Swarm and Evolutionary Computation 70 (2022) 101040

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Research Paper

Beyond exploitation: Measuring the impact of local search in swarm-based memetic algorithms through the interactions of individuals in the population

Clodomir Santana[a,*], Marcos Oliveira[a], Carmelo Bastos-Filho[b], Ronaldo Menezes[a]

[a] Department of Computer Science, University of Exeter, Exeter, UK
[b] Department of Computer Engineering, University of Pernambuco, Recife, BR

ARTICLE INFO

Keywords:
Memetic algorithms
Convergence
Exploration-Exploitation
Interaction networks
Artificial Bee Colony
Particle Swarm Optimization

ABSTRACT

Memetic algorithms are known for their enhanced solution refinement capabilities. These capabilities are a result of incorporating local-search methods into population-based metaheuristics such as swarm and evolutionary algorithms. However, designing a memetic algorithm is not a trivial task. The inclusion of local-search procedures must consider the exploitation–exploration balance and its interplay with other algorithm operators. Due to these variables, there is no universal methodology to design a memetic algorithm. Although previous works have investigated the impact of local search procedures on genetic and ant colony algorithms, we have limited knowledge about the impact of these procedures on other types of swarm-based algorithms. For swarm-based algorithms, the interactions within the population are vital to the emergence of collective intelligence and shape the algorithm's behaviour. Here, we model these interactions into a network and analyse the impact of local search in swarm-based algorithms. We selected the Particle Swarm Optimization (PSO), the Artificial Bee Colony (ABC), and one memetic version of each algorithm as a case study. We examined the effects of the modifications proposed in the memetic variants. The results obtained indicate that the networks of interactions capture several characteristics of the algorithms and the impact of the local search strategies. The impact of local search operators can be gauged by the temporal analysis of the changes in the structural properties of the algorithm's network (e.g. study of the weight and distribution of the network's connections). These changes are linked to the algorithms' convergence signature and can be used as a proxy to assess the differences between the algorithms studied and their memetic versions.

1. Introduction

Memetic Algorithms (MAs) were first introduced by Moscato et al. [1,2] initially as a class of algorithms that combines evolutionary algorithms with local search methods. The term "memetic" comes from Richard Dawkins concept of *memes* as a unity of cultural evolution which can be propagated and exhibit local enhancements [3,4]. These algorithms were first proposed and extensively applied to improve solution refinement in genetic algorithms [4]. Although it is a simple concept, developing a novel memetic algorithm is challenging because it requires more than just combining a metaheuristic with a local search operator. It involves crucial factors, such as selecting the proper local search operator that interplays with the other operators in the metaheuristic. Despite advances in the field, however, there is still no consolidating methodology for developing new MAs and assessing the impact of the local search operators.

After the initial focus on genetic algorithms, MAs were extended to improve the local search capabilities of not only evolutionary strategies but also swarm-based and other population-based metaheuristics [5]. Moreover, several types of local search strategies such as hill-climbing [6], simulated annealing [7], tabu search [8], and pattern search [9] were also employed to improve the solution refinement capabilities of the swarm-based metaheuristics.

One of the reasons behind the increasing popularity of MAs is their performance on high-dimensional and non-differentiable search spaces [5] and their enhanced exploitation capabilities [10]. As a result, we can see a variety of papers proposing or improving or using memetic algorithms for discrete, continuous, constrained, multi-objective and other classes of optimisation problems [11]. We can also find examples of applications of MAs in several real-world problems; for example, circuit layout [12], automatic test suite generation [13], capacitated arc routing [14], community detection in networks [15], and job shop scheduling [16].

---

* Corresponding author.
E-mail address: csantana@biocomplexlab.org (C. Santana).

https://doi.org/10.1016/j.swevo.2022.101040

Concerning the study of MAs' capabilities, studies report that monitoring population convergence while preserving the diversity of the solutions is a crucial factor to avoid sub-optimal solutions and continuously discover better solutions [11]. The work of Ong et al. [17] presents an example of the convergence analysis of memetic algorithms. In their study, Ong et al. employed Markov chains theory to extend previous efforts on convergence analysis of GAs. Moreover, Moscato and Cotta [18] argue that a population restarting procedure is an essential and recurrent strategy adopted by MAs to overcome premature convergence to sub-optimal solutions.

In addition to the convergence speed, all the population-based algorithms must manage the balance between exploration-exploitation of the solutions [11]. On MAs, this balance depends on several aspects, such as the interplay between the local search procedure and the other features of the algorithm [19]. Despite the efforts to analyse convergence, exploration and exploitation on memetic algorithms, there is still no consensus of a unified methodology, and new methods continue to emerge.

Furthermore, because of the high number of variables involved in the process of designing a MA, the choice of the local search strategy and the definition of its execution frequency in many cases are considered an "art" [20]. Even though some works have shown that the effects of the local search might differ depending on the algorithm used [21,22], the impact of local search methods in other types of MAs is still unclear, and new studies are necessary in these cases.

Previous works have successfully applied a network-based approach to study stagnation, convergence, exploration and exploitation of particle swarm optimisers aiming to reduce this research gap [23,24]. This approach, named interaction networks, relies on the collective intelligence that emerges from the interactions patterns within the population and allows simple reactive individuals to tackle complex tasks. The interaction networks capture the population interactions in networks where the nodes represent individuals from the population, and links are created when individuals interact. For example, an interaction can be when an individual shares information about its current location or the location of optimal best it found.

In this paper, we propose modifications to the interaction network creation methodology and analyse the characteristics of these networks using the portrait divergence [25] and the interaction diversity [26] metric. As a case study, we selected the artificial bee colony (ABC), particle swarm optimization (PSO), and two memetic variations of these algorithms. Besides being two prominent optimisation algorithms from the swarm intelligence family, we selected the ABC and PSO due to their differences in behaviour. The PSO generally presents high exploitation capability [27]. If not done correctly, integrating an additional local search operation into the PSO can lead to premature convergence. In contrast, the ABC is known to be more efficient in exploration than exploitation [28]. For this reason, the memetic variant of this algorithm has the potential to improve its solution refinement capabilities. We show how the interaction networks can encode characteristics of these swarm-based algorithms. Furthermore, we employ these networks to examine the enhanced exploitation capabilities, faster convergence of memetic algorithms, and the changes caused by introducing the local search in the selected memetic ABC and PSO variants. Although we are conducting our analysis using the PSOs and ABCs, the interaction networks have been applied to other types of swarm-based algorithms [29–31]. By employing these networks, we can model and monitor direct and indirect interactions patterns between individuals. We can also study the changes in these patterns during the optimisation process, under different optimisation problems, or when different parameter values/operators are used.

The remainder of this paper is structured as follows. Section 2 presents the memetic algorithms studied, the interaction networks and the metrics used to assess the networks. Next, in Section 3, we describe the experiments performed and analyse the results. Lastly, Section 4 summarise the paper and presents the conclusions.

2. Theoretical background

This section briefly describes the theoretical background of this research. It includes the explanation on the interaction networks and the modifications proposed to them (Section 2.1), the portrait divergence and interaction diversity metrics (Section 2.2), and the memetic algorithms selected (Section 2.3).

2.1. The interaction networks

When dealing with population-based algorithms, the interactions between the individuals within the population result in the emergence of a global intelligence which is vital to the algorithm's performance [32]. These interactions happen between individuals that collaborate or compete to achieve a particular goal. In both the collaborative and competitive interactions, the individuals frequently share information related to the quality and location of solutions found. Given the importance of these interactions to the performance of population-based, Oliveira et al. proposed in 2014 a technique to represent those interactions as a structure called interaction networks [33]. The interaction networks are graphs where the nodes (or vertices) represent the individuals from the algorithm's population. The connections (or edges) indicate interactions that happened during the optimisation process. The definition of the interaction networks (I) is shown by Equation (1)

$$I_t^w = \sum_{t'=t}^{t+w-1} I(t'), \tag{1}$$

where t is a given iteration, w is the size of the time window, $T - t \geq w \geq 1$ and T is the total number of iterations. The process to create these networks is illustrated in Fig. 1. It is worth mentioning that, when the size of the time window is equal to one, the networks in Fig. 1 (2) are the same as the ones in Fig. 1 (3).

The literature has works that use this approach to study several aspects of swarm-based algorithms; for example, their characteristics [23,31,34], convergence and exploration-exploitation pace [24,26]. Also, recent works demonstrated the application of this approach to the analysis of several population-based algorithms [29,30]; however, to the best of our knowledge, the application of interaction networks on the evaluation of memetic algorithms is a novelty.

Another contribution of this work is a modification proposed in the process to create the interaction networks. Unlike previous works that create/update the connections using a fixed weight value (usually equal one), here we are exploring a different strategy that uses the distance between the nodes as the weight of the connection. In this way, the strength of connections will reflect how spatially close the nodes are and, hence, the degree of similarity between the solution that they represent.

Furthermore, with this edge weight representation, we can better understand the convergence status of the swarm by analysing the evolution of the weight of the connections over time. We also define that in the interaction network, besides the label, all nodes will have a value that represents the fitness of the individual when the connection was created/updated. This information can give us insights related to the status of the swarm.

2.2. Selected metrics

To compare the swarm interaction networks, we selected the *portrait divergence* (PD), which is a graph-invariant metric that measures the structural similarities between networks regardless of their sizes and of them being directed or undirected [35].

We also employ the Interaction Diversity [29]: a metric that quantifies the diversity of the information flow in a network. This measurement can be linked to convergence/stagnation and to analyse the balance exploration/exploitation in the swarm [24,26,29].

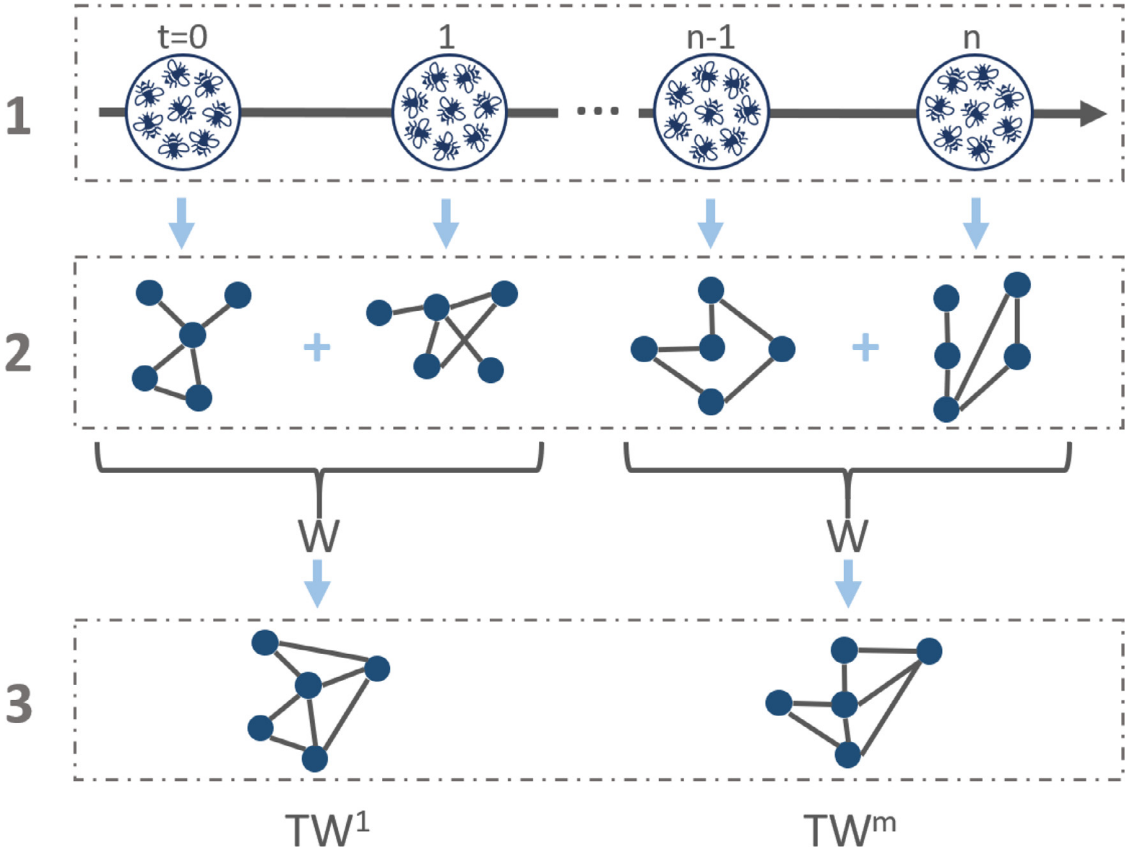

Fig. 1. Creation of the swarm interaction network. The first level (1) represents the state of the swarm at each iteration of the algorithm. Level (2) has the interaction networks created based on the information from level 1. In the last level (3), we have the networks used to analyse the swarm. We generate these networks by combining a set of consecutive networks from level 2. The number of networks combined is determined by the size of the time window used (w). In this example, the time windows have a size equal to two, and the total number of networks (m) in the third level is $n/2$.

2.2.1. *Portrait divergence*

The portrait divergence (PD) stands out for working with networks with different sizes, direct/undirected, weighted/unweighted, and label invariant. Based on the principle that a network can be represented entirely by its adjacency matrix (i.e. an N by N matrix whose values represents the weight of the connections between the set of nodes). Using the edges/nodes information the *network portrait*, also known as B-matrix, is calculated and encodes the structural information of a network [25]. The B-matrix matrix is calculated as indicated by Equation (2)

$$B_{\ell,k} = NP_{\ell}(k), \tag{2}$$

where $B_{\ell,k}$ represents that the ℓ^{th} row of the B-matrix, N is the number of nodes in the network, and P is a network invariant — property of the network that holds for all the possible networks with permuted labels — of the network. It indicates the number of nodes in the network that have k neighbours at a distance equal to ℓ. This distance is calculated by counting the minimum number of edges (shortest path) that connects the two nodes. The B-matrix disregards the nodes' labels and can represent direct and undirected networks. Because of this property of the B-matrix, the PD is a label-invariant metric that works with directed and undirected networks. Concerning the weighted networks, to create the B-matrix, a binning strategy to estimate the distances is used. This strategy also has a low computational cost when dealing with small and medium-sized networks [35].

Based on the B-matrices, the PD metric employs the Jensen-Shannon divergence to measure the distance between two portraits:

$$PD(N_1, N_2) = \frac{1}{2}KL(P_{N_1}||M) + \frac{1}{2}KL(P_{N_2}||M), \tag{3}$$

where N_1, N_2 are two networks, P_{N_1} and P_{N_2} are the network invariant of N_1 and N_2 respectively, $M = \frac{1}{2}(N_1 + N_2)$, and $KL(\cdot||\cdot)$ is calculated according to Eq. 4.

$$KL(P_{N_1}||P_{N_2}) = \sum_{\ell=0}^{max(d_1,d_2)} \sum_{k=0}^{N} P_{N_1}(k,\ell) log\left(\frac{P_{N_1}(k,\ell)}{P_{N_2}(k,\ell)}\right), \tag{4}$$

where d_1 and d_2 are the diameter (longest shortest path in the network) of N_1 and N_2. PD gives a value between zero and one, where zero means that the matrices compared are identical, and one means that they are different.

An example of the effectiveness of the PD to measure the differences between the interaction networks of algorithms with different characteristics is presented in Fig. 2. In this example, we used the PD to compare 30 networks of the ABC and PSO. Note that bluish colours mean that the networks compared are structurally similar. In contrast, reddish colours indicate that the networks have different structures.

The comparison of networks for the same algorithms (e.g. Fig. 2 A and B) using the PD metric reveals values that indicate the similarities between the structure of the networks (PD values close to 0). Nevertheless, when we compare the networks from algorithms with different characteristics (Fig. 2 C), the PD values are closer to 1, indicating the differences in the networks' structure. It is worth mentioning that previous studies have shown that, depending on the characteristics of the application, the limit used to characterise two networks as similar or dissimilar varies [25,35]. However, because the investigation of this limit for the interaction networks is beyond the scope of this work, we opted to perform it in a future study.

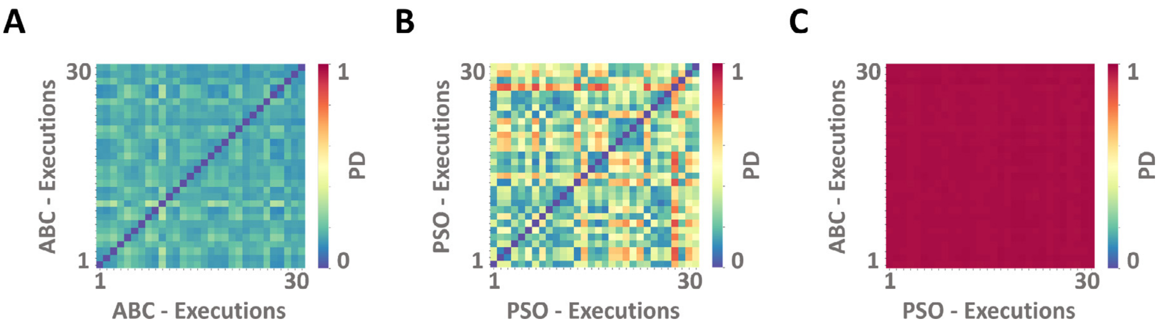

Fig. 2. Application of the PD metric to measure the similarities of the networks for the PSO and the ABC algorithms in 30 independent executions. In this example, both algorithms were executed with a population of 100 individuals in the Sphere benchmark function with 50 dimensions. The stop criteria used was 500 iterations and the time window has a size equal to 10. In these heat maps, the blue colour indicates that the networks have a similar structure (PD values close to 0). As we can see, the comparisons of the networks for the same algorithms (A and B) in most cases result in PD values closer to 0 than to 1, indicating the similarity of the networks. However, when we compare two algorithms with distinct interaction patterns, such as the ABC and the PSO (C), we can see that the PD values are close to 1, indicating the dissimilarity between the networks.

2.2.2. Interaction diversity

Interaction diversity (ID) is a metric proposed by Oliveira et al. [26] which is used to measure the diversity of the information flow of the swarm. Because the information flow is linked to the spatial distribution of the solutions in the swarm, this metric can help detect stagnation/convergence, exploitation and exploration.

The idea behind the ID calculation is that the more diverse the flow of information the swarm is, the more connected the nodes in the interaction network will be. Hence, if several isolated sub-networks are created when we remove a fraction of the weakest links (i.e. lowest edge weights), the network is not well connected (low diversity). In other words, we can measure diversity by analysing how easily the network can be divided into sub-networks when we remove a certain number of edges.

To calculate the interaction diversity of a given network, we use the following equation:

$$ID_t = 1 - \frac{A(I_t)}{|S|}, \tag{5}$$

where $|S|$ represents the size of the population (i.e., the number of nodes in the network) and $A(t)$ is the area under the destruction curve for the network I_t. The area is calculated employing the composite trapezoidal rule. The destruction curve where each point describes the number of disconnected sub-networks (isolated components) created by removing a fraction e of the weakest edges in the network, and the value of e varies from 0 to 1.

In the literature, we can find researches that applied the interaction diversity to analyse stagnation phenomenon on different communication topologies for the PSO [26], study the exploitation-exploration balance in the PSO [24], and compare the exploitation-exploration of different swarm-based algorithms [29]. In this work, we will employ this metric to analyse the exploitation-exploration of the algorithms selected.

2.3. Selected memetic algorithms

As a case study for the analysis of convergence, exploration-exploitation capabilities and the effects of modifications made on MAs, we selected two memetic versions of the two most prominent swarm-based algorithms: the PSO and the ABC. As aforementioned, these algorithms have distinct search behaviours. The ABC is known more for its exploration capabilities, while the PSO has enhanced exploitation. Section 2.3.1 describes the memetic PSO selected while Section 2.3.2 details the memetic ABC. It is worth mentioning that these algorithms were selected as a case study. The interaction networks can be extended to other memetic algorithms and a variety of other population-based algorithms, as demonstrated in previous works [29–31].

2.3.1. PSO with pattern search

The first memetic algorithm chosen is a variant of the PSO with the pattern search (PS) [36] as its local search method producing the PSO-PS [5]. It was selected due to the strategy that the authors adopted to propose this algorithm which does not change the PSO's rules for updating the particle's position. Instead, it includes an additional step that performs the local search around every particle in a selected group. This group is mainly composed of the best particles in the swarm. Algorithm 1 describes how the PSO-PS works.

Algorithm 1. PSO with Pattern Search

```
Initialise all particles' positions and velocity randomly;
Evaluate the swarm and update the best position;
while stop criterion is not reached do
    \\PSO Phase
    for each particle $i = 1, ..., SwarmSize$ do
        Update particle's velocity and position;
        Evaluate the new position;
        if new position is better than the pbest then
            Update the personal best position (pbest);
        end
    end
    Update the global best solution;
    \\Memetic Phase
    Apply the probabilistic selection strategy to select the particles that will
      perform the memetic phase;
    Perform the pattern search on the positions of the selected particles;
    Update the best solution;
end
Return the best solution.
```

2.3.2. ABC with golden section search

The second algorithm selected was the memetic ABC (MeABC) [37], which incorporates the Golden Section Search (GSS) [38] to enhance the exploitation capabilities of the ABC. In contrast to the PSO-PS, the MeABC includes a memetic phase (as in PSO-PS) and modifies the solutions' update rule by adding the current global best solution when updating the employed and onlooker bees positions. Equation (6) illustrates these changes

$$v_{i,d} = x_{i,d} + \phi(x_{i,d} - x_{j,d}) + \psi(x_{best,d} - x_{j,d}), \tag{6}$$

where $x_{i,d}$ is the current position of the food source i in the dimension d, j is a food source selected accordingly to the employed or onlooker selection rule, ϕ is a uniformly generated random number between -1 and 1, and ψ is a uniformly generated random number between 0 and c, where c is one parameter of the MeABC. In the memetic phase, the GSS is used to exploit only the current best solution in the swarm. Algorithm 2 describes the MeABC.

Algorithm 2. Memetic ABC

```
Initialise all the food sources' with random positions and calculate the fitness
  of them;
Send the employed bees to the food sources;
while stop criterion is not reached do
    \\ABC' Phase
    for each employed bee $i = 1, ..., SwarmSize$ do
        Select a random food source $j$ from the list o food sources;
        Find a new food source in its neighbourhood using Equation (6) and
          evaluate its fitness;
        if fitness of new food source $>$ current food source's fitness then
            Move the food source location to the new food source;
        end
        else
            Increment number of trials of food source $i$;
        end
    end
    Calculate the probability $p_i$ of each food source;
    for each onlooker bee do
        for each food source $i = 1, ..., SwarmSize$ do
            Generate $r = rand(0, 1)$;
            if $r > p_i$ then
                Send current onlooker to the $i$-th food source;
                Find a new food source and evaluate its fitness;
                if fitness of new food source $>$ current food source's fitness
                  then
                    Move the food source location to the new food source;
                end
                else
                    Increment number of trials of food source $i$;
                end
            end
        end
    end
    if number of trials of one of the food source $\geq$ trials limit then
        Generate a random food source and calculate its fitness;
    end
    \\Memetic Phase
    Perform the GSS on the position of the current best solution;
    if new solution is better than the current one then
        Update the best solution;
    end
end
Return the best food source;
```

3. Experiments and results

We implemented the memetic PSO using a global best topology, and we adopted the recommended values for the parameters described in the proposal paper [5]. The values of the parameters are: adopted $c_1 = c_2 = 1.496$ and a linear decrease for the inertia factor (from 0.9 to 0.4). The pattern search of the PSO-PS had a radius of search of 2.0, and the initial delta was 1.0. We used a population of 100 particles and 500 iterations as the stop criteria in all experiments performed. Furthermore, in each experiment, we performed 30 independent simulations of the algorithms. The same values were used for the standard PSO.

Regarding the parameter selection for the ABC and MeABC, we used a population composed of 50 employees and 50 onlookers bees (when necessary, one of the onlookers acts as the scout bee), and the stop criteria were 500 iterations (as in the PSO and PSO-PS). The trials limit for the ABC and MeABC was set as 10, and the MeABC used $c = 1.5$, stop criteria for the GSS as $epsilon < 0.01$ and the selection threshold equals to 0.4. Again, these values were based on the recommended values in their respective proposal papers. We also performed 30 simulations of these algorithms in every experiment conducted.

This section contains experiments to explain the relationship between the interaction networks and the algorithms they represent and how these networks can help us understand phenomena such as convergence, exploitation, and exploration. Section 3.1 explains how the structure of interaction networks reflects the features of the algorithms that they represent. In Section 3.2, we perform experiments to study the exploration-exploitation using the networks. Lastly, Section 3.3 shows that interaction networks can also be used to assess the convergence of the algorithms.

In some experiments performed, we base the analysis using the Sphere function as a case study. We acknowledge that this is an easy function and that the potential of the memetic algorithms would be better assessed in more challenging scenarios. Nevertheless, we selected the Sphere because it is a more didactic example, and its results can be explained more simply. Also, as mentioned before, several works have focused on the performance of MAs in challenging problems. Furthermore, even when we base the initial analysis using the Sphere, we later show the results for other benchmark functions. The following list summarises the investigations conducted in this study, describing the goals and the expected results of each analysis.

- **Characterisation of the Interaction Networks**: in this step, the study the characteristics of the network and how it changes along with the optimisation process. We observe aspects such as the network degree distribution, hubs and clusters' evolution, the evolution of the weight of the connections and compare the networks generated in different stages of the optimisation. One of the goals of this step is to validate the model by observing if the networks capture the characteristics of the algorithms they represent. For example, algorithms that elect leaders from the population that guides the search process, resulting in networks with hubs (i.e. presence of nodes that poses a large number of connections), algorithms that operate with sub-populations have networks with clusters (i.e. highly connected sub-networks). Another goal is to study how the networks evolve throughout the optimisation, measuring the structural differences from the networks at distinct stages. This study allows us, for example, to identify the convergence patterns of the algorithms. Furthermore, we can assess the similarities between different algorithms by comparing their convergence patterns.
- **Assessment of Exploration-Exploitation**: to analyse the exploration and exploitation behaviour of the algorithms from their respective interaction networks, we can study the evolution of the edge weight and node degree of the networks. Because the edge weight is associated with the spatial proximity of the swarm and the node degree is linked to the amount of interaction, these indicators can give insights into the exploration/exploitation. For example, when an algorithm such as the PSO explores a solution, the network connections will be concentrated in a reduced number of nodes (i.e. decrease in the average node degree). Alternatively, the ID metric can also be employed to perform these analyses.
- **Analysis of Convergence**: for this analysis, first, we should make sure that the algorithms had the opportunity to converge or go through multiple cycles of convergence. A simple way to ensure that is to select and less complex problem with a reduced number of dimensions. After that, the study of the PD convergence patterns of the algorithms reals how the networks converge, if it has a mechanism to increase population diversity and avoid premature convergence, and, comparing two algorithms, we can identify which one has faster convergence. In these experiments, we expect the memetic variants to converge faster than the original one. Also, because of the presence of the scout bee on the ABCs, we hope to see different convergence cycles.

3.1. Characterisation of the algorithms' interaction networks

The analysis of the algorithms behaviour and the characteristics of the networks generated was based on their results with the Sphere benchmark function [39]. We adopted the number of dimensions equal to 50, and for each algorithm, we performed 30 independent executions using 500 iterations as the stop criteria.

To generate the interaction networks used in this analysis, we divided the networks created by 500 iterations into 50 groups of 10 iter-

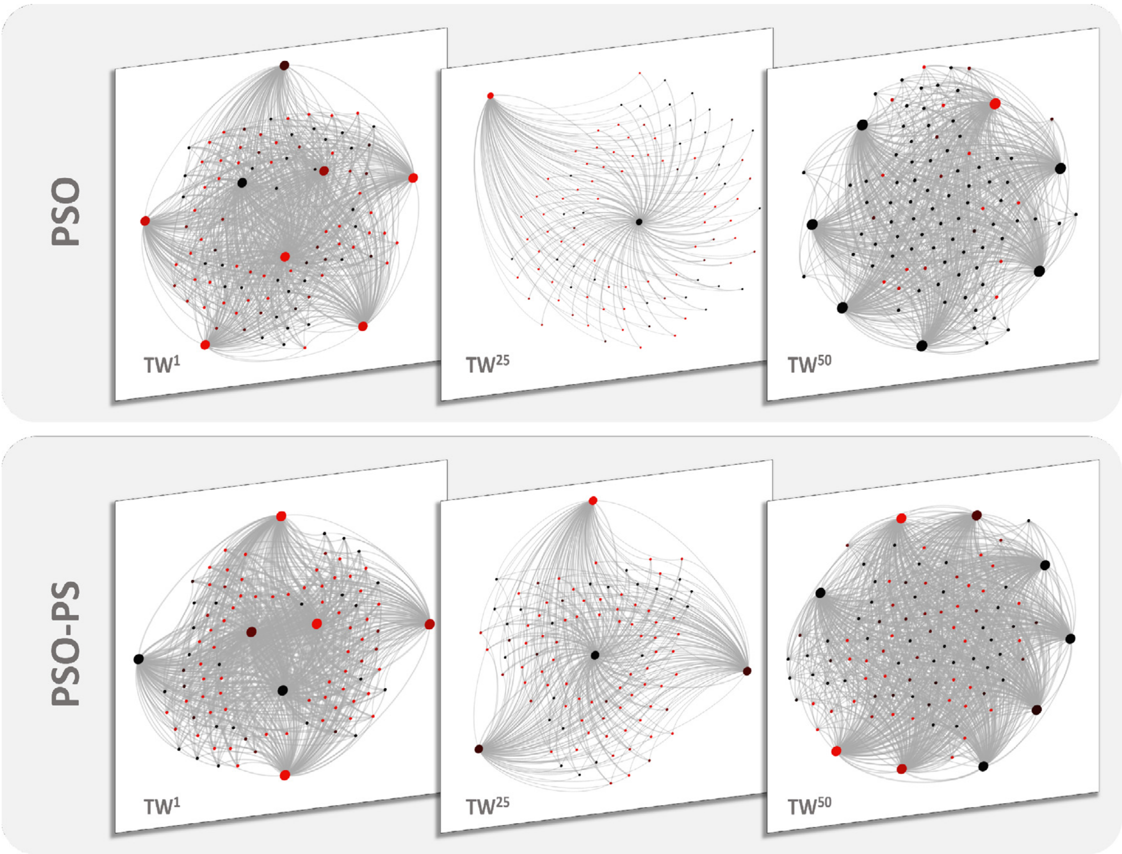

Fig. 3. Examples of integration networks generated for the PSO and PSO-PS at the beginning (TW^1), middle (TW^{25}) and end (TW^{50}) of the optimisation process. The colour represents the fitness of the node (the darker the colour, the best is the represented by the node), size is proportional to the degree of influence of the node (i.e. the number of connections that point inward at the node).

ations. Then, we added together the adjacency matrices of the items in each group to generate a single structure that encodes the interaction patterns of a specific time window. Fig. 3 and Fig. 4 presents, respectively, the interaction networks for the ABC and MeABC, and PSO and PSO-PS in the beginning (TW^1), middle (TW^{25}) and end (TW^{50}) of the optimisation process.

As we can see in Fig. 3, the networks for both the PSO and the PSO-PS are characterised by the presence of hubs (highest connected nodes) which represents the global best individuals in the swarm. Because a single time window can comprise several iterations, it is possible to have more than one hub in the network. In PSO/PSO-PS, there is a relationship between the number of hubs in the network and the convergence of the swarm. At the beginning of the execution, because the solutions are generated randomly across the search space, it is relatively easy to improve them, leading to the creation of new leaders.

Similarly to the initial scenario (TW^1), the last time window (TW^{50}) have a high number of nodes acting as hubs in the network; however, in this case, the creation of hubs is due to the spatial proximity of particles as a result of the swarm converging to an optimum position. Because the particles are closer to each other, their fitness tends to be similar, and minor improvements can result in new leaders (i.e. hubs).

The main difference between the network in the first and last time window is related to the fitness of the hubs. Initially, the minority of the nodes has a good quality (few hubs with black colour); however, in the last time window (TW^{50}), for both PSO and PSO-PS, we see that the majority of the hubs have a black colour. In other words, for algorithms, the convergence of the swarm is reflected in a network with a high number of bubs with similar quality. Moreover, in the last time widows, we can see that most of the nodes in the network have a dark colour, which is also different from the first time window.

Differently from the TW^1 and TW^{50}, Fig. 3 indicates that for TW^{25}, we have fewer hubs when compared to TW^1 and TW^{50}. The low number of hubs might be because they started to converge in this stage of the optimisation process. As the initial solution set is refined, some become prominent and exploited by the other. Since the remaining particles in the swarm are not necessary spatially close to the leader, the probability of overcoming the global best is not high. This scenario changes as particles start to converge, generating a network similar to the one in TW^{50}.

In Fig. 4, the colour of the nodes indicates the quality of the solutions. We can see that both the ABC and its memetic version initially present a balance in the swarm related to the quality of the solution (the number of red nodes is similar to the number of black nodes). This balance is an indication that, in both algorithms, there is a high degree of diversity in the set of initial solutions. A high level of diversity in the swarm is vital to explore better the search space in the first iterations of the algorithm. It also can reduce the risk of premature convergence due to a lack of diversity in the swarm.

The main difference between the networks of ABC and MeABC is the presence of hubs in the networks of the latter. As can be seen in Fig. 4, in the MeABC, for all values of TW, we can see the presence of one or more nodes that poses a high number of connections. These hubs result from the modifications proposed in the MeABC, which introduces in the position update rules of the employed and onlooker bees. The employed and onlooker bees also move towards the best food source in the colony, and the hubs in Fig. 4 represents these food sources. Note that because a single time window can group several iterations, it is possible to have more than one hub.

Furthermore, comparing the last time window (TW^{50}) of the original and memetic ABC, one can notice that the memetic algorithm seems

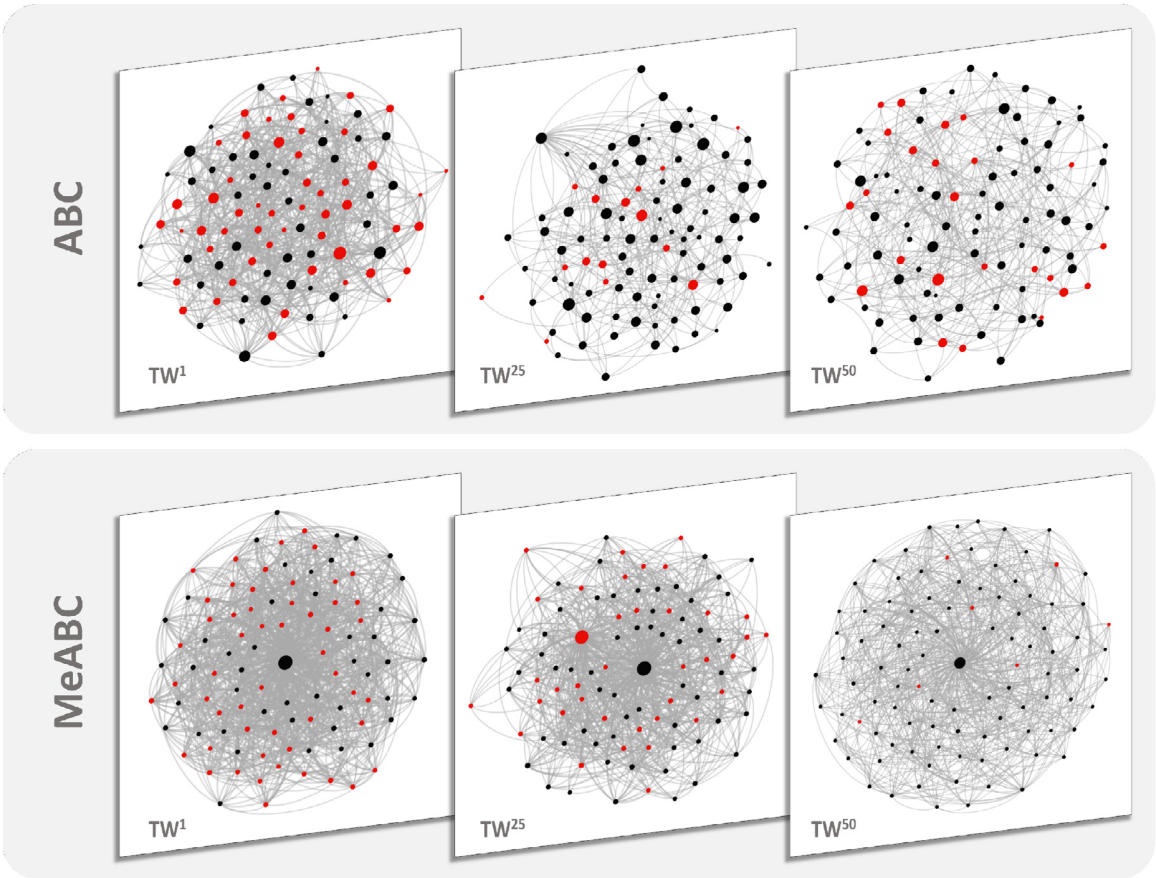

Fig. 4. Examples of integration networks generated for the ABC and MeABC at the beginning (TW^1), middle (TW^{25}) and end (TW^{50}) of the optimisation process. The colour represents the node's fitness (darker the colour, the best), size is proportional to the degree of influence of the node (i.e. the number of connections that point inward at the node).

to have a swarm more good solutions than the original ABC. This convergence could be forced by the introduction of the movement of the bees towards the best elements. Moreover, adopting a global best seems to reduce the effect of the scout bee, which is a mechanism of the ABC to maintain diversity in the swarm by generating random solutions when they cannot be improved after the $limit$ number of attempts. Since this situation usually happens at the end of the execution when the swarm converges to a spot, and it is difficult to improve the solution, the scout bee operates more in the last iterations.

We wanted to compare and contrast different algorithm networks in the next experiments to understand their stochastic nature. Moreover, we also wanted to analyse how the structure of the network changes along the optimisation process. For this reason, we used the PD metric to compare the degree of similarity between the networks in both cases (i.e. different executions and different time windows). Fig. 5 presents the results of the comparison of the networks in different time windows of the PSOs while Fig. 6 shows the results for the ABCs.

Fig. 5 shows the comparison of the networks in different time windows for the PSO and the PSO-PS. In both algorithms, the network at the beginning of the execution is structurally different from those at the end (time windows close to the 50). This difference might result from the convergence of the swarm, which changes the interaction patterns and changes the weight of the connections between individuals. Furthermore, comparing the heat map for the PSO (Fig. 5 A and D) with the PSO-PS (Fig. 5 B and E), we can see that the pattern of the PSO-PS is similar to the PSO after the first ten time windows. Because the network structure can be associated with the exploration/exploitation and convergence state of the swarm, this shift can result from the enhanced exploitation capabilities of this algorithm compared to the PSO.

The comparison between the PSO and the PSO-PS (Fig. 5 C and F) reveals that the structure of the network for these algorithms are considerably similar with a difference that for the PSO-PS, thanks to the enhanced local search capabilities, the convergence happens earlier than the PSO. This early convergence produces the shift in the pattern of Fig. 5 C and makes the network at the last time window of the PSO be more similar to the networks around the time window 45 of the PSO-PS. Again, we expect this result since the memetic PSO used in this work do not modify the interaction rules of the PSO.

In contrast to the PSO and PSO-PS comparison, the difference between the ABC and the MeABC is more prominent, indicating a faster convergence and changes in the interactions patterns of the algorithms. The results depicted in Fig. 6 show that the modifications proposed in the MeABC changed the structure of the network. In the ABC, the pattern indicates a high degree of similarity of a network in a given time window and the time windows before and after it. Nevertheless, in the MeABC, we can see that the pattern became more narrow, which indicates that the network presents a particular type of structure at each time window. Since this characteristic is also present in the PSO networks, adopting the position update rule that used a global best individual can be the cause of the pattern displayed in the MeABC results.

Comparing the ABC to the MeABC, we can see in Fig. 6 C and F that the shift in the pattern is more significant than the one depicted in Fig. 5 C, which can be due to the introduction of the local search combined with the usage of a global best bee to guide the swarm in the MeABC. Another interesting aspect illustrated by Fig. 6 C is that we can see an abrupt change in the pattern around the time window 14. Analysing the characteristics of the networks before and after TW^{14}, we noticed a reduction in the pace at which the average node degree

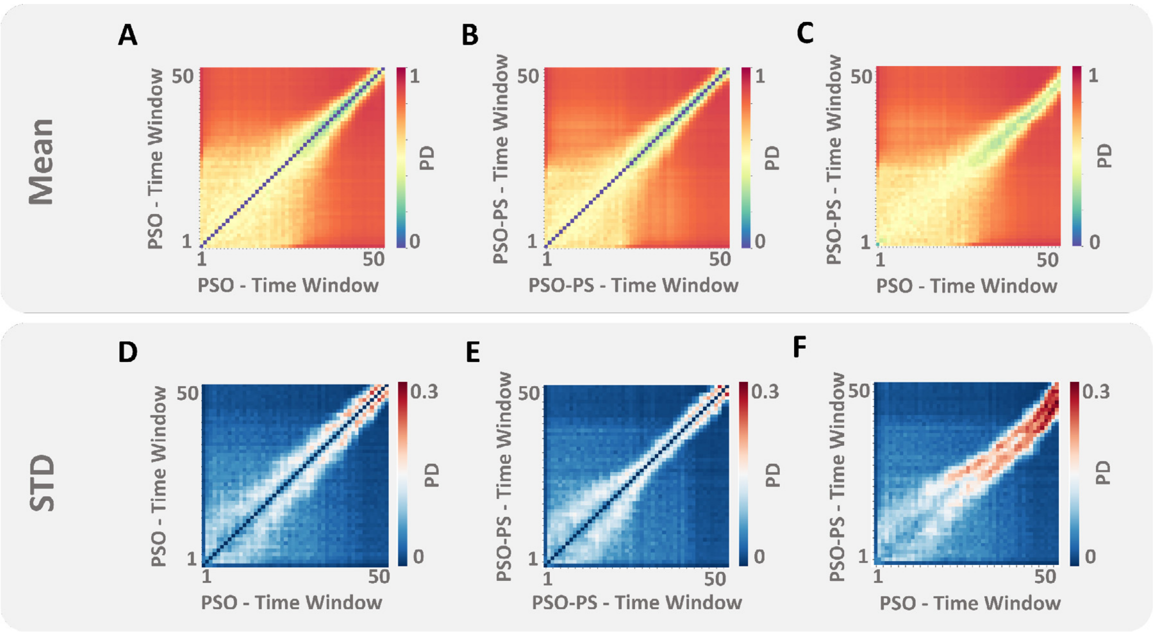

Fig. 5. Differences between the networks of the PSO and PSO-PS. We employed the mean(A - C) and standard deviation (D - F) of the PD metric to compare the last networks produced along the optimisation process. In both cases, we adopted a time window with a size of 10 iterations (we merge the results of 10 iterations into a single network). The colours indicate the degree of similarity of the networks compared; the reddish is the colour, the more different are the networks, while the bluish is the colour, the more similar they are in their structure. The values in these plots are the average of 30 independent simulations of the algorithms.

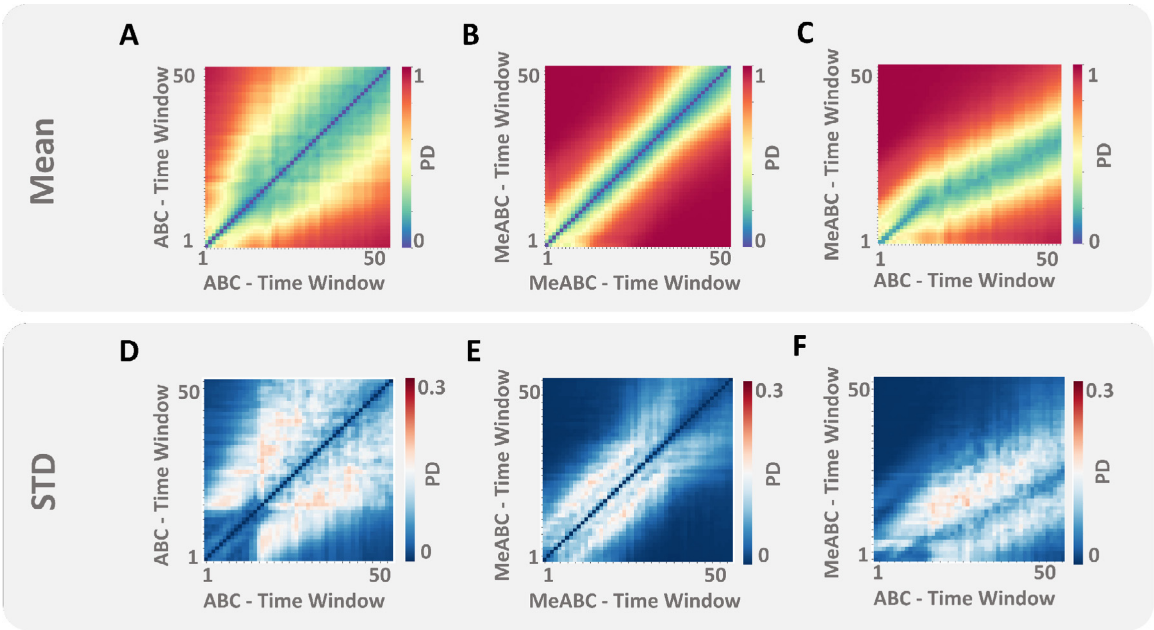

Fig. 6. Differences between the networks of the ABC and MeABC. We employed the mean(A - C) and standard deviation (D - F) of the PD metric to compare the last networks produced along the optimisation process. In both cases, we adopted a time window with a size of 10 iterations (we merge the results of 10 iterations into a single network). The colours indicate the degree of similarity of the networks compared; the reddish is the colour, the more different are the networks, while the bluish is the colour, the more similar they are in their structure. The values in these plots are the average of 30 independent simulations of the algorithms.

was decreasing in the ABC. It is worth mentioning that these results are related to the scenario simulated and might change with different conditions.

Furthermore, in the MeABC, after TW^{14}, we detected a significant drop in the number of edges in the networks. In the memetic ABC, two situations can lead to exploitation and stagnation behaviour. In the former scenario, the bees keep on exploiting the same best(s) solution, which means that instead of creating new edges with other agents in the swarm, the bees keep on updating the connection that they have with a bee/group of bees (i.e. exploiting the solution). In the latter scenario, because the onlooker and scout bees have greed operators, they will not move due to stagnation or convergence if the bees cannot improve their solution. Hence, they will not move and thus not create new connections.

To identify the swarm stagnation/convergence in ABC-based algorithms, we can observe the frequency in which the scout bees are used to generate new solutions. In Fig. 7 C, the green and purple curves indicate when the scout bee generates a new solution in the ABC and

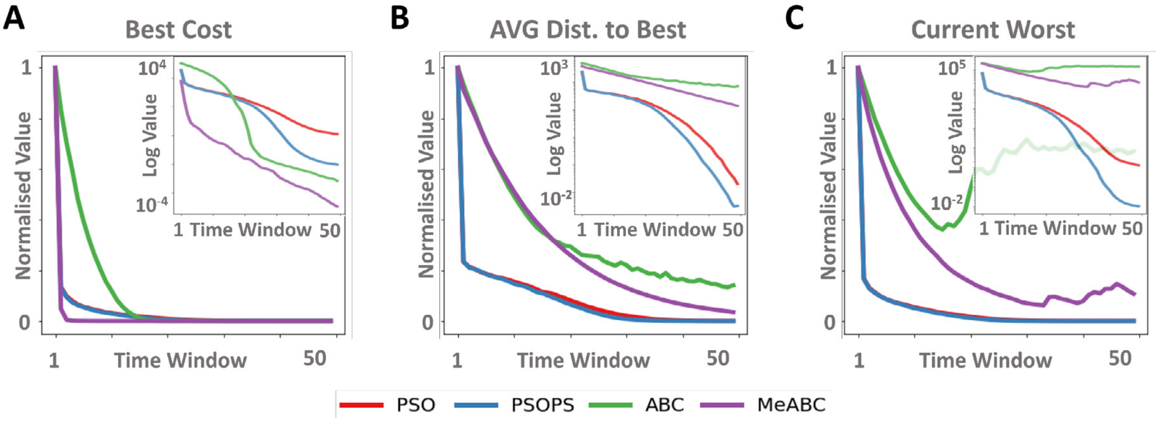

Fig. 7. Curves showing the evolution of the swarm along with the time windows in the Sphere function with 50D. The large plots have the curves normalised (i.e. we divide each curve by its highest value) to analyse the characteristics of the curves rater, then comparing the quality of the solution of the algorithms. In contrast, the inset plots have the y-axis in a logarithm scale, and this was done to show that the MeABC and PSO-PS versions were able to improve the performance of the ABC and PSO, respectively. It is worth mentioning that the current worst cost is the fitness value of the worst element in the swarm at that specific time.

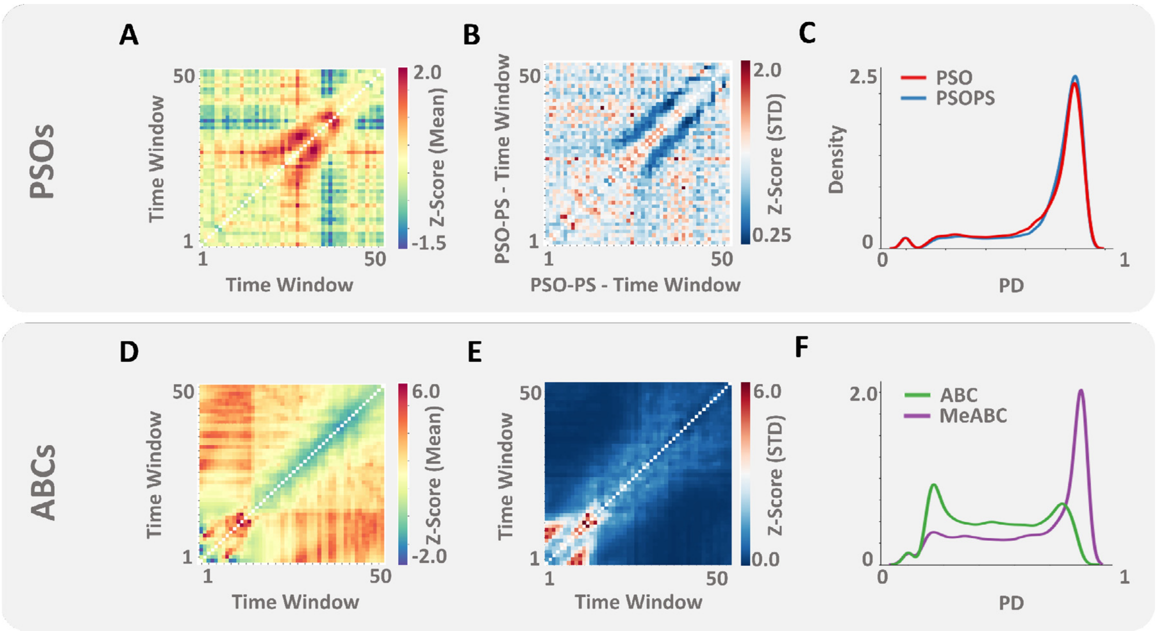

Fig. 8. Z-Score computed using the PSO as the reference for the PSO-PS and the ABC as the reference for the MeABC. Panels A and D depict the mean, and panel B and E illustrates the standard deviation. Note that, for the Z-Score, values close to zero indicate the similarities between the algorithm reference and its memetic version. Panel C and F represent the convergence signature of the PSO, PSO-PS, ABC and MeABC as a kernel density estimation (KDE), visible respectively in Fig. 5 A and B and Fig. 6 A and B. Note that the memetic PSO has a similar signature to the PSO. In contrast, the memetic ABC presents a distinct pattern to the ABC.

MeABC. We can see in Fig. 7 C that for the ABC, the scout bee was first used around the TW^{14} time window, which could indicate that at that point, the swarm converged to a point where the bees had difficulties improving the solutions. We can also see no substantial improvements in the cost of the best solution found after that point.

Moreover, Fig. 7 B and C shows that after TW^{14}, the curves for the MeABC continues to improve. This behaviour can be an indication that the best solution and the swarm continue to improve. It is also worth mentioning Fig. 7 A indicates that changes proposed in the memetic algorithms made achieve better solutions and improved the convergence pace of the swarm.

One explanation for the most notable difference between the ABCs and the PSOs has to do with the more extensive modification proposed in the memetic ABC. The number of best individuals in a time window, represented as hubs in the network, can vary from execution to execution in the PSO. In contrast, the ABC usually does not have hubs or another characteristic that can change considerably from one execution to others, making the networks more similar. The presence of hubs caused by the global best individual might also be the reason behind the difference between ABC and MeABC.

We highlight that, in this work, we focus on studying the behaviour of the algorithms, and for this reason, we present the convergence curves with values normalised. We understand that this approach prevents comparisons related to the quality of the solutions found; however, the inset graphs of Fig. 7 and previous works [5,37] give us indications of the performance of these algorithms in terms of the quality of the solutions found.

The following results present three methods to simplify the analysis of the heatmaps and the comparison between the algorithms. These methods rely on the analysis of the PD convergence patterns (panels A and B of Fig. 5 and Fig. 6) as data distributions (Fig. 8 and Table 1) and networks (Fig. 9).

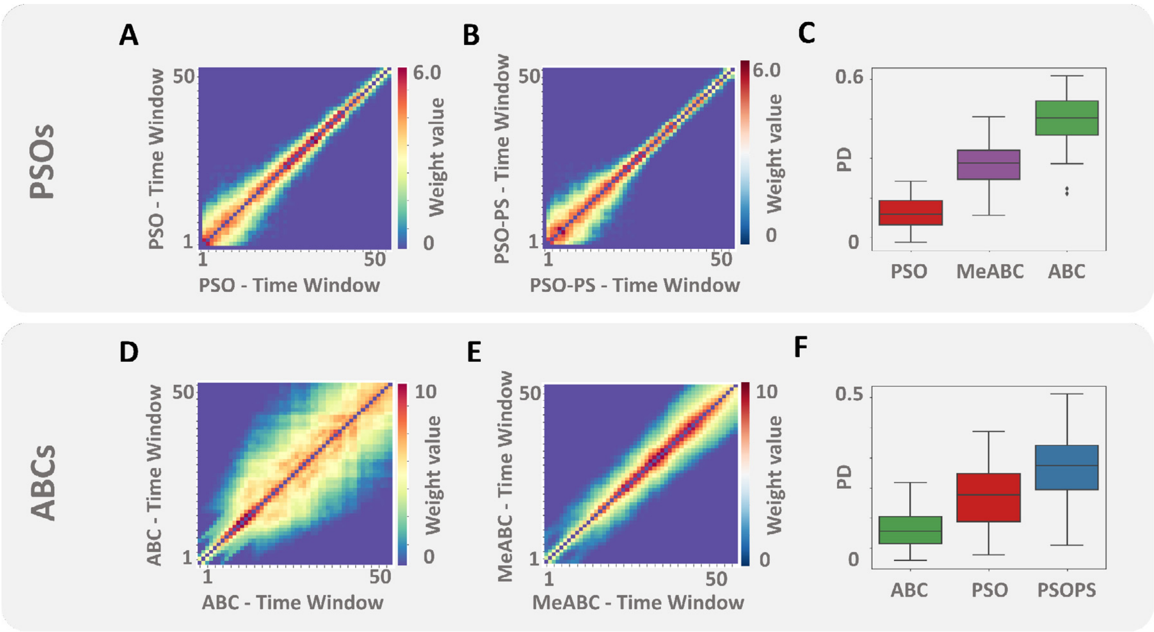

Fig. 9. Application of the PD to compare the similarities between the convergence pattern of the algorithms. To apply the PD metric, we considered the matrices represented in Fig. 5 A and B and Fig. 6 A and B as an adjacency matrix and, for each algorithm and every execution, we create a network where the nodes are the time windows. The weight of the edge is equal to the inverse of value in the respective matrix ($1/PD$). To not produce fully connected networks, we applied a threshold to remove all values greater than 0.5. Panel A, B, D and E depict the resultant adjacency matrix used to create the networks of the PSO, PSO-PS, ABC and MeABC, respectively. Panel C and F show, respectively, the PD results of comparing the PSO-PS and MeABC to the other algorithms. Note that the PSO-PS is more similar to the PSO; however, the modifications introduced in the MeABC made it more similar to the PSO/PSO-PS than the ABC.

Table 1
Results of other metrics used to compare the distributions of the algorithms (8 C and F). All the results present are statistically significant (with *p-value* < 0.05). We highlight that for the Kolmogorov-Smirnov test, values equal to zero indicates that we cannot disregard the Null hypothesis (i.e. the two distributions are identical). In contrast, for the Spearman and Kendall's Tau correlations, values close to one mean a positive correlation between the data compared. Note that the results for these metrics agree with the previous analysis of Figs. 5, 6, 8 and 9.

Comparison	Kolmogorov-Smirnov	Spearman Correlation	Kendall's Tau
PSO-PS - PSO	0.053	0.854	0.680
PSO-PS - ABC	0.442	0.353	0.247
PSO-PS - MeABC	0.382	0.387	0.275
MeABC - ABC	0.331	0.880	0.696
MeABC - PSO	0.380	0.468	0.339

In the first method, we flatten the matrices transforming them into a vector and then compare the data distribution and measure the correlation between the distribution of different algorithms.The flatten operation is employed to reduce the problem dimensionality. It is performed by generating a one-dimensional vector resultant of the sequential concatenation of the matrix rows. Hence, for a given matrix A with M rows and N columns, the resultant vector V is described by $V[idx] = A[i][j]$ for i in $[1, 2, \dots, N-1, N]$ and j in $[1, 2, \dots, M-1, M]$, where $idx = (N \times i) + j$. Fig. 8 panel A and B depict, respectively, the mean and standard deviation of the comparison between the PSO and PSO-PS using the Z-Score. The idea is to show as a heatmap the time windows that present the most significant differences (values far from zero) from the PSO and PSO-PS. As we can see in Fig. 8 panel A and B, the main difference between the PSO and PSO-PS happen within time window 22 and 42. These differences might be explained by the results illustrated in Fig. 5 A, B and C, where we see that the differences in the pattern of PSO and PSO-PS become more evident around time window 20. Moreover, analysing the kernel density estimation (KDE) of the PSOs, we can see significant similarities between them, supporting the conclusions of our previous analysis.

In contrast to the results of PSO and its memetic version, the Z-Score comparison between the ABC to the MeABC (Fig. 8 E and F) reveals significant disparities, especially along the main diagonal of the heatmap. Again, the analysis of the convergence patterns of these algorithms helps to explain these differences. Observing the patterns on Fig. 6 A, B and C, we can see that the dissimilarities between the ABC and MeABC are related to the changes in the shape of the convergence patterns and the values of PD (darker green colours on the MeABC compared to the ABC). Also, in Fig. 8 F looking at the KDE comparison, we can also see significant differences separating the ABC and the MeABC.

Using the same idea of transforming the matrices into vectors and compute the KDE, use selected three metrics to correlate the results of the PSOs and ABCs. Table 1 presents the results of the comparisons using the Kolmogorov-Smirnov test and Spearman and Kendalls Tau correlations. It is worth mentioning that all the results presented in the table have *p-value* < 0.05. Also, the Kolmogorov-Smirnov test has the opposite behaviour as the correlation metrics, and values close to zero indicate similarities between the data compared.

Observing the results on Table 1, we can conclude that all three metrics agree that the PSO and ABC are better matched with their respective memetic version. Also, the MeABC shares more similarities with the PSOs than the ABC. However, the Kolmogorov-Smirnov test was the only metric capable of showing that the PSO-PS is considerably more similar to the PSO than the MeABC is to the ABC.

A third way to compare the PD convergence signature of the algorithms is depicted in Fig. 9. This approach uses the heatmap as the adjacency matrix of a network with nodes equal to time windows and connections representing their similarity. To avoid producing a fully connected network and self-loops, we remove all edges with a PD value greater than 0.5 and equal to 0 (main diagonal). Also, because we wanted that stronger connection to represent high PD similarity, we defined the connection's weight as the inverse of the PD value

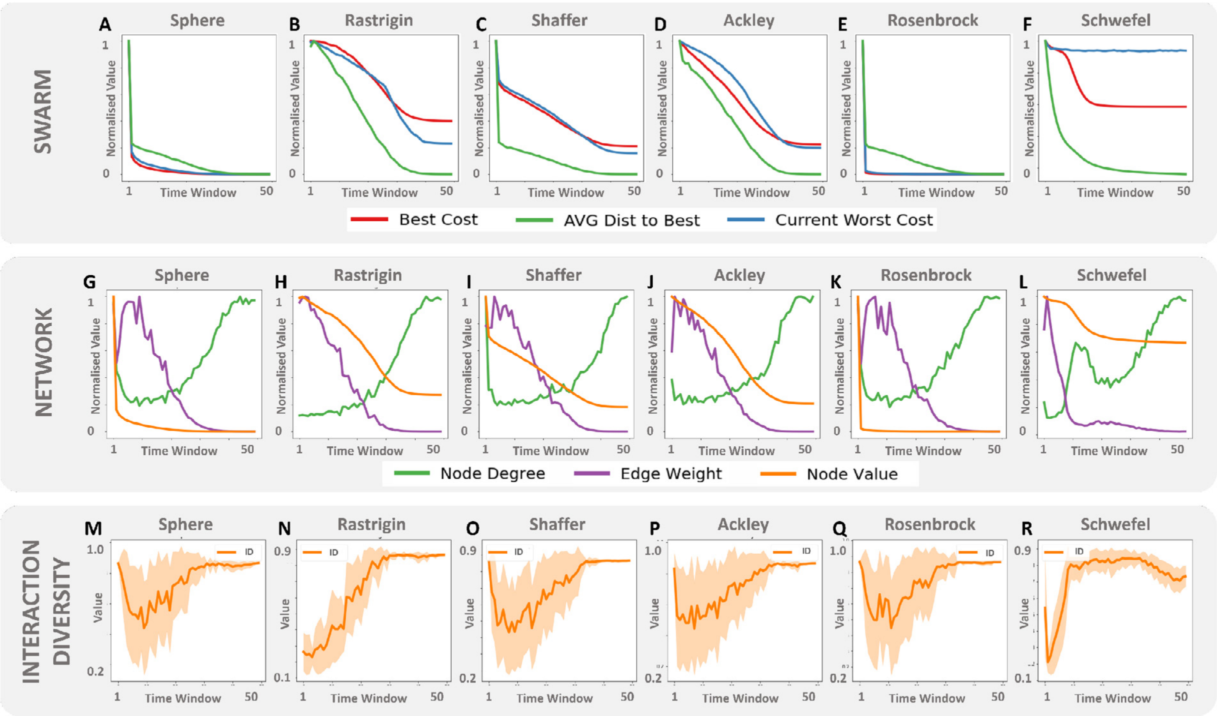

Fig. 10. Characteristics of the swarm, the interaction network, and the interaction diversity metric of the PSO on the benchmark functions. Notice that, for better visualisation, we normalised each curve on the swarm and network levels by their respective highest value.

$(1/PD)$. Fig. 9 A, B, D and E illustrates examples of adjacency matrices used.

Next, since we are dealing with networks, we can use the PD metric to assess their structural similarities. Looking at Fig. 9 C and F, we can again see that the results support our previous analysis: the PSO and ABC are more similar to their respective memetic version. Moreover, the MeABC is more similar to the PSOs than the ABC, and the degree of similarity between the PSO and PSO-PS (0.261 ± 0.040) is greater than the ABC and MeABC (0.283 ± 0.033).

3.2. Assessment of exploration-exploitation

To study the behaviour of the algorithm, we selected a set of benchmark optimisation functions with different characteristics (e.g. unimodal vs multimodal, convex vs non-convex, separable vs non-separable.). The selected function were: Ackley, Rastrigin, Rosenbrock, Shaffer's F7, Schwefel, and Sphere. Previous studies have shown that the ID metric can assess exploration/exploitation behaviours in particle swarm optimisers. The expected result for PSO-based algorithms is to start with low high values of ID (associated with exploration). It transitions to an exploitation phase (reduction in the ID values), and, after the algorithm converges, the ID starts to rise again. This last increase is caused by the spatial proximity of the particles that increase the probability of alternating the global best particle from one iteration to another.

For each benchmark function, we adopted the number of dimensions equal to 50 and executed 30 independent simulations of the algorithms using the stop criteria of 500 iterations. All the algorithms used a population of 30 individuals, and the number of problem dimensions was set as 100. The parameters used are described in the algorithms' description sections (Section 2.3.1 and Section 2.3.2).

Fig. 10 depicts the behaviour in the swarm level (A to F) and the interaction network level (G to L). We wanted to show these two types of plots to show the networks can capture the behaviour of the swarm; for example, the average node value is equivalent to the average cost of the swarm. Also, they can help us to have a better understanding of the ID metric.

The interaction networks in Fig. 10 also can provide insights concerning the exploration-exploitation behaviour in the swarm. In the exploitation phase, the swarm attempts to improve the best solution(s) found. From the information flow perspective, the population will be using the bests element(s) information to perform their search. In the interaction networks, we notice this behaviour on the values of the nodes' degree. Because interactions happen around the best nodes in the exploitation, the average degree of the network decrease in this phase. In contrast, an increase in the average node degree is an indication of exploration behaviour.

As we can see in Fig. 10 (G to L), in the majority of the functions, the PSO starts with an exploration phase, performs exploitation and then return to an exploration behaviour. The changes in the swarm search mode of the PSO are related to the frequency of changes in the particles that have the best information. These changes can happen in the initial stage of optimisation when the solutions of the particles can be improved easy leading to a high probability of changing the global best. Moreover, a similar situation happens when the swarm starts to converge. In this case, the alternations in the best particle occur because of the proximity of the solution in the search space.

One can also observe in Fig. 10 (G to L) that highly multimodal functions (e.g. Rastrigin and Schwefel) with several local minima tend to have a shorter initial exploration phase. This more brief initial exploration can be because highly multimodal can be much more challenging than the others. Hence, improving the set of initial random solution is becomes more difficult. As the optimisation progress and the swarm start to converge, the influence of the random initialisation decreases and the particles begin to alternate the role of the leader more frequently.

Previous works [24,26,29] have shown that the interaction diversity metric can be applied to analyse the balance between exploration

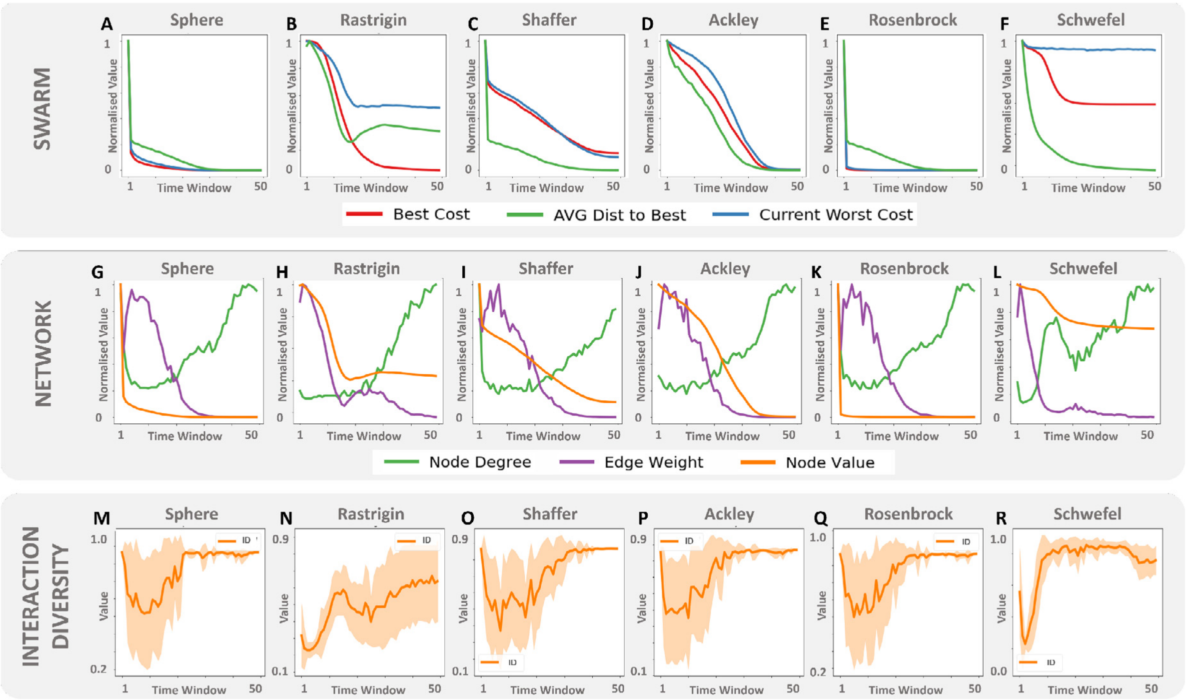

Fig. 11. Characteristics of the swarm, the interaction network, and the interaction diversity metric of the PSO-PS on the benchmark functions. Notice that, in the network plots, the values for curves represent the average node degree, average edge weight and average node value. Also, for better visualisation, we normalised each curve in the swarm and network plots by their respective highest value.

and exploitation in swarm-based algorithms; hence, we employed this metric and the modified version proposed to evaluate the networks of the algorithms studied. Fig. 10 (M - R) shows how interaction diversity metrics behave along the optimisation process.

The results of the ID metric for the PSO (Fig. 10 M - R) supports the analysis based on the value of the average node degree. Besides in the Rastrigin and Schwefel functions, the PSO presented high interaction diversity (related to exploration behaviour [24,26]) in the firsts time windows. Then, it progressively switches to exploitation (low iteration diversity) and returns to the exploration towards the end.

Concerning the PSO-PS, because the Pattern Search does not change the interaction pattern of the PSO but helps to increase local search capabilities and speed up the convergence, as expected, we do not see significant changes in the curves (Fig. 11). The most noticeable difference between the PSO and the PSO-PS happened in the Rastrigin function, where we can see that around the TW^{15} time window, the interaction diversity reach a peak and then start to decrease. After the TW^{22} time window, it increases again. Because this behaviour was not observed in the PSO, the adoption of the pattern search could have increased the probability of leaving a local minimum. When a new better solution is found, the swarm exploits it (decrease the ID). When the particles begin to converge, new better solutions are found by different particles, which increase the interaction diversity.

Furthermore, we can also observe in Fig. 11 B that, in the swarm level, around TW^{15}, the average distance to the best solution starts to increase. This increase supports the argument that a new best solution was found (probably out of the local minimum). Moreover, around TW^{22}, we can see that the average distance to the best starts to decrease again, indicating the convergence of the swarm.

Differently from the PSO, the analysis of the average node degree on the interaction network of the ABC does not display a clear relation with the exploration/exploration phases of the algorithm. As we can see in Fig. 12, the curves of the ABC in the swarm, network and interaction diversity level are different from the PSO and PSO-PS. Note that the greedy operators of the ABC are one of the reasons for the decrease in the average node degree of the network. As the solutions are refined, it becomes more challenging to improve their quality because the edges are created only when there are improvements. Hence, the network experience a decrease in the average node degree.

As explained before, only the onlooker bee uses the quality of the solution as a criterion for interaction selection. In contrast, the employed bee has a random selection, and the scout bee does not produce interactions between the solutions in the swarm. Due to this characteristic, the ABC network tends not to have hubs as the PSO has. However, using the interaction diversity, we can still have a sense of the exploration and exploitation phases of the ABC.

Fig. 12 (M to R) depicts the results obtained for all the benchmark functions. As can be observed, the interaction diversity for the ABC presented values superior to 0.5 in all the scenarios simulated. This result might indicate a high exploration component of the algorithm. Moreover, while the curve shows a downward trend (reducing exploration) for the Rastrigin and Schwefel function, for the Sphere, Ackley and Rosenbrock, we can see an upward trend (increasing exploration).

Comparing the results of the Fig. 12 to the MeABC Fig. 13 we can see several differences in all the levels. In the swarm level, we can see that the convergence was faster, and the effects of the scout bee were reduced (observe the behaviour of the red line). At the network level, we can see that for the Sphere, Rastrigin, Shaffer and Schwefel, the average node degree falls to a certain level and continues fluctuating around this level instead of decreasing.

Besides the differences in the swarm and network level, the interaction diversity level of the MeABC is also different from the ABC. As we can see in Figure 13 M to R, although the ID value is superior to 0.5 as in the ABC, for the MeABC, it does not change as much as in the ABC. These differences could result from the introduction of the global best information in the position update rules of the ABC and the usage of the memetic phase on the best solution in the swarm.

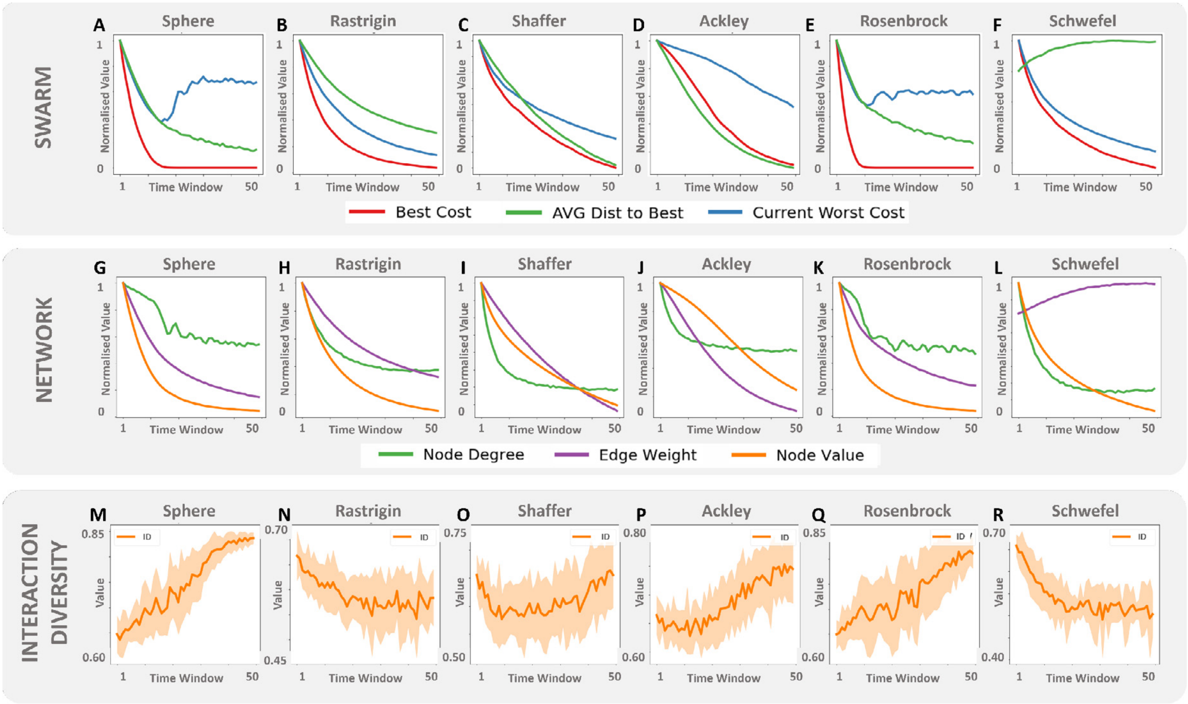

Fig. 12. Characteristics of the swarm, the interaction network, and the interaction diversity metric of the ABC on the benchmark functions. Notice that, for better visualisation, we normalised each curve on the swarm and network levels by their respective highest value.

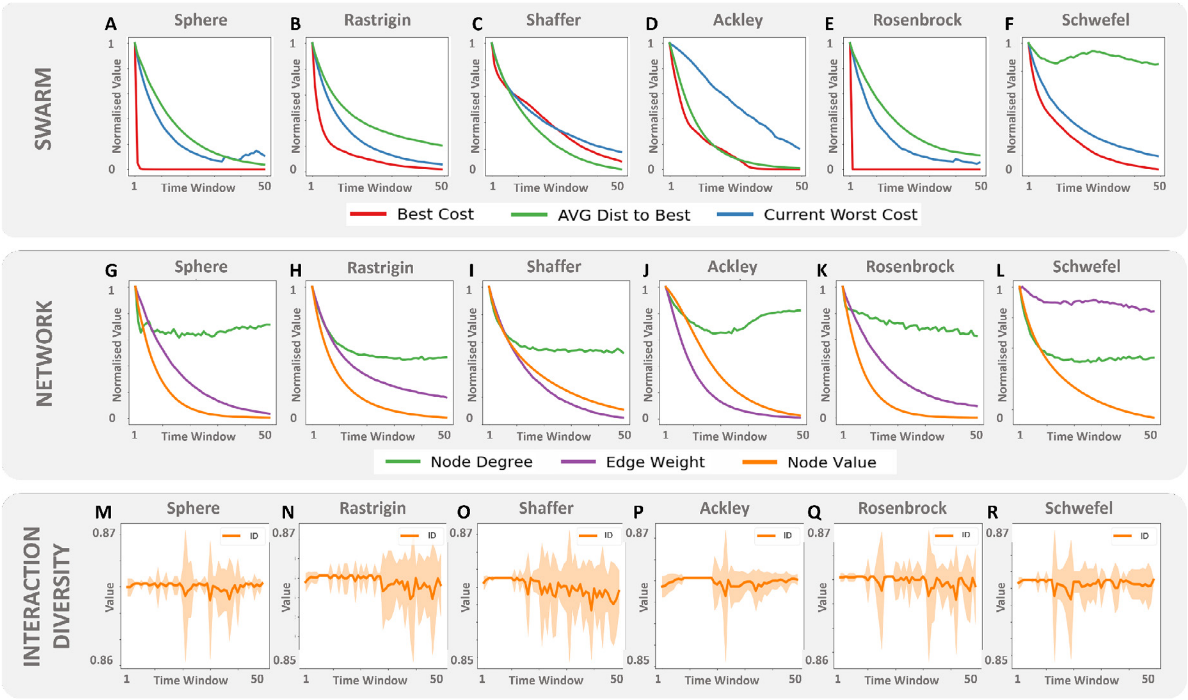

Fig. 13. Characteristics of the swarm, the interaction network, and the interaction diversity metric of the MeABC on the benchmark functions. Notice that, for better visualisation, we normalised each curve on the swarm and network levels by their respective highest value.

3.3. Analysis of convergence

To analyse the convergence behaviour of the algorithms, we selected the Sphere and Rastrigin functions to verify the differences between the convergence behaviour in both unimodal and multimodal problems. Also, to ensure that the algorithms would converge, we kept the exact parameters of the previous experiments but lowered the number of dimensions of the problem from 50 to 5.

Because of the characteristics of the algorithms selected, we expect to see two different behaviours in a unimodal function: PSO-based algorithms should converge to a solution and continue there until the stop criteria are met. In contrast, ABC-based algorithms should present multiple cycles of convergence divergence due to the action of the scout bee.

Fig. 14 shows the comparison of the networks of the algorithms in different stages of the optimisation process in the Sphere function. The

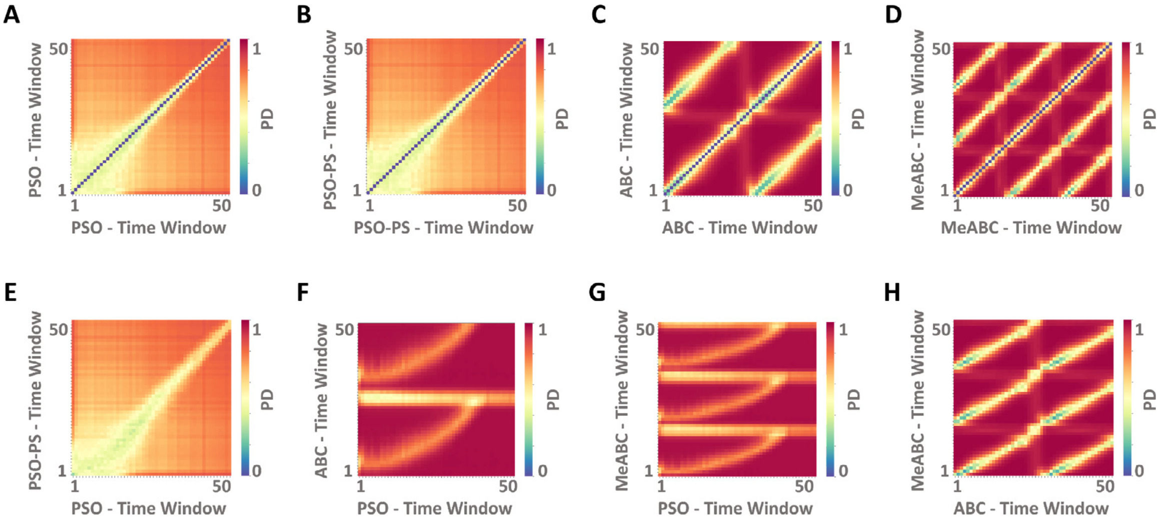

Fig. 14. Comparison between the interaction networks of the algorithms at different time windows in the Sphere function with five dimensions. The value at each point in the heat map is the average value for 30 independent simulations. It is worth mentioning that the values close to 0 (blue colour) indicates a high degree of similarities between the structure of the networks compared.

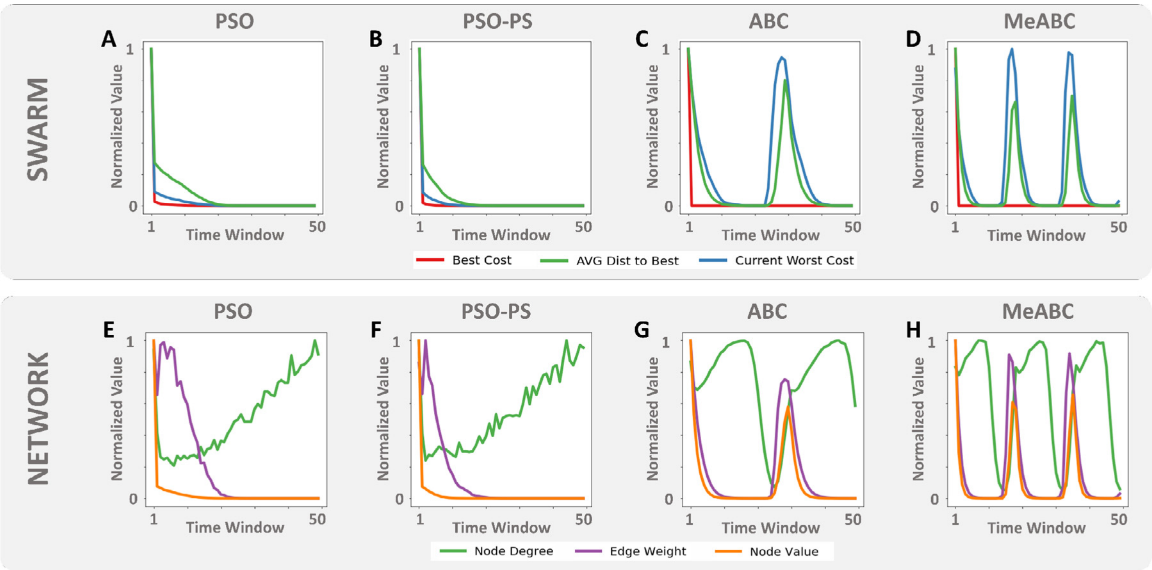

Fig. 15. Convergence in the Sphere function observed from the swarm and network level. Note the similarity between the plots of both levels for all the algorithms. Also, due to these similarities, the network level can be used to monitor the convergence of the swarm.

results for the PSO-based algorithms are quite similar (Fig. 14 A and B), the difference being that the memetic PSO converges faster, thus producing the distorted pattern on Fig. 14 E. These results supports our hypothesis about the convergence behaviour of the PSO and PSO-PSO. Note that the faster convergence of the PSO-PS compared to the PSO becomes more noticeable as the problem complexity increases, as in the previous experiments with 50 dimensions.

For the ABC and the MeABC, Fig. 14 C and D presents a more precise indication of convergence. The number of repetitions in the pattern in the main diagonal corresponds to the number of times that the swarm have converged. Again, the recurrences occur due to the action of the Scout bee, which generates new solutions every time that a food source cannot be improved after a consecutive number of trials. As claimed in previous works, such a mechanism to reset part of the population in case of convergence is beneficial to MAs. It also reduces the probability of the population converge to a sub-optimal solution [19].

Furthermore, Fig. 14 H illustrates that in one execution, the MeABC has three cycles of convergence while the MeABC has two. Comparing the ABC-based algorithms to the PSO-based ones, in Fig. 14 F and G, it is noticeable that the convergence cycles of the ABC and MeABC are also apparent. The horizontal line yellow line in the plots represents when the ABC/MeABC starts to transition from the convergence.

To test these claims, Fig. 15 depicts the same scenario (i.e. Sphere function with five dimensions) as Fig. 14 but under the perspective of the swarm and network levels. Moreover, the results in Fig. 15 A, B, E and F indicate that the analysis made based on the PD heat maps matches with the plots for the PSO and PSO-PS. Similarly, when we analyse the plots for the ABC and MeABC (Fig. 15 C, D, G and H), we can see that the convergence cycles are reflected as the peaks in the curves.

Regarding the results on a multimodal function, Fig. 16 shows the PD heat maps for the networks generated on the five dimensions Rastrigin function. We can notice in Fig. 16 A and B that around TW^{40} and TW^{35} for the PSO and PSO-PS, respectively, the pattern begins to change. This

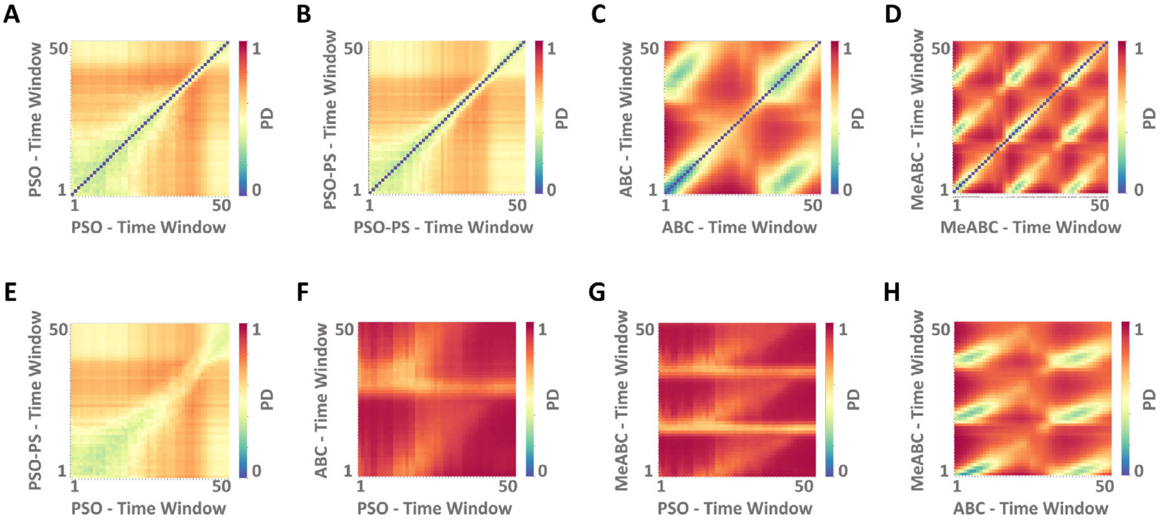

Fig. 16. Comparison between the interaction networks of the algorithms at different time windows in the Rastrigin function with five dimensions. The value at each point in the heat map is the average value for 30 independent simulations. It is worth mentioning that the values close to 0 (blue colour) indicates a high degree of similarities between the structure of the networks compared.

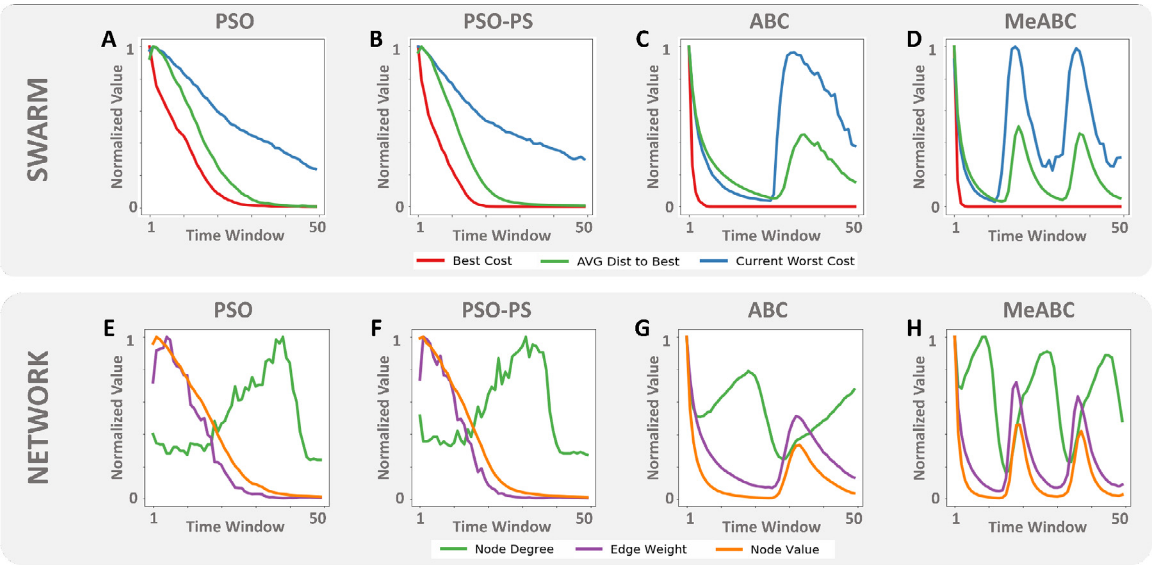

Fig. 17. Convergence in the Rastrigin function observed from the swarm and network level. Note the similarity between the plots of both levels for all the algorithms. Also, due to these similarities, the network level can be used to monitor the convergence of the swarm.

change could be due to the discovery of a better solution in the swarm. The comparison between the PSO and the PSO-PS shows a similar result; however, the faster convergence of the PSO-PS might have caused a shift in the pattern on Fig. 16 E.

Observing the results on the Rastrigin function for the ABC and MeABC on Fig. 16 C, D, G and H, we can see a similar situation to the unimodal function where the ABC presented two convergence cycles while the MeABC had three. Nonetheless, one can notice that the convergence pattern in the multimodal function cycles is different from the one for the unimodal function.

The analysis of the PSO and PSO-PS under the network level helps us to understand the patterns depicted in Fig. 16 A and B. When the PD heat maps pattern begins to change, we see a decrease in the average node degree in the network. Because there are no substantial changes in the other curves, this is an indication that the swarm found a better solution in the neighbourhood and started to exploit it. For the ABC and MeABC, Fig. 17 C, D, H and H confirms the faster convergence of the MeABC over the ABC, and the decreasing height of the peaks might be an indication that the convergence time could be decreasing from one cycle to another.

4. Conclusions

The local search capabilities of memetic algorithms are one of the reasons behind their performance in various optimisation problems. Even though we can find in the literature examples of works that study the impact of local search in genetic and ant colony optimisers, there is still room for contributions that helps to understand the extent of the effect that local search mechanisms has on MAs in general.

This paper employed a modified version of the interaction networks to capture the interaction patterns in population-based optimisation algorithms. We showed that these networks encode characteristics of the algorithms. In conjunction with the portrait divergence and interaction diversity metrics, they can be used to study convergence, exploration and exploitation capabilities.

The comparison between the PSO and the PSO-PS indicated that the inclusion of local search as an extra step of the optimisation process increases the convergence speed but does not change the PSO characteristics. However, this additional local search step can compromise the balance exploration-exploitation balance and lead to premature convergence when not performed correctly.

For the memetic ABC, because the modifications were more substantial, we noticed a significant change in the MeABC compared to the ABC. The memetic variant not only was able to converge faster but also find better solutions. Furthermore, as suggested in previous works, the presence of mechanisms that re-start part population in case of stagnation/convergence—the scout bee of the ABC/MeABC—can reduce the risk of premature convergence to sub-optimal regions.

In summary, the simple inclusion of an independent local search procedure in swarm-based algorithms might accelerate the convergence of the population. However, this does not change the characteristics of the algorithm and can lead to premature convergence. Also, better results were observed when the modifications proposed in the memetic version go beyond the inclusion of the local search. In this case, the inclusion of the local search procedure is followed by other modification and considers the features and operators of the algorithms. These changes are reflected in the behaviour of the algorithms and, consequently, in their interaction networks.

Although we adopted the PSO and ABC as examples in our analysis in this paper, interactions networks can model different types of swarm-based algorithms besides the PSO and ABC [30,31,40]. In fact, it may be extended to other types of population-based algorithms, such as evolutionary algorithms (EAs). We argue that for EAs, the network representation of these algorithms would be similar to the *phylogenetic networks* (i.e. tree-like structures that represents evolutionary relationships between individuals) [41]. The main difference between the swarm-based and the evolutionary-based networks is that the nodes in the former networks remain the same throughout the optimisation process. In contrast, the latter nodes change due to the generation of new individuals using evolutionary operators. In general terms, the networks for the EA could be seen as a dynamic network whose topology changes over time. Given that the metrics used do not depend on the networks' labels, they could also be applied to study the interaction networks of EA. Besides the investigations on evolutionary algorithms, we intend to extend this study by assessing the impact of different local search methods on the same algorithm.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Clodomir Santana: Conceptualization, Methodology, Software, Validation, Investigation, Visualization, Writing – original draft. **Marcos Oliveira:** Conceptualization, Methodology, Writing – review & editing. **Carmelo Bastos-Filho:** Conceptualization, Methodology, Writing – review & editing. **Ronaldo Menezes:** Supervision, Conceptualization, Methodology, Writing – review & editing.

Acknowledgements

This work is partially supported by a scholarship provided by the College of Engineering, Mathematics and Physical Sciences (CEMPS) at the University of Exeter. The authors would like to thank James Bagrow for providing us with his implementation for the portrait divergence.

References

[1] P. Moscato, et al., On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms, Caltech concurrent computation program, C3P Report 826 (1989) 1989.

[2] P. Moscato, M.G. Norman, A memetic approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems, Parallel computing and transputer applications 1 (1992) 177–186.

[3] R. Dawkins, The selfish gene, Oxford university press, 2016.

[4] N. Krasnogor, J. Smith, A tutorial for competent memetic algorithms: model, taxonomy, and design issues, IEEE Trans. Evol. Comput. 9 (5) (2005) 474–488.

[5] Y. Bao, Z. Hu, T. Xiong, A pso and pattern search based memetic algorithm for svms parameters optimization, Neurocomputing 117 (2013) 98–106.

[6] H. Wang, D. Wang, S. Yang, A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems, Soft comput 13 (8–9) (2009) 763–780.

[7] C.-H. Mu, J. Xie, Y. Liu, F. Chen, Y. Liu, L.-C. Jiao, Memetic algorithm with simulated annealing strategy and tightness greedy optimization for community detection in networks, Appl Soft Comput 34 (2015) 485–501.

[8] A. El Fallahi, C. Prins, R.W. Calvo, A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem, Computers & Operations Research 35 (5) (2008) 1725–1741.

[9] F. Kang, J. Li, H. Li, Artificial bee colony algorithm and pattern search hybridized for global optimization, Appl Soft Comput 13 (4) (2013) 1781–1791.

[10] P.T.H. Nguyen, D. Sudholt, Memetic algorithms outperform evolutionary algorithms in multimodal optimisation, Artif Intell (2020) 103345.

[11] F. Neri, C. Cotta, Memetic algorithms and memetic computing optimization: a literature review, Swarm Evol Comput 2 (2012) 1–14.

[12] S. Areibi, Z. Yang, Effective memetic algorithms for vlsi design= genetic algorithms+ local search+ multi-level clustering, Evol Comput 12 (3) (2004) 327–353.

[13] G. Fraser, A. Arcuri, P. McMinn, A memetic algorithm for whole test suite generation, Journal of Systems and Software 103 (2015) 311–327.

[14] Y. Mei, K. Tang, X. Yao, Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem, IEEE Trans. Evol. Comput. 15 (2) (2011) 151–165.

[15] M. Gong, B. Fu, L. Jiao, H. Du, Memetic algorithm for community detection in networks, Physical Review E 84 (5) (2011) 056101.

[16] R. Mencia, M.R. Sierra, C. Mencia, R. Varela, Memetic algorithms for the job shop scheduling problem with operators, Appl Soft Comput 34 (2015) 94–105.

[17] Y.-S. Ong, M.-H. Lim, N. Zhu, K.-W. Wong, Classification of adaptive memetic algorithms: a comparative study, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 36 (1) (2006) 141–152.

[18] P. Moscato, C. Cotta, A Gentle Introduction to Memetic Algorithms, in: Handbook of metaheuristics, Springer, 2003, pp. 105–144.

[19] P. Moscato, C. Cotta, An Accelerated Introduction to Memetic Algorithms, in: Handbook of Metaheuristics, Springer, 2019, pp. 275–309.

[20] F. Neri, C. Cotta, A Primer on Memetic Algorithms, in: Handbook of Memetic Algorithms, Springer, 2012, pp. 43–52.

[21] U. Aickelin, E.K. Burke, J. Li, An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering, Journal of the Operational Research Society 58 (12) (2007) 1574–1585.

[22] J. Levine, F. Ducatelle, Ant colony optimization and local search for bin packing and cutting stock problems, Journal of the Operational Research society 55 (7) (2004) 705–716.

[23] M.A. Oliveira-Júnior, C.J. Bastos-Filho, R. Menezes, Assessing Particle Swarm Optimizers Using Network Science Metrics, in: Complex Networks IV, Springer, 2013, pp. 173–184.

[24] M. Oliveira, D. Pinheiro, M. Macedo, C. Bastos-Filho, R. Menezes, Better exploration–exploitation pace, better swarm: Examining the social interactions, in: 2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI), IEEE, 2017, pp. 1–6.

[25] J.P. Bagrow, E.M. Bollt, An information-theoretic, all-scales approach to comparing networks, Applied Network Science 4 (1) (2019) 45.

[26] M. Oliveira, D. Pinheiro, B. Andrade, C. Bastos-Filho, R. Menezes, Communication diversity in particle swarm optimizers, in: International Conference on Swarm Intelligence, Springer, 2016, pp. 77–88.

[27] F. Rezaei, H.R. Safavi, Guaspso: a new approach to hold a better exploration–exploitation balance in pso algorithm, Soft comput 24 (7) (2020) 4855–4875.

[28] J.C. Bansal, H. Sharma, K. Arya, K. Deep, M. Pant, Self-adaptive artificial bee colony, Optimization 63 (10) (2014) 1513–1532.

[29] M. Oliveira, D. Pinheiro, M. Macedo, C. Bastos-Filho, R. Menezes, Uncovering the social interaction network in swarm intelligence algorithms, Accepted for publication on Applied Network Science (2020).

[30] C. Santana, E. Keedwell, R. Menezes, An approach to assess swarm intelligence algorithms based on complex networks, in: Proceedings of the 2020 Genetic and Evolutionary Computation Conference, 2020, pp. 31–39.

[31] N. Gurrapadi, L. Taw, M. Macedo, M. Oliveira, D. Pinheiro, C. Bastos-Filho, R. Menezes, Modelling the social interactions in ant colony optimization, in: International Conference on Intelligent Data Engineering and Automated Learning, Springer, 2019, pp. 216–224.

[32] M. Mavrovouniotis, C. Li, S. Yang, A survey of swarm intelligence for dynamic optimization: algorithms and applications, Swarm Evol Comput 33 (2017) 1–17.

[33] M. Oliveira, C.J. Bastos-Filho, R. Menezes, Towards a network-based approach to analyze particle swarm optimizers, in: 2014 IEEE Symposium on Swarm Intelligence, IEEE, 2014, pp. 1–8.

[34] L. Taw, N. Gurrapadi, M. Macedo, M. Oliveira, D. Pinheiro, C. Bastos-Filho, R. Menezes, Characterizing the social interactions in the artificial bee colony algorithm, arXiv preprint arXiv:1904.04203(2019).

[35] M. Tantardini, F. Ieva, L. Tajoli, C. Piccardi, Comparing methods for comparing networks, Sci Rep 9 (1) (2019) 1–19.

[36] V. Torczon, On the convergence of pattern search algorithms, SIAM J. Optim. 7 (1) (1997) 1–25.

[37] J.C. Bansal, H. Sharma, K. Arya, A. Nagar, Memetic search in artificial bee colony algorithm, Soft comput 17 (10) (2013) 1911–1928.

[38] C. Tsai, J. Kolibal, M. Li, The golden section search algorithm for finding a good shape parameter for meshless collocation methods, Eng Anal Bound Elem 34 (8) (2010) 738–746.

[39] F.-M. De Rainville, F.-A. Fortin, M.-A. Gardner, M. Parizeau, C. Gagné, Deap: A python framework for evolutionary algorithms, in: Proceedings of the 14th annual conference companion on Genetic and evolutionary computation, 2012, pp. 85–92.

[40] M. Macedo, L. Taw, N. Gurrapadi, R.C. Lira, D. Pinheiro, M. Oliveira, C. Bastos–Filho, R. Menezes, Fishing for interactions: a network science approach to modeling fish school search, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2021, pp. 40–48.

[41] D.H. Huson, R. Rupp, C. Scornavacca, Phylogenetic networks: Concepts, algorithms and applications, Cambridge University Press, 2010.