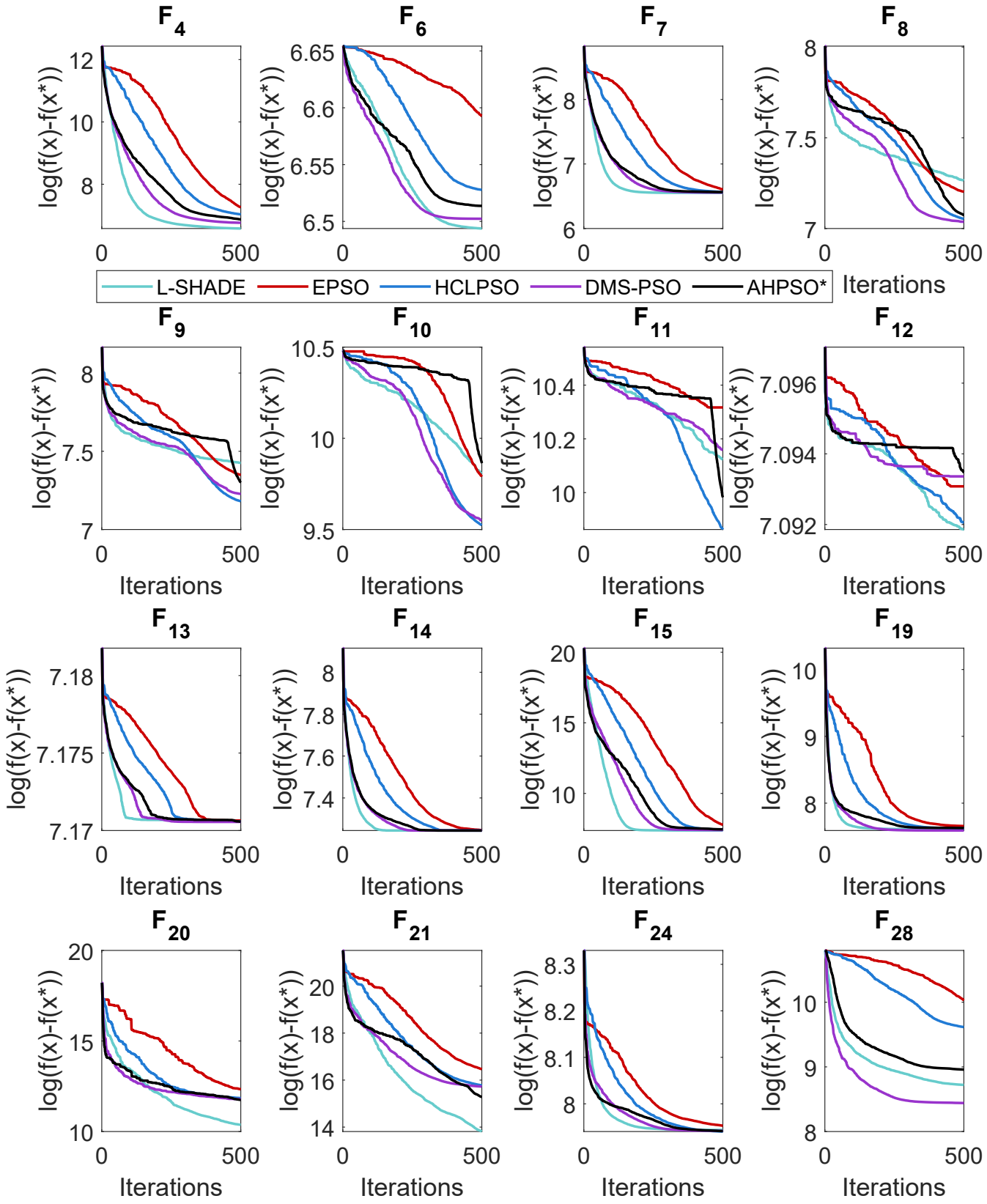

Figure 20. Average convergence rate comparison of AHPSO with L-SHADE, EPSO, HCLPSO and DMS-PSO on various 100-dimensional CEC'14 problems.

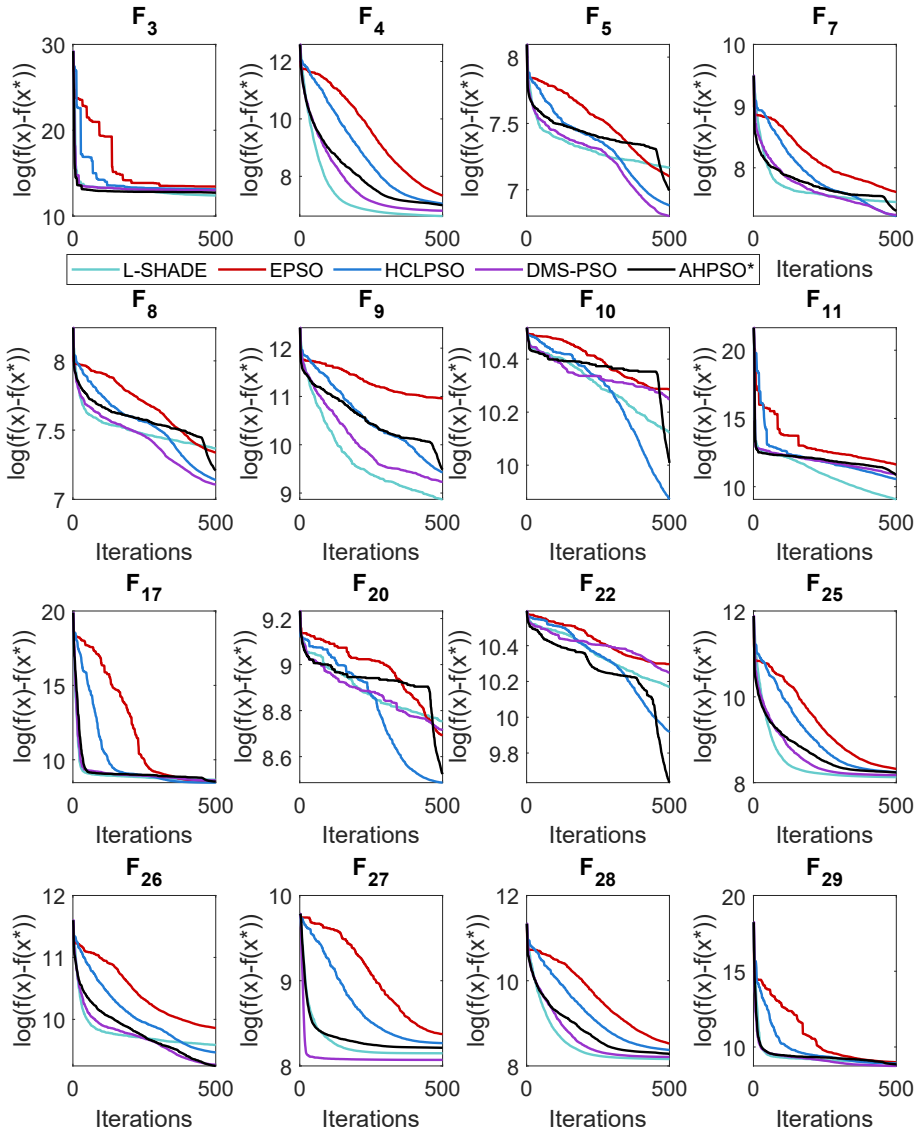

Figure 21. Average convergence rate comparison of AHPSO with L-SHADE, EPSO, HCLPSO and DMS-PSO on various 100-dimensional CEC'17 problems.

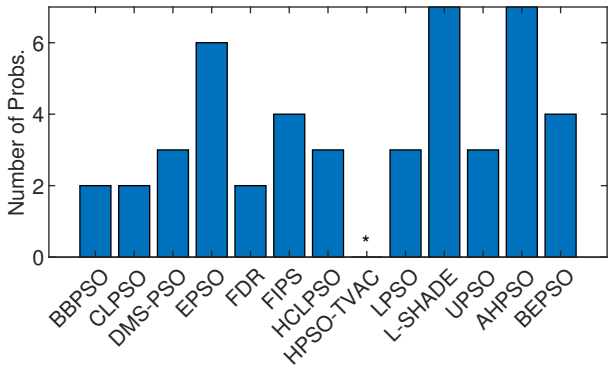

Figure 22. Total number of constrained real-world problems solved (lowest obtained mean error value) by the proposed and comparator algorithms. * No best solutions found.

Table 9. BEPSO statistical tests. Wilcoxon signed-rank test results with a significance level of $p = 0.05$ for **constrained real-world** problems. The last row shows p values.

BEPSO versus (Constrained Problems)											
L-SHADE	**HPSO-TVAC**	**BBPSO**	**FIPS**	**LPSO**	**HCLPSO**	**CLPSO**	**DMS-PSO**	**UPSO**	**EPSO**	**FDR**	**+/=/−**
=	+	+	+	=	=	+	+	=	=	+	6/5/0
0.58	1.2×10^{-4}	2.7×10^{-3}	0.03	0.24	0.10	1.2×10^{-3}	4.6×10^{-3}	1.0	0.38	0.014	

Table 10. AHPSO statistical tests. Wilcoxon signed-rank test results with a significance level of $p = 0.05$ for **constrained real-world** problems. The last row shows p values.

AHPSO versus (Constrained Problems)											
L-SHADE	**HPSO-TVAC**	**BBPSO**	**FIPS**	**LPSO**	**HCLPSO**	**CLPSO**	**DMS-PSO**	**UPSO**	**EPSO**	**FDR**	**+/=/−**
+	+	=	=	=	=	+	+	=	=	=	4/7/0
0.02	1.2×10^{-4}	6.7×10^{-2}	0.56	0.94	0.83	2.6×10^{-2}	2.9×10^{-2}	0.25	0.16	0.46	

6. Discussion

Both BEPSO and AHPSO algorithms consistently performed particularly well on the higher-dimensional, more complex problems, including unconstrained multimodal and composition problems and problems involving many constraints, matching or bettering the performance of all other comparator algorithms. However, while their performance on simpler, lower-dimensional problems was perfectly adequate, it did not match that of the best of the comparator algorithms. This limitation seems to be associated with the deliberately high agent-level heterogeneity that is inherent to the designs of BEPSO and AHPSO. This property significantly aids the algorithms in high-dimensional and complex search spaces while being less effective in low-dimensional spaces.

The various stochastic mechanisms embedded in both algorithms were partly designed to maintain diversity while enabling efficient search. Figure 23 shows that both BEPSO and AHPSO achieve maintenance of diversity. The graphs use the following diversity quantification method proposed in [1]:

$$Diversity(S(t)) = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\sum_{d=1}^{D}(\boldsymbol{x}_i^d - \overline{\boldsymbol{x}^d})^2} \tag{28}$$

where n is the population size, D is the problem dimension and $\overline{\boldsymbol{x}^d}$ is the average value of $\boldsymbol{x}^d$ (the dth component of the solution vector).

The figure reveals that while EPSO consistently exhibited a more diverse population compared to the other high-performing algorithms on the 100-dimensional CEC'17 problems, BEPSO and AHPSO maintained better population diversity compared to all the rest, namely L-SHADE, LPSO, DMS-PSO and HCLPSO. The trend was repeated across the other test suites for the majority of functions.

While the multiple, mainly stochastic components of the new BEPSO and AHPSO algorithms tend to complement each other and, in combination, maintain diversity; balance exploration and exploitation; and enable efficient, high-quality search, they do increase the overall complexity of the algorithm. But it is worth noting that many recent variants of metaheuristics, such as that proposed in [102,103]. and several of the comparator algorithms used in the current study have a much more complex structure than their canonical versions. These variants are discernibly more efficient and are able to handle more diverse

problems. Hence, the algorithms proposed in this paper are not unusual in comprising a more complex structure and set of interaction mechanisms in comparison to canonical algorithms. However, their individual elements are all simple. Indeed, the slightly more complex framework of the proposed algorithms is inspired by the observation that biological degeneracy plays a vital role in boosting evolvability in nature and, therefore, can improve the efficiency of search processes [104]. Biological degeneracy, whereby multiple interacting mechanisms enable multiple different ways of producing an outcome, is an ubiquitous property of biological systems at all levels of organisation [105,106] reveals that systems with simple redundancy have considerably lower evolvability than degenerate (e.g., highly versatile) systems with selectable changes of behaviour, which enable compensatory actions to occur within the system (exactly what happens at the core of the new algorithms presented in this paper).

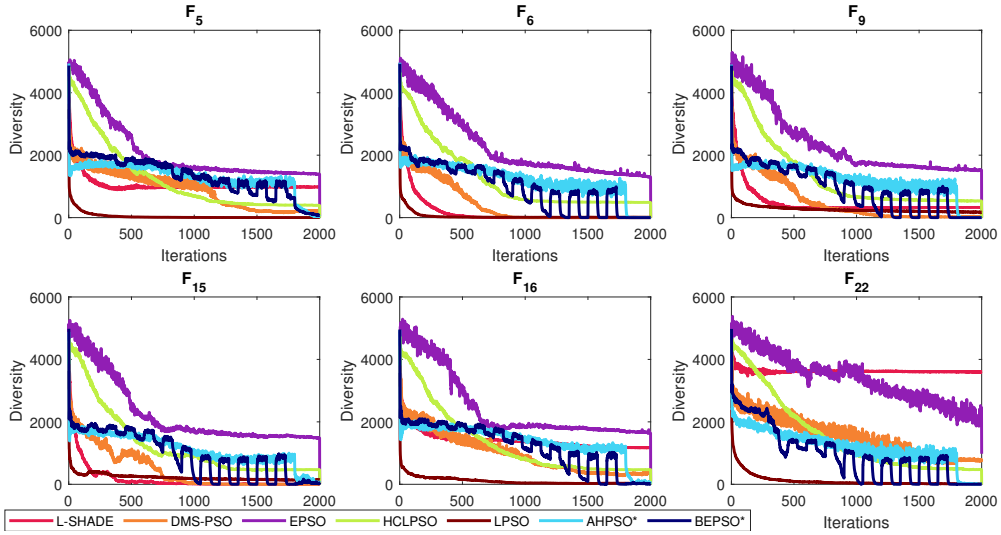

Figure 23. Diversity comparison for best-performing algorithms for various representative 100-dimensional CEC'17 problems.

However, this increased algorithmic complexity does not necessarily equate to significantly increased computational cost relative to other high-performing algorithms. Figure 24 shows the time complexity (T_c) of BEPSO, AHPSO and the comparator algorithms at various problem dimensions calculated using the metric described in Section 4.

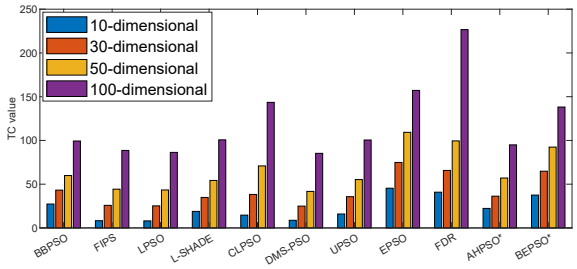

Figure 24. Time complexity of BEPSO, AHPSO and comparator algorithms for four dimension sizes.

As expected, the basic canonical PSO obtained the lowest T_c values across all dimensions (although it is the worst-performing algorithm in terms of solution quality). However, the complexity of the high-performing AHPSO, L-SHADE and HCLPSO algorithms across all four dimensions is very similar and not much higher than that of basic PSO. BEPSO's complexity is a little higher but still very competitive. In summary, BEPSO and AHPSO exhibit similar and, in some cases, better complexities compared to their peers; however, it is worth highlighting that they exhibit a significantly lower increase in complexity with problem dimensionality (especially AHPSO).

7. Conclusions

The two novel bio-inspired search algorithms presented in this paper, namely BEPSO and AHPSO, performed very well across a wide range of unconstrained and constrained optimisation problems, using a single set of parameters for all. On the CEC13 test suite, across all dimensions, both BEPSO and AHPSO performed statistically significantly better than 10 of the 15 comparator algorithms, while none of the remaining 5 algorithms performed significantly better than either BEPSO or AHPSO. On the CEC17 test suite, on the 50-D and 100-D problems, both BEPSO and AHPSO performed statistically significantly better than 11 of the 15 comparator algorithms, while none of the remaining 4 algorithms performed significantly better than either BEPSO or AHPSO. On the constrained problem set, in terms of mean rank across 30 runs on all problems, BEPSO was first, and AHPSO was third. This provides further evidence that bio-inspiration—in this case, various kinds of animal group behaviours—continues to be a very fruitful source for algorithm design.

Although both algorithms employ heterogeneous population models, are inspired by animal group behaviours and are designed to maintain diversity and avoid premature convergence and stagnation, the underlying metaphors and inspirations are very different, as are the behavioural rules used to guide particle movement. This illustrates one of the great strengths of the basic PSO framework, namely that there are numerous ways to modify and improve it, often, as in the new algorithms described here, by dynamically changing population structures and movement strategies.

Both algorithms consistently performed particularly well on the higher-dimensional, more complex problems, including unconstrained multimodal and composition problems and problems involving many constraints, matching or bettering the performance of all other comparator algorithms.

The novel mechanisms introduced in these heterogeneous PSO variants were able to maintain population diversity while enabling rapid convergence to optimal or near-optimal solutions. They provided a highly efficient balance between exploration and exploitation.

Considering the efficacy of the new algorithms on high-dimensional problems, further investigation on very large-scale problems would be another interesting direction for future work. AHPSO's superior performance on constrained power electronic problems and the fact that other researchers have recently successfully used it to find the optimal design parameters for hybrid active power filters [107] suggest that an expanded investigation of its applications to other problems in the power electronics domain might be very fruitful.

Although canonical PSO is a very efficient algorithm for many small-scale optimisation problems, as with many metaheuristics, it suffers from the curse of dimensionality. Although the performance of the new PSO algorithms presented in this paper were much less impacted by problem dimensionality compared with other algorithms used in the experiments, various other limitations were identified for possible future improvements. It is clear that the heterogeneous nature of BEPSO and AHPSO provides various strong advantages for the general process of optimisation. However, it is this very property that makes them so good at complex (multimodal, composition, etc.), higher-dimensional problems, which means that their performance on (simpler) unimodal problems, while perfectly adequate, is consistently worse than that of some of their competitors, as their speed of finding solutions is slower. This could be accepted as a clear example of the No Free Lunch Theorem for optimization [108], which tells us that no method can be uniformly excellent on all possible problems, or it could be a prompt to try and improve the algorithms with suitable mechanisms to address this particular issue without destroying their power on other problem types. In general, unimodal problems do not require such a careful balance between exploration and exploitation; typically, intensive exploitation is expected to dominate the search process. This is a possible area for future research, either attempting to expand the general efficacy of the methods or, alternatively, to produce variants of the algorithms that are specialized to specific problem classes.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/biomimetics9090538/s1, details of initial parameter tuning investigations.

Author Contributions: Conceptualization, F.T.V.; methodology, F.T.V. and P.H.; software, F.T.V.; experiments and data collection, F.T.V.; formal analysis, F.T.V.; data curation, F.T.V.; writing—original draft preparation, F.T.V. and P.H.; writing—review and editing, F.T.V. and P.H.; visualization, F.T.V.; supervision, P.H.; project administration, P.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Code for AHPSO and BEPSO algorithms can be found at https://github.com/FTVarna [Accessed 3 September 2024]; full experimental data and can be found at https://doi.org/10.25377/sussex.26146594 [Accessed 3 September 2024]; full details of the extensive parameter investigations can be found in the Supplementary Material.

Acknowledgments: Thanks to members of the AI group at Sussex, including Chris Buckley and Andy Philippides, for useful discussions on this work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Engelbrecht, A. *Fundamentals of Computational Swarm Intelligence*; Wiley: Hoboken, NJ, USA, 2005.
2. Yang, X.S.; Cui, Z.; Xiao, R.; Gandomi, A.H.; Karamanoglu, M. (Eds.) *Swarm Intelligence and Bio-Inspired Computation*; Elsevier: Oxford, UK, 2013.
3. Kiran, M.S.; Beskirli, M. A New Approach Based on Collective Intelligence to Solve Traveling Salesman Problems. *Biomimetics* **2024**, *9*, 118. [CrossRef]
4. Pham, Q.V.; Nguyen, D.C.; Mirjalili, S.; Hoang, D.T.; Nguyen, D.N.; Pathirana, P.N.; Hwang, W.J. Swarm intelligence for next-generation networks: Recent advances and applications. *J. Netw. Comput. Appl.* **2021**, *191*, 103141. [CrossRef]
5. Camci, E.; Kripalani, D.R.; Ma, L.; Kayacan, E.; Khanesar, M.A. An aerial robot for rice farm quality inspection with type-2 fuzzy neural networks tuned by particle swarm optimization-sliding mode control hybrid algorithm. *Swarm Evol. Comput.* **2018**, *41*, 1–8. [CrossRef]
6. Ehteram, M.; Binti Othman, F.; Mundher Yaseen, Z.; Abdulmohsin Afan, H.; Falah Allawi, M.; Bt. Abdul Malek, M.; Najah Ahmed, A.; Shahid, S.; P. Singh, V.; El-Shafie, A. Improving the Muskingum Flood Routing Method Using a Hybrid of Particle Swarm Optimization and Bat Algorithm. *Water* **2018**, *10*, 807. [CrossRef]
7. Cao, Y.; Ye, Y.; Zhao, H.; Jiang, Y.; Wang, H.; Shang, Y.; Wang, J. Remote sensing of water quality based on HJ-1A HSI imagery with modified discrete binary particle swarm optimization-partial least squares (MDBPSO-PLS) in inland waters: A case in Weishan Lake. *Ecol. Inform.* **2018**, *44*, 21–32. [CrossRef]
8. Thanga Ramya, S.; Arunagiri, B.; Rangarajan, P. Novel effective X-path particle swarm optimization based deprived video data retrieval for smart city. *Clust. Comput.* **2019**, *22*, 13085–13094. [CrossRef]
9. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; IEEE: Piscataway, NJ, USA, 1995; Volume 4, pp. 1942–1948. [CrossRef]
10. Engelbrecht, A. Particle swarm optimization: Velocity initialization. In Proceedings of the 2012 IEEE Congress on Evolutionary Computation, Brisbane, QLD, Australia, 10–15 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 1–8. [CrossRef]
11. Engelbrecht, A.P.; Bosman, P.; Malan, K.M. The influence of fitness landscape characteristics on particle swarm optimisers. *Nat. Comput.* **2022**, *21*, 335–345. [CrossRef]
12. Jiang, M.; Luo, Y.; Yang, S. Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Inf. Process. Lett.* **2007**, *102*, 8–16. [CrossRef]
13. Trelea, I.C. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Inf. Process. Lett.* **2003**, *85*, 317–325. [CrossRef]
14. Varna, F.T.; Husbands, P. HIDMS-PSO: A New Heterogeneous Improved Dynamic Multi-Swarm PSO Algorithm. In Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, 1–4 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 473–480. [CrossRef]
15. Varna, F.T.; Husbands, P. HIDMS-PSO with Bio-inspired Fission-Fusion Behaviour and a Quorum Decision Mechanism. In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1398–1405. [CrossRef]
16. Varna, F.T.; Husbands, P. Genetic Algorithm Assisted HIDMS-PSO: A New Hybrid Algorithm for Global Optimisation. In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1304–1311. [CrossRef]

17. Wang, S.; Liu, G.; Gao, M.; Cao, S.; Guo, A.; Wang, J. Heterogeneous comprehensive learning and dynamic multi-swarm particle swarm optimizer with two mutation operators. *Inf. Sci.* **2020**, *540*, 175–201. [CrossRef]
18. Yu, G.R.; Chang, Y.D.; Lee, W.S. Maximum Power Point Tracking of Photovoltaic Generation System Using Improved Quantum-Behavior Particle Swarm Optimization. *Biomimetics* **2024**, *9*, 223. [CrossRef] [PubMed]
19. Yue, Y.; Cao, L.; Chen, H.; Chen, Y.; Su, Z. Towards an Optimal KELM Using the PSO-BOA Optimization Strategy with Applications in Data Classification. *Biomimetics* **2023**, *8*, 306. [CrossRef]
20. Shankar, T.; Shanmugavel, S.; Rajesh, A. Hybrid HSA and PSO algorithm for energy efficient cluster head selection in wireless sensor networks. *Swarm Evol. Comput.* **2016**, *30*, 1–10. [CrossRef]
21. Sahoo, B.M.; Pandey, H.M.; Amgoth, T. GAPSO-H: A hybrid approach towards optimizing the cluster based routing in wireless sensor network. *Swarm Evol. Comput.* **2021**, *60*, 100772. [CrossRef]
22. Liang, J.; Qin, A.; Suganthan, P.; Baskar, S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.* **2006**, *10*, 281–295. [CrossRef]
23. Lynn, N.; Suganthan, P.N. Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. *Swarm Evol. Comput.* **2015**, *24*, 11–24. [CrossRef]
24. Li, L.; Yang, S.; Nguyen, T.T. A Self-Learning Particle Swarm Optimizer for Global Optimization Problems. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2012**, *42*, 627–646. [CrossRef]
25. Kennedy, J. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; IEEE: Piscataway, NJ, USA, 1999; pp. 1931–1938. [CrossRef]
26. Kennedy, J.; Mendes, R. Population structure and particle swarm performance. In Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002, Honolulu, HI, USA, 12–17 May 2002; IEEE: Piscataway, NJ, USA, 2002; Volume 2, pp. 1671–1676. [CrossRef]
27. Suganthan, P. Particle swarm optimiser with neighbourhood operator. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; IEEE: Piscataway, NJ, USA, 1999; pp. 1958–1962. [CrossRef]
28. Varna, F.T.; Husbands, P. HIDMS-PSO Algorithm with an Adaptive Topological Structure. In Proceedings of the 2021 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 5–7 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–8. [CrossRef]
29. Liang, J.J.; Suganthan, P.N. Dynamic multi-swarm particle swarm optimizer. In Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS 2005), Pasadena, CA, USA, 8–10 June 2005; IEEE: Piscataway, NJ, USA, 2005; Volume 2005, pp. 127–132. [CrossRef]
30. Trillo, P.A.; Benson, C.S.; Caldwell, M.S.; Lam, T.L.; Pickering, O.H.; Logue, D.M. The Influence of Signaling Conspecific and Heterospecific Neighbors on Eavesdropper Pressure. *Front. Ecol. Evol.* **2019**, *7*, 292. [CrossRef]
31. Lilly, M.V.; Lucore, E.C.; Tarvin, K.A. Eavesdropping grey squirrels infer safety from bird chatter. *PLoS ONE* **2019**, *14*, e0221279. [CrossRef]
32. Hamilton, W.D. The genetical evolution of social behaviour. I. *J. Theor. Biol.* **1964**, *7*, 1–16. [CrossRef]
33. Roberts, G. Cooperation: How Vampire Bats Build Reciprocal Relationships. *Curr. Biol.* **2020**, *30*, R307–R309. [CrossRef] [PubMed]
34. Connor, R.C. Pseudo-reciprocity: Investing in mutualism. *Anim. Behav.* **1986**, *34*, 1562–1566. [CrossRef]
35. Xu, Y.; Pi, D. A hybrid enhanced bat algorithm for the generalized redundancy allocation problem. *Swarm Evol. Comput.* **2019**, *50*, 100562. [CrossRef]
36. Santiago, A.; Dorronsoro, B.; Fraire, H.J.; Ruiz, P. Micro-Genetic algorithm with fuzzy selection of operators for multi-Objective optimization: FAME. *Swarm Evol. Comput.* **2021**, *61*, 100818. [CrossRef]
37. Chen, X.; Tianfield, H.; Li, K. Self-adaptive differential artificial bee colony algorithm for global optimization problems. *Swarm Evol. Comput.* **2019**, *45*, 70–91. [CrossRef]
38. Tighzert, L.; Fonlupt, C.; Mendil, B. A set of new compact firefly algorithms. *Swarm Evol. Comput.* **2018**, *40*, 92–115. [CrossRef]
39. Salgotra, R.; Singh, U.; Saha, S.; Gandomi, A.H. Self adaptive cuckoo search: Analysis and experimentation. *Swarm Evol. Comput.* **2021**, *60*, 100751. [CrossRef]
40. Gupta, S.; Deep, K. A novel Random Walk Grey Wolf Optimizer. *Swarm Evol. Comput.* **2019**, *44*, 101–112. [CrossRef]
41. Skackauskas, J.; Kalganova, T.; Dear, I.; Janakiram, M. Dynamic impact for ant colony optimization algorithm. *Swarm Evol. Comput.* **2022**, *69*, 100993. [CrossRef]
42. Holland, J.H. *Adaptation in Natural and Artificial Systems*; The MIT Press: Cambridge, MA, USA, 1992. [CrossRef]
43. Theraulaz, G.; Goss, S.; Gervet, J.; Deneubourg, J.L. Task Differentiation in Polistes Wasp Colonies: A Model of Self-Organizing Groups of Robots. In *From Animals to Animats*; The MIT Press: Cambridge, MA, USA, 1991; pp. 346–355. [CrossRef]
44. Hersovici, M.; Jacovi, M.; Maarek, Y.S.; Pelleg, D.; Shtalhaim, M.; Ur, S. The shark-search algorithm. An application: tailored Web site mapping. *Comput. Netw. Isdn Syst.* **1998**, *30*, 317–326. [CrossRef]
45. Dorigo, M.; Birattari, M.; Stutzle, T. Ant Colony Optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [CrossRef]
46. Passino, K.M. Biomimicry of Bacterial Foraging for Distributed Optimization and Control. *IEEE Control Syst.* **2002**, *22*, 52–67. [CrossRef]

47. Yang, X.S.; Suash Deb. Cuckoo Search via Levy flights. In Proceedings of the 2009 World Congress on Nature and Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 210–214. [CrossRef]
48. Karaboga, D.; Basturk, B. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* **2008**, *8*, 687–697. [CrossRef]
49. Yang, X.S. Firefly Algorithms for Multimodal Optimization. In *Proceedings of the Stochastic Algorithms: Foundations and Applications (SAGA 2009), Sapporo, Japan, 26–28 October 2009*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 169–178. [CrossRef]
50. Yang, X.S. A New Metaheuristic Bat-Inspired Algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; Gonzalez, J., Ed.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74. [CrossRef]
51. Yang, X.S. Flower Pollination Algorithm for Global Optimization. In *Proceedings of the 11th International Conference Unconventional Computation and Natural Computation, Orléans, France, 3–7 September 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 240–249. [CrossRef]
52. Cui, Z.; Cai, X. Artificial Plant Optimization Algorithm. In *Swarm Intelligence and Bio-Inspired Computation*; Elsevier: Amsterdam, The Netherlands, 2013; pp. 351–365. [CrossRef]
53. Jain, M.; Singh, V.; Rani, A. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. *Swarm Evol. Comput.* **2019**, *44*, 148–175. [CrossRef]
54. Zervoudakis, K.; Tsafarakis, S. A mayfly optimization algorithm. *Comput. Ind. Eng.* **2020**, *145*, 106559. [CrossRef]
55. Ren, T.; Luo, T.; Jia, B.; Yang, B.; Wang, L.; Xing, L. Improved ant colony optimization for the vehicle routing problem with split pickup and split delivery. *Swarm Evol. Comput.* **2023**, *77*, 101228. [CrossRef]
56. Liu, J.; Anavatti, S.; Garratt, M.; Abbass, H.A. Multi-operator continuous ant colony optimisation for real world problems. *Swarm Evol. Comput.* **2022**, *69*, 100984. [CrossRef]
57. Stodola, P.; Otřísal, P.; Hasilová, K. Adaptive Ant Colony Optimization with node clustering applied to the Travelling Salesman Problem. *Swarm Evol. Comput.* **2022**, *70*, 101056. [CrossRef]
58. Zhang, Y.; Lin, Q.; Li, L.; Xiao, Z.; Ming, Z.; Leung, V.C. Multiobjective band selection approach via an adaptive particle swarm optimizer for remote sensing hyperspectral images. *Swarm Evol. Comput.* **2024**, *89*, 101614. [CrossRef]
59. Abd Elaziz, M.; Yousri, D.; Aseeri, A.O.; Abualigah, L.; Al-qaness, M.A.; Ewees, A.A. Fractional-order modified heterogeneous comprehensive learning particle swarm optimizer for intelligent disease detection in IoMT environment. *Swarm Evol. Comput.* **2024**, *84*, 101430. [CrossRef]
60. Wu, Q.; Xia, X.; Song, H.; Zeng, H.; Xu, X.; Zhang, Y.; Yu, F.; Wu, H. A neighborhood comprehensive learning particle swarm optimization for the vehicle routing problem with time windows. *Swarm Evol. Comput.* **2024**, *84*, 101425. [CrossRef]
61. Niu, B.; Tan, L.; Liu, J.; Liu, J.; Yi, W.; Wang, H. Cooperative bacterial foraging optimization method for multi-objective multi-echelon supply chain optimization problem. *Swarm Evol. Comput.* **2019**, *49*, 87–101. [CrossRef]
62. Sudhakar Babu, T.; Priya, K.; Maheswaran, D.; Sathish Kumar, K.; Rajasekar, N. Selective voltage harmonic elimination in PWM inverter using bacterial foraging algorithm. *Swarm Evol. Comput.* **2015**, *20*, 74–81. [CrossRef]
63. Panda, R.; Naik, M.K.; Panigrahi, B. Face recognition using bacterial foraging strategy. *Swarm Evol. Comput.* **2011**, *1*, 138–146. [CrossRef]
64. Mohapatra, P.; Chakravarty, S.; Dash, P. An improved cuckoo search based extreme learning machine for medical data classification. *Swarm Evol. Comput.* **2015**, *24*, 25–49. [CrossRef]
65. Majumder, A.; Laha, D. A new cuckoo search algorithm for 2-machine robotic cell scheduling problem with sequence-dependent setup times. *Swarm Evol. Comput.* **2016**, *28*, 131–143. [CrossRef]
66. Trachanatzi, D.; Rigakis, M.; Marinaki, M.; Marinakis, Y. A modified Ant Colony System for the asset protection problem. *Swarm Evol. Comput.* **2022**, *73*, 101109. [CrossRef]
67. Kar, A.K. Bio inspired computing—A review of algorithms and scope of applications. *Expert Syst. Appl.* **2016**, *59*, 20–32. [CrossRef]
68. Shi, Y.; Eberhart, R.C. Parameter selection in particle swarm optimization. In *Proceedings of the Evolutionary Programming VII, San Diego, CA, USA, 25–27 March 1998*; Porto, V.W., Saravanan, N., Waagen, D., Eiben, A.E., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; pp. 591–600.
69. Kohler, M.; Vellasco, M.M.; Tanscheit, R. PSO+: A new particle swarm optimization algorithm for constrained problems. *Appl. Soft Comput.* **2019**, *85*, 105865. [CrossRef]
70. Xue, B.; Zhang, M.; Browne, W.N. Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Appl. Soft Comput.* **2014**, *18*, 261–276. [CrossRef]
71. Wan, Z.; Wang, G.; Sun, B. A hybrid intelligent algorithm by combining particle swarm optimization with chaos searching technique for solving nonlinear bilevel programming problems. *Swarm Evol. Comput.* **2013**, *8*, 26–32. [CrossRef]
72. Das, P.; Behera, H.; Panigrahi, B. A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. *Swarm Evol. Comput.* **2016**, *28*, 14–28. [CrossRef]
73. Jiang, B.; Wang, N.; Wang, L. Particle swarm optimization with age-group topology for multimodal functions and data clustering. *Commun. Nonlinear Sci. Numer. Simul.* **2013**, *18*, 3134–3145. [CrossRef]
74. Kang, L.; Chen, R.S.; Cao, W.; Chen, Y.C. Non-inertial opposition-based particle swarm optimization and its theoretical analysis for deep learning applications. *Appl. Soft Comput.* **2020**, *88*, 106038. [CrossRef]

75. Khurana, M.; Massey, K. Swarm algorithm with adaptive mutation for airfoil aerodynamic design. *Swarm Evol. Comput.* **2015**, *20*, 1–13. [CrossRef]
76. Wang, X.; Tang, L. A discrete particle swarm optimization algorithm with self-adaptive diversity control for the permutation flowshop problem with blocking. *Appl. Soft Comput.* **2012**, *12*, 652–662. [CrossRef]
77. Zervoudakis, K.; Mastrothanasis, K.; Tsafarakis, S. Forming automatic groups of learners using particle swarm optimization for applications of differentiated instruction. *Comput. Appl. Eng. Educ.* **2020**, *28*, 282–292. [CrossRef]
78. Tian, D.; Zhao, X.; Shi, Z. DMPSO: Diversity-Guided Multi-Mutation Particle Swarm Optimizer. *IEEE Access* **2019**, *7*, 124008–124025. [CrossRef]
79. Tian, D.; Xu, Q.; Yao, X.; Zhang, G.; Li, Y.; Xu, C. Diversity-guided particle swarm optimization with multi-level learning strategy. *Swarm Evol. Comput.* **2024**, *86*, 101533. [CrossRef]
80. Wickelgren, W.A. Speed-accuracy tradeoff and information processing dynamics. *Acta Psychol.* **1977**, *41*, 67–85. [CrossRef]
81. Varna, F.T.; Husbands, P. BIS: A New Swarm-Based Optimisation Algorithm. In Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, 1–4 December 2020; IEEE: Piscataway, NJ, USA, 2020, pp. 457–464. [CrossRef]
82. Liang, J.; Qu, B.; Suganthan, P. *Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization, Vol. 34: Technical Report 201212*; Technical Report; Computational Intelligence Laboratory, Zhengzhou University: Zhengzhou, China; Nanyang Technological University: Singapore, 2013.
83. Liang, J.; Qu, B.; Suganthan, P. *Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*; Technical Report; Computational Intelligence Laboratory, Zhengzhou University: Zhengzhou China; Nanyang Technological University: Singapore, 2014.
84. Awad, N.; Ali, M.; Suganthan, P.; Liang, J.; Qu, B. *Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*; Technical Report; School of EEE, Nanyang Technological University: Singapore; School of Computer Information Systems, Jordan University of Science and Technology: Ar-Ramtha, Jordan; School of Electrical Engineering, Zhengzhou University: Zhengzhou, China, 2016.
85. Yue, N.; Yue, P.; Price, K.; Suganthan, P.; Liang, J.; Ali, M.; Qu, B. *Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization*; Technical Report; Computational Intelligence Laboratory, Zhengzhou University: Zhengzhou China, 2019.
86. Parsopoulos, K.E.; Vrahatis, M.N. Unified Particle Swarm Optimization for Solving Constrained Engineering Optimization Problems. In *Proceedings of the Advances in Natural Computation, Changsha, China, 27–29 August 2005*; Wang, L., Chen, K., Ong, Y.S., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 582–591.
87. Coello, C. Self-adaptive penalties for GA-based optimization. In Proceedings of the Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; Volume 1, pp. 573–580. [CrossRef]
88. Tanabe, R.; Fukunaga, A.S. Improving the search performance of SHADE using linear population size reduction. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1658–1665. [CrossRef]
89. Kennedy, J. Bare bones particle swarms. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706), Indianapolis, IN, USA, 26 April 2003; IEEE: Piscataway, NJ, USA, 2003; pp. 80–87. [CrossRef]
90. Settles, M.; Soule, T. Breeding swarms. In Proceedings of the 2005 Conference on Genetic and Evolutionary Computation—GECCO'05, Washington, DC, USA, 25–29 June 2005; p. 161. [CrossRef]
91. Mendes, R.; Kennedy, J.; Neves, J. The Fully Informed Particle Swarm: Simpler, Maybe Better. *IEEE Trans. Evol. Comput.* **2004**, *8*, 204–210. [CrossRef]
92. Peram, T.; Veeramachaneni, K.; Mohan, C. Fitness-distance-ratio based particle swarm optimization. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706), Indianapolis, IN, USA, 26 April 2003; IEEE: Piscataway, NJ, USA, 2003, pp. 174–181. [CrossRef]
93. Parsopoulos, K.; Vrahatis, M. UPSO: A Unified Particle Swarm Optimization Scheme. In *International Conference of Computational Methods in Sciences and Engineering 2004 (ICCMSE 2004)*; CRC Press: Boca Raton, FL, USA, 2019; pp. 868–873. [CrossRef]
94. Lynn, N.; Suganthan, P.N. Ensemble particle swarm optimizer. *Appl. Soft Comput.* **2017**, *55*, 533–548. [CrossRef]
95. Ratnaweera, A.; Halgamuge, S.; Watson, H. Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients. *IEEE Trans. Evol. Comput.* **2004**, *8*, 240–255. [CrossRef]
96. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the IEEE International Conference on Evolutionary Computation 1998, Anchorage, AK, USA, 4–9 May 1998; IEEE: Piscataway, NJ, USA,1998. [CrossRef]
97. Hasanzadeh, M.; Meybodi, M.R.; Ebadzadeh, M.M. Adaptive Parameter Selection in Comprehensive Learning Particle Swarm Optimizer. In *Artificial Intelligence and Signal Processing*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 267–276. [CrossRef]
98. Beskirli, A.; Dag, I. I-CPA: An Improved Carnivorous Plant Algorithm for Solar Photovoltaic Parameter Identification Problem. *Biomimetics* **2023**, *8*, 569. [CrossRef] [PubMed]
99. Derrac, J.; Garcia, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [CrossRef]
100. Varna, F.T. Design and Implementation of Bio-Inspired Heterogeneous Particle Swarm Optimisation Algorithms for Unconstrained and Constrained Problems. Ph.D. Thesis, University of Sussex, Brighton, UK, 2023.

101. Kumar, A.; Wu, G.; Ali, M.Z.; Mallipeddi, R.; Suganthan, P.N.; Das, S. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm Evol. Comput.* **2020**, *56*, 100693. [CrossRef]
102. Li, W.; Meng, X.; Huang, Y.; Fu, Z.H. Multipopulation cooperative particle swarm optimization with a mixed mutation strategy. *J. Inf. Sci.* **2020**, *529*, 179–196. [CrossRef]
103. Wood, D. Hybrid cuckoo search optimization algorithms applied to complex wellbore trajectories aided by dynamic, chaos-enhanced, fat-tailed distribution sampling and metaheuristic profiling. *Nat. Gas Sci. Eng.* **2016**, *34*, 236–252. [CrossRef]
104. Husbands, P.; Philippides, A.; Vargas, P.; Buckley, C.; Fine, P.; DiPaolo, E.; O'Shea, M. Spatial, temporal, and modulatory factors affecting GasNet evolvability in a visually guided robotics task. *Complexity* **2010**, *16*, 35–44. [CrossRef]
105. Edelman, G.; Gally, J. Degeneracy and complexity in biological systems. *Proc. Natl. Acad. Sci. USA* **2001**, *98*, 13763–13768. [CrossRef] [PubMed]
106. Whitacre, J.; Bender, A. Degeneracy: A design principle for achieving robustness and evolvability. *J. Theor. Biol.* **2010**, *263*, 143–153. [CrossRef] [PubMed]
107. Chauhan, D.; Yadav, A. Optimizing the parameters of hybrid active power filters through a comprehensive and dynamic multi-swarm gravitational search algorithm. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106469. [CrossRef]
108. Wolpert, D.H.; Macready, W.G. No Free Lunch Theorems for Optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]