

# An Approach to Assess Swarm Intelligence Algorithms Based on Complex Networks

Clodomir Santana  
University of Exeter  
Exeter, Devon  
cj413@exeter.ac.uk

Edward Keedwell  
University of Exeter  
Exeter, Devon  
e.c.keedwell@exeter.ac.uk

Ronaldo Menezes  
University of Exeter  
Exeter, Devon  
r.menezes@exeter.ac.uk

## ABSTRACT

The growing number of novel swarm-based meta-heuristics has been raising debates regarding their novelty. These algorithms often claim to be inspired by different concepts from nature but the proponents of these seldom demonstrate whether the novelty goes beyond the natural inspiration. In this work, we employed the concept of *Interaction Networks* to capture the interaction patterns that take place in the algorithms during the optimisation process. The analyses of these networks reveal aspects of the algorithm such as the tendency to achieve premature convergence, population diversity, and stability. Furthermore, we propose the usage of *Portrait Divergence*, a state-of-the-art metric to assess the structural similarities between networks. Using this approach to analyse the Cat Swarm Optimisation algorithm, we were able to identify some of the algorithm's characteristics, assess the impact of one its parameters, and compare it to two other well-known methods (Particle Swarm Optimisation and Artificial Bee Colony). Lastly, we discuss the relationship between the interaction network and the performance of the algorithm and demonstrate the similarities between Cat Swarms and Particle Swarms.

## CCS CONCEPTS

• Theory of computation → Bio-inspired optimization; • Mathematics of computing → Evolutionary algorithms; Network flows;

## KEYWORDS

Swarm Intelligence, Complex Networks, Interaction Networks, Cat Swarm Optimisation

### ACM Reference Format:

Clodomir Santana, Edward Keedwell, and Ronaldo Menezes. 2020. An Approach to Assess Swarm Intelligence Algorithms Based on Complex Networks. In *Proceedings of the Genetic and Evolutionary Computation Conference 2020 (GECCO '20)*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/nmmnnnn.nnnnnnn>

## 1 INTRODUCTION

Several published studies indicate that the number of proposals of “novel” optimisation methods has been increasing in recent

years [12, 22]. This is mostly related to the proposal of methods claiming to have a different source of inspiration but that in reality have only this inspiration as novelty given that they can behave like (or be a particular case of) another existing method [28]. In fact, many authors have pointed out that several new methods can use inaccurate or unconvincing metaphors to justify the proposal of algorithms which often can be considered a simplification/variation of another method [5, 8, 20, 21].

As a sub-class of nature-inspired algorithms, Swarm Intelligence (SI) methods also are susceptible to this issue. Part of this problem comes from the fact that, currently, there is no comprehensive method to explain, assess, classify or compare those algorithms. In fact, the most common type of comparison between SI algorithms relies on the fitness results. Moreover, despite the rich literature in the field, the complex behaviour that emerges from the interactions in swarm-based algorithms is still not well understood.

When it comes to SI algorithms, regardless of their source of inspiration they all share the premise that the elements which constitute the population present some degree of information exchange among them. In fact, the intelligence of the swarm emerges from these interactions. For example, in the Particle Swarm Optimisation (PSO) [4], the particles exchanges positional information of the global/local best particle in the swarm. In the Artificial Bee Colony (ABC) [10] the onlooker bees select a food source to explore based on the information obtained from the employed bees.

The existence of such common feature can be used to develop tools to assess, classify and compare SI algorithms from the interaction level. One possible way of analysing the information flow and interaction patterns in a swarm algorithm is to model the interaction between the elements as a network and use it to obtain information about the algorithm.

This idea of creating a network to capture the flow of information or interactions patterns in a swarm-based algorithm was introduced in 2013 by Oliveira et al. [17]. This initial concept, known as the Influence Graph (later renamed to Interaction Network), was applied the assess particle swarm optimisers from a network science perspective. At each iteration, a graph was created by connecting the particles which shared information (e.g. a connection would be created between all particles in the swarm and the best particle in a PSO with global topology). The authors examined the characteristics of the influence network for the PSO with global, ring and dynamic topologies and analysed the Laplacian Matrix, R-value and Density Spectrum of the networks. The experiments conducted indicates that the influence graph could be used to analyse the search behaviour of the particles and diagnose stagnation.

In 2014 the authors introduced the concept of Influence Graph with History [14]. The main motivation for the introduction of

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GECCO '20, July 8–12, 2020, Cancún, Mexico

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nmmnnnn.nnnnnnn>

the concept of history was to have a method to record all the information exchange that occurred in a given time window and assess the whole history of interactions. This was achieved by summing the adjacency matrix of the networks generated at each iteration within the time window. This approach was also tested in the PSO with global, ring and dynamic topologies and they analysed the distribution of node degrees and the impact of edge removal on the network structure.

A new metric which could be applied over the influence graph to obtain information on the behaviour of the algorithm was introduced in 2016 [15]. This metric, called Communication Diversity, measures the existence of different information flows within the swarm during the optimisation process. The experiments performed with the PSO indicated that communication diversity can be used to predict the stagnation of the algorithm. Furthermore, a further study from 2017 showed how this metric could be applied to examine the exploration/exploitation capabilities of the PSO [16].

Besides the applications of the influence graph to analyse PSO, in 2019 this method, which was renamed to Interaction Network and applied on the Artificial Bee Colony (ABC) [24] and Ant Colony Optimisation (ACO) [9]. These examples showed that the influence graph/interaction network could be used to assess other swarm-based algorithms.

Among all the swarm-based algorithms which were not assessed using the interaction networks, the Cat Swarm Optimisation (CSO) [6, 7] figures as a promising candidate since it is a fairly new algorithm which shares similarities with the PSO and has been applied to tackle a wide list of real-world optimisation problems [11, 18, 19, 25, 27].

In this work, we present a model for the interaction network of the CSO algorithm and the experiments performed allowed us to identify important characteristics of the algorithm regarding the flow of information, convergence capabilities and the influence of one of its parameters. More importantly, this is the first work which looks more deeply into variations that take place within the interaction network by adopting a recently-proposed metric to estimate structural similarities between networks [1]. Using this metric we were able to perform a stability analysis of the algorithm and compare it to the PSO (global and local topology) and the ABC. We also conducted experiments to study how the interaction network is related to the performance of the CSO.

## 2 THEORETICAL BACKGROUND

This section is devoted to explain the theoretical background of this work. It describes the Cat Swarm Optimisation algorithm, the Interaction Networks and the Portrait Divergence metric.

### 2.1 Cat Swarm Optimisation

The Cat Swarm Optimisation (CSO) is a bio-inspired meta-heuristic which was proposed in 2006 by Chu et al. [6, 7]. As in the PSO, the cat (akin to a particle in the PSO) is comprised of a position within the search space (candidate solution), velocity, and a fitness value which quantifies the quality of its current solution.

The main feature of this algorithm is that the cats have two types of behaviour named seeking and tracing. In the seeking, mode the cats perform a local search around their current position using

a mutation operation applied to some of the dimensions of the cat's current position, given by a parameter called *CDC* (Seeking Memory Pool). However, the first step in the seeking mode is to make several copies of itself and apply the mutation to all but one of these copies; the number of copies is a system parameter called *SMP* (Seeking Memory Pool). Note that one of the copies is left intact to represent the original cat. The mutation is described by Equation 1:

$$\vec{c}_{i,d}(t+1) = \begin{cases} (1 + SRD)\vec{x}_{i,d}(t) & \text{if } rand < 0.5 \\ (1 - SRD)\vec{x}_{i,d}(t) & \text{otherwise,} \end{cases} \quad (1)$$

where  $\vec{c}_{i,d}$  is the position of the  $i^{\text{th}}$  copy in the  $d^{\text{th}}$  dimension of the problem,  $t$  is the current iteration,  $rand$  is a random number in the interval  $[0, 1]$  drawn using a uniform distribution, and  $SRD$  is a parameter which controls the radius of the local search. After performing the mutations, the fitness of all (new) cats is calculated and a roulette wheel based on such fitness used to select one cat to replace the original cat.

In the tracing mode, the cats have a position update rule similar to the PSO with a global best topology. Hence, this mode is guided by the current best cat in the swarm and the position is updated according to Equation 2:

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

where  $\vec{x}_i$  is the position of the  $i^{\text{th}}$  cat and  $\vec{v}_i(t+1)$  is the velocity of this cat calculated as described by Equation 3.

$$\vec{v}_i(t+1) = \omega\vec{x}_i(t) + r_1c_1[\vec{x}_{\text{best}}(t) - \vec{x}_i(t)] \quad (3)$$

in which  $\omega$  is the inertia factor,  $r_1$  is a random number in the interval  $[0, 1]$  drawn using a uniform distribution,  $c_1$  is a constant defined by the user, and  $\vec{x}_{\text{best}}$  is the position of the current best cat in the swarm. It is worth mentioning that at the beginning of each iteration the swarm is divided into the group of cats that will perform the seeking mode and the ones who will have the tracing mode. The percentage of cats in each mode is determined by the *MR* parameter. The CSO is summarised in Algorithm 1.

### 2.2 Interaction Networks

The interactions among elements within swarm is a key factor for swarm-based algorithm and the Interaction Networks emerged from the strategy of capturing these interactions into a network structure. In this network, the nodes represent the elements in the swarm and the edges indicate some interaction (i.e. information exchange) between the two nodes. Note that at each iteration a new network is created. The analyses of these structures can be made based on the network of each iteration or by combining several networks using the sum of all the networks in a given interval (time window). Equation 4 shows the definition of the interaction networks ( $I$ )

$$I_t^w = \sum_{t'=t-w+1}^t I(t') \quad (4)$$

where  $t$  is a given iteration,  $w$  is the size of the time window and  $t \geq w \geq 1$ . Figure 1 illustrates the procedure to create the interaction

**Algorithm 1:** CSO Algorithm

---

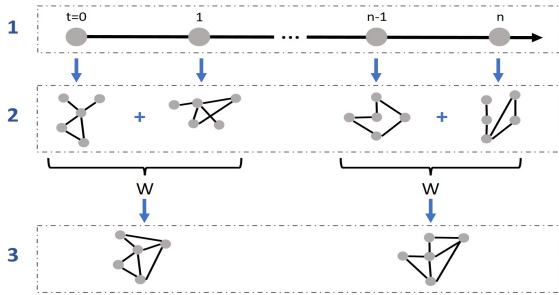
```

1 Initialise all cats positions and velocity randomly;
2 while stop criterion is not reached do
3   Evaluate each cat and update  $\vec{x}_{best}$ ;
4   Choose randomly  $MR\%$  of the cats to perform the
   tracing and the seeking mode;
5   for each cat do
6     if  $\vec{x}_i(t)$  is in seeking mode then
7       Make  $SMP$  copies of the current cat;
8       for each of the  $SMP - 1$  copies do
9         Select  $CDC$  dimensions and update them
         according to Equation 1;
10      end
11      Evaluate the fitness of the  $SMP - 1$  copies;
12      Apply roulette wheel method using the  $SMP$ 
      copies to select the candidate to replace the
      original cat;
13    end
14    if  $\vec{x}_i(t)$  is in tracing mode then
15      Update cat's velocity with Equation 3;
16      Apply Equation 2 to update the cat's position;
17    end
18  end
19 end
20 Return the  $\vec{x}_{best}$  as the final solution.

```

---

networks from the algorithm's interactions. Note that, for  $w = 1$  the set of networks in 2 is equal to the networks on 3.



**Figure 1: The process to create the interaction network. Where 1 represents the iterations of the algorithms, 2 illustrates the networks generated for each iteration and 3 is the networks which result from the application of the time window  $w$ .**

Previous works [9, 14–17, 24] used the interaction networks to assess several aspects of SI algorithms such as the influence of parameters and operators in the performance of the algorithm, exploration/exploitation balance, premature convergence analysis, and communication diversity.

In this work, we intend to expand the set of applications of the interaction networks by measuring the similarities between the networks generated. This approach can be useful to perform a stability analysis of the algorithm (comparing the similarity between

the network of multiple executions of the same algorithm) and to quantify the degree of similarity between two different algorithms.

To be able to measure the degree of similarity between networks we used the idea of *Portrait Divergence* (PD) [1]. According to a study comparing various metrics to compare networks [23], the PD figures as a graph invariant metric suitable to measure structural similarities between networks regardless of them being directed or undirected.

### 2.3 Portrait Divergence

Portrait Divergence is a metric which quantifies the structural similarities of networks [1]. It was proposed by Bagrow and Boltt and performs the comparisons based on the idea of *portraits* of complex networks. A *Network Portrait* is the name given to a matrix ( $B$ -matrix) that encodes the structural information of a given network [2]. The  $B$ -matrix is calculated as indicated by Equation 5.

$$B_{\ell,k} = NP_{\ell}(k) \quad (5)$$

where  $B_{\ell,k}$  is the  $\ell$  row of the  $B$ -matrix and represents the number of nodes which have  $k$  neighbours at a distance  $\ell$ . It is worth noting that the network portrait independent of the network labels which means that networks with the same structure will have the same matrix. Also, for weighted networks, the matrix is calculated using a binning strategy to estimate the distance  $\ell$ .

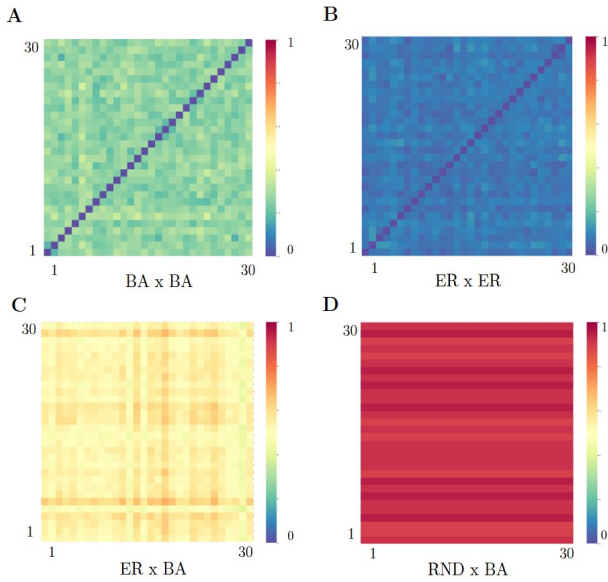
Portrait Divergence is defined based on the portraits of two networks. It uses the Jensen-Shannon divergence to calculate the distance between the two portraits. Moreover, the computational cost of this method is low for small and medium-sized networks [23]. It produces values between zero and one, where zero means that the two networks are identical and one means that they are completely different.

To illustrate the effectiveness of the Portrait Divergence to measure the structural distance between networks, we applied the proposed approach to networks generated from three different models: Erdős–Rényi (ER), Random Regular (RND) and Barabási–Albert (BA) Networks. To determine if there will be a connection between nodes in a Erdős–Rényi network, a random number is drawn from a uniform distribution and if it is greater than a probability  $p$ , a link will be created. The Random Regular is a type of the random networks where all the nodes have the same number of connections ( $d$ ). Lastly, the Barabási–Albert model generates networks in which nodes are connected to  $k$  other nodes and have a tendency to create connections with nodes that are highly connected. For each type of network, 30 networks were generated and the results are depicted in Figure 2.

As can be noticed, when comparing the networks generated from the same model (Figure 2 (A) and (B)) the Portrait Divergence value was less than 0.5; however, the comparison between networks generated from different models presented  $PD > 0.5$ . These results indicate that Portrait Divergence seems to capture the differences between networks.

## 3 EXPERIMENTS AND RESULTS

The first step to model the interaction network of a swarm-based meta-heuristic is to identify in the algorithm the locations in which information is directly or indirectly exchanged among elements in



**Figure 2: Application of the Portrait Divergence to measure the distance (difference) between Barabási–Albert (BA), Erdős–Rényi (ER) and Random (RND) Networks. The BA networks were generated with  $n = 100$ ,  $k = 50$  and  $p = 0.25$ , the ER had  $n = 100$  and  $p = 0.25$ , and the RND used  $n = 100$  and  $d = 5$ . (A) Comparison between 30 different BA networks, (B) 30 different ER networks, (C) comparison between ER and BA and (D) comparison between RND and BA networks.**

the swarm. For the CSO, because in the seeking mode the cats only perform a mutation which is not influenced by other cats, all the interactions take place in the tracing mode. In this case, since the tracing mode is similar to the update rule of particles in the PSO with global best topology, we can adopt a strategy similar to the one used to model the network of the PSO in previous works [14, 15]. This strategy consists of creating a link between the elements in the swarm and the best element of the current iteration. The main difference is that for the CSO the link is established between the best cat and the other cats in the tracing mode.

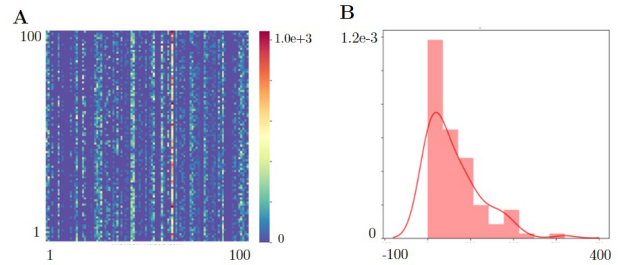
### 3.1 Analysis of the CSO Network

In order to execute CSO we selected the parameters so that  $MR = 20\%$ ,  $SMP = 5$ ,  $CDC = 0.8$ ,  $PMO = 0.2$ ,  $C_1 = 2.0$ , and  $\omega$  decreases linearly from 0.9 to 0.4. The population size was set to 100 and the stopping criteria was a limitation of 500 iterations. The benchmark problem selected was the Sphere function with 100 dimensions and we performed 30 independent simulations of the algorithm. These values were defined based on previous works in the literature [6, 26].

Figure 3 presents the adjacency matrix (i.e. representation of the graph’s connections as a matrix) and the degree histogram of the network created combining the last 10 iterations (time window equals to 10) of the CSO executions. These plots give us information related to the structural characteristics of the network. For instance, if one column of the adjacency matrix has more red colour than

the other, it means that the cat represented by that column guided the swarm in most of the iterations (e.g. a uniform degree distribution in the interaction network can be an indication of swarm convergence).

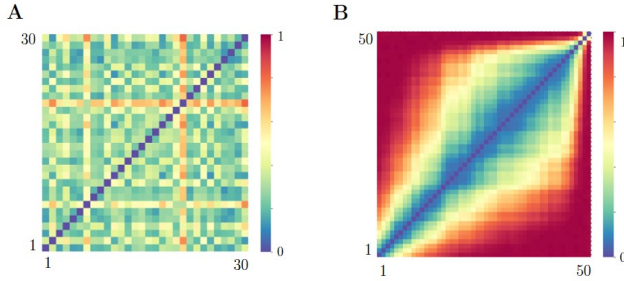
As can be noticed, the network is not fully connected due to the fact that the connections are concentrated around the best cat. Also, the degree histogram indicates the distribution of the number of connection that the cats have during the optimisation. Furthermore, there is no guarantee that all cats will have tracing behaviour during the iterations. In fact, the degree histogram shows that a few cats comprise most of the connections. Notice, however, that the connections are not focused on one single element of the swarm. This indicates that during the optimisation process the best solution, which guides the search process in the tracing mode, is represented by different cats. The changing of the best cat can be important to reduce the chances of getting trapped in a local optimum.



**Figure 3: Adjacency matrix (A) and network degree histogram (B) of the network created combining the last 10 iterations of the CSO.**

In order to measure the impact of the stochastic components of the algorithm on the interaction network, we applied the network portrait metric to compare thirty different executions of the CSO and the result is presented in Figure 4 (A). As can be noticed, in most of the executions, the PD value was less than 0.5 which indicates a moderated degree of similarity between the networks. In other words, the algorithm tends to converge to networks that share similarities in their structure. This is result could be an indication that there is some type of “signature” of the algorithm which is encoded in the network and does not change drastically regardless of the random components of the meta-heuristic. The existence of such a signature could be useful to create methods to classify and compare different SI methods.

Figure 4 (B) depicts the application of the PD to compare the networks generated throughout the iterations. This experiment has the objective to identify if the networks of the algorithm change during the optimisation process. The results illustrated by 4 (B) reviews that the characteristic of the network changes significantly during the optimisation. The initial networks are considerably different ( $PD > 0.8$ ) to the final ones. One possible explanation is that events, such as the convergence of the algorithm, can modify the structural characteristics of the network.



**Figure 4: Characteristics of the CSO's interaction network. Using the Portrait Divergence to compare different executions of the CSO (A) and to compare the evolution of the networks throughout the iterations (B).**

### 3.2 Comparison of CSO with other Swarm Algorithms

To test the hypothesis that we can use this approach to compare two different algorithms, we implemented the networks for the Artificial Bee Colony (ABC) [10] and the Particle Swarm Optimization (PSO) with local best or ring topology (LPSO) and global best topology (GPSO) [4]. The network for the ABC was modelled as described in [24] and the PSO follow the same modelling as the one adopted for the CSO. The trials limit for the ABC was set as 100 and the colony has 100 bees. For the PSO, we employed  $C_1 = C_2 = 1.49$  and 100 particles. we used the same fitness function and these two algorithms were executed 30 times with stop criteria of 500 iterations.

Figure 5 shows the adjacency matrix and the degree histogram of the networks of the CSO (A)(E), GPSO (B)(F), LPSO (C)(G) and ABC (D)(H), respectively. As in the previous case, the network was created combining the last 10 iterations ( $w = 10$ ) of the algorithms' executions. As can be seen in Figure 5 (A) to (D), the network of the CSO is similar to the GPSO since they both share the interactions based on the spread of a global best information. The difference between them is mainly because not all cats perform the position update based on the global best (tracing mode). In fact, the network degree histogram of the CSO is also similar to the GPSO, supporting the argument that both networks have a similar structure.

On the other hand, the network for the ABC presents a more chaotic pattern, which can be explained by the characteristics of the operators in this method. In this algorithm, the interactions occur when the employee bees randomly select a food source, and when the onlooker bees select a food source based on a roulette wheel mechanism. For the LPSO with the ring topology, the network does not experience much change and the interactions are limited to the particle's neighbours. For both LPSO and ABC we can see that the degree distribution is more even (the majority of the nodes possess a similar number of connections).

Concerning how the networks of these algorithms change from one execution to another due to the influence of the stochastic operators, Figure 6 shows that the LPSO is the one with less variation due to its fixed communication topology. The ABC and the GPSO

presents a similar degree of variation, while the CSO falls between the them.

Figure 7 depicts the network comparison over the optimisation process. As can be observed, the pattern displayed on the CSO x GPSO comparison (Figure 7 (B)) is similar to the one displayed for the CSO x CSO (Figure 7 (A)) but with some shift in the results. In fact, the results presented in Table 1 shows that excluding the first and last time windows, the network of the CSO for a  $Network_i$  will have an equivalent GPSO network at  $Network_{i-d}$ , where the value of  $d$  increases over the iterations. Again, this can be associated to the fact that only 20% of the cats displays a behaviour akin to what happens in the GPSO at each iteration.

**Table 1: Best interaction network match between the CSO and the GPSO for a  $w = 10$ .**

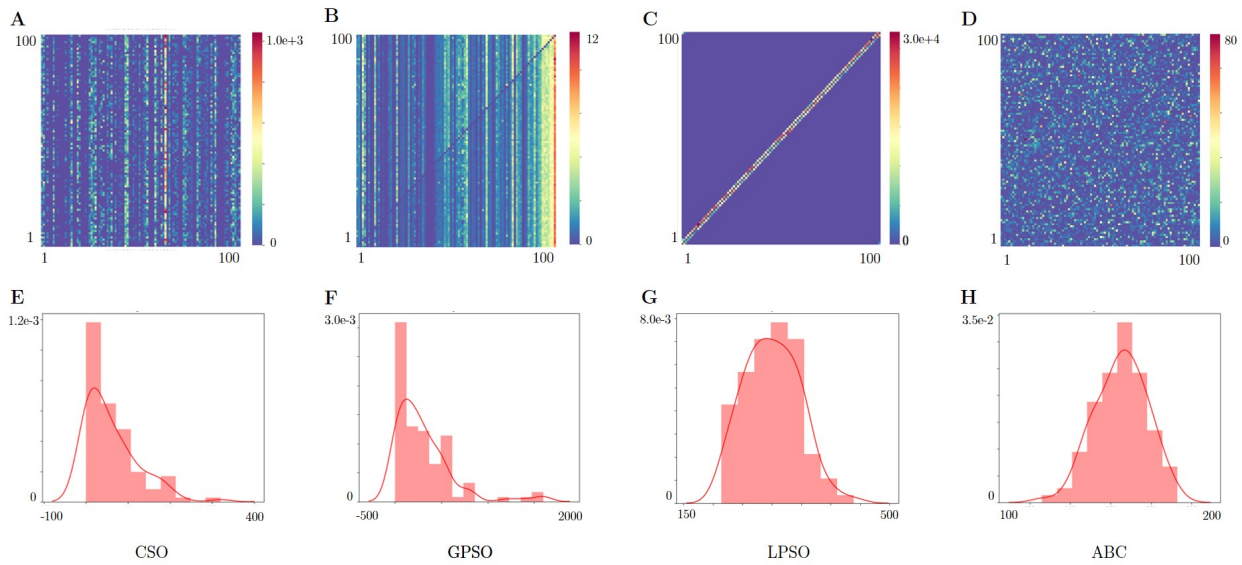
CSO Network	Best GPSO Network Match	PD Value
1	1	0.4039
2	2	0.0825
3	2	0.1458
4	3	0.0752
5	3	0.1221
6	4	0.0750
7	4	0.1273
8	5	0.0714
9	6	0.1807
10	7	0.3237

Regarding the comparison to the LPSO (Figure 7 (C)), we can observe that these algorithms start with a network structure relatively similar to each other and, but as the iterations progress, their structure becomes increasingly different. The highest degree of similarity ( $PD = 0.2959$ ), was achieved between the first time window of CSO and the fourth of the LPSO.

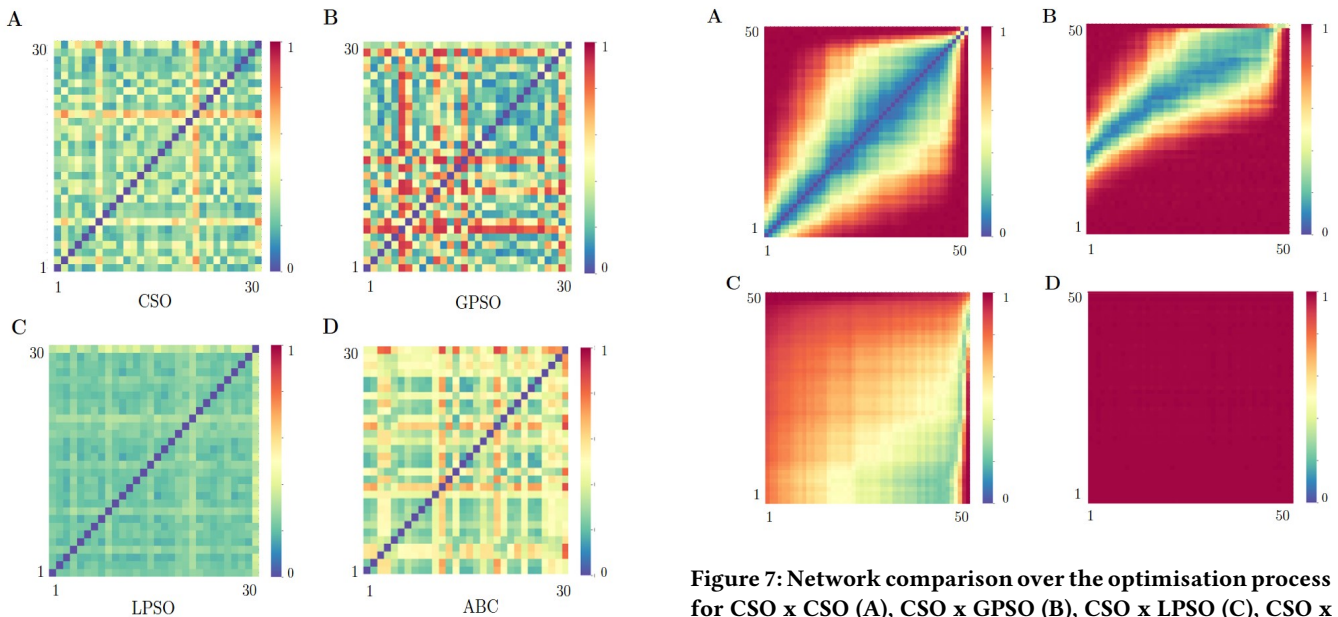
For the ABC, Figure 7 (D) indicates that during all the optimisation process, the degree of similarity between the networks of CSO and ABC was low. In fact, the minimum Portrait Divergence value comparing these algorithms was 0.9862.

From what we have seen so far, the Cat Swarm algorithm is considerably similar to the Particle Swarm Optimisation with global best topology. It was stated that the main difference between the interaction patterns of these algorithms is that in the CSO, a percentage of the cats is selected randomly each iteration to perform the position update based on the global best information. There could be other factors which could also contribute to these differences such as the absence of the  $pBest$  on the CSO and the usage of the mutation operator. However, since the interaction network, at this stage, do not capture auto-loop patterns, we will focus the next analysis on the number of cats performing the PSO-based position update rule.

The  $MR$  parameter controls the percentage of the swarm which performs the tracing mode and Figure 8 presents the network produced from CSO executions with values of  $MR$  equals to 10%, 40%, 60%, and 100%. As can be noticed, as the value of  $MR$  increases the adjacency matrix of the CSO (Figure 8 (A) – (D)) becomes more



**Figure 5: Comparison between the adjacency matrix and degree distribution of the final networks of the CSO (A) and (E), GPSO (B) and (F), CSO (C) and (G) and ABC (D) and (H).**

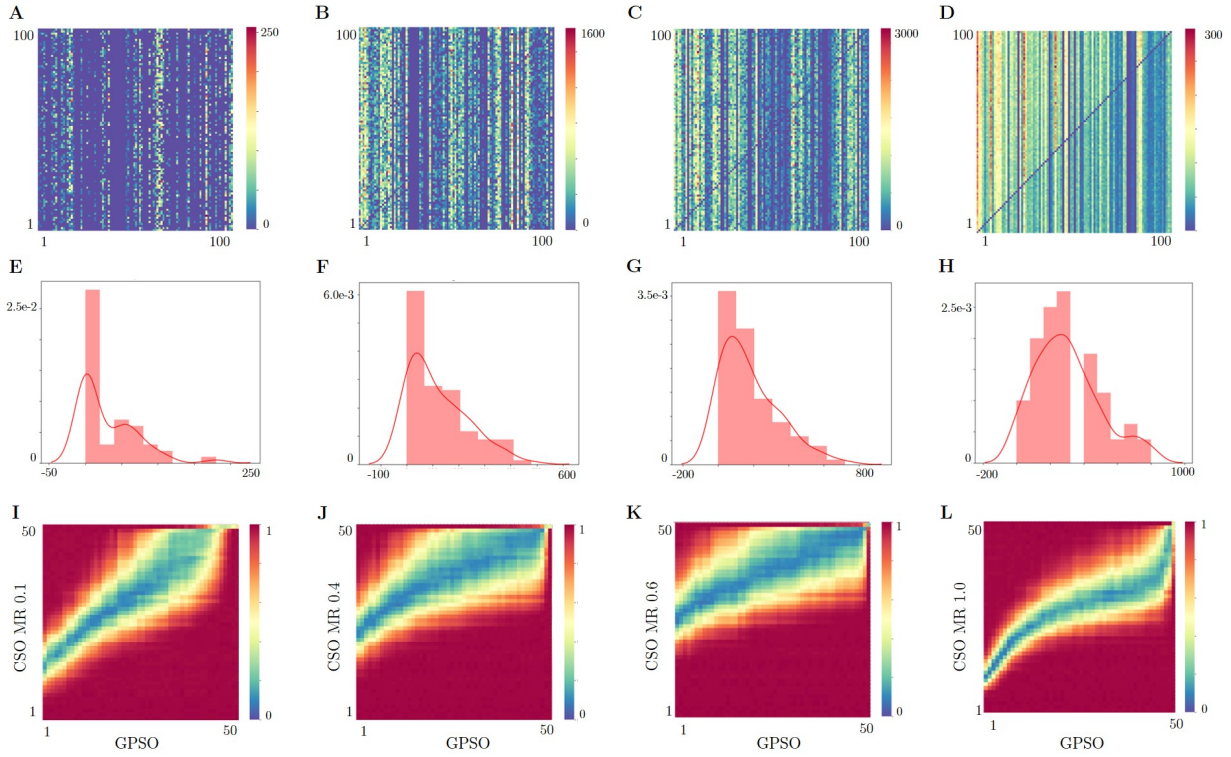


**Figure 6: Comparison between the 30 different executions of the CSO (A), GPSO (B), LPSO (D), and ABC (E).**

similar to the GPSO matrix. Nonetheless, for 100% the degree histogram (Figure 8 (E) – (H)) displays a different characteristic. A possible explanation for this difference is that for  $MR = 100\%$  the seeking mode the swarm only explores regions around the global best location and this could lead to premature convergence of the swarm. This premature convergence makes the fitness of the cats

**Figure 7: Network comparison over the optimisation process for CSO x CSO (A), CSO x GPSO (B), CSO x LPSO (C), CSO x ABC (D).**

more similar to each other, this could make the alternation of the global best more frequent and, as a result, the degree of the nodes in the network become more similar as in the ABC or the LPSO. The PD divergence comparison (Figure 8 (I) – (L)) also displays this behaviour where the network for  $MR = 100\%$  presents fewer similarity points to the GPSO than the network for the other values of this parameter.



**Figure 8: Experiments with different values for the  $MR$  parameter. (A) – (D) shows, respectively, the adjacency matrix for the CSO network with  $MR = [10\%, 40\%, 60\%, 100\%]$ , (E) – (H) shows the degree histogram of these networks and (I) – (L) compares the  $w$  of those networks to the GPSO.**

### 3.3 Relation between the Interaction Networks and the Performance of the Algorithms

The last experiments made aim to identify possible relationships between the characteristics of the interaction network of a given algorithm and its performance in terms of fitness results. Figure 9 shows the results of the Pearson correlation between the  $PD$  value and the fitness difference for  $w = 10$ . The autocorrelation analysis of the CSO, Figure 9 (A), indicates that similar networks ( $PD < 0.5$ ) have a similar fitness value. However, different networks can have similar fitness. Moreover, for this algorithm high difference in the fitness implies that the networks are different ( $PD > 0.5$ ).

Comparing the CSO with the GPSO (Figure 9 (B)) we can see that for  $PD < 0.5$  the CSO presented results similar (fitness difference close to zero) or superior (negative fitness difference) to the GPSO. Nevertheless, given that  $R = 0.18$  we can say that there is no linear correlation between these two metrics of the correlation is weak.

However, for LPSO we can observe a moderate negative correlation. There is also an indication that the bigger is the difference between the networks, the better is the CSO in terms of fitness value when compared to the LPSO. Concerning the ABC, we cannot say that there is a correlation between the  $PD$  value and the fitness difference. It is worth mentioning that, among all the algorithms, positive values for  $R$  were just observed when comparing the CSO to the GPSO.

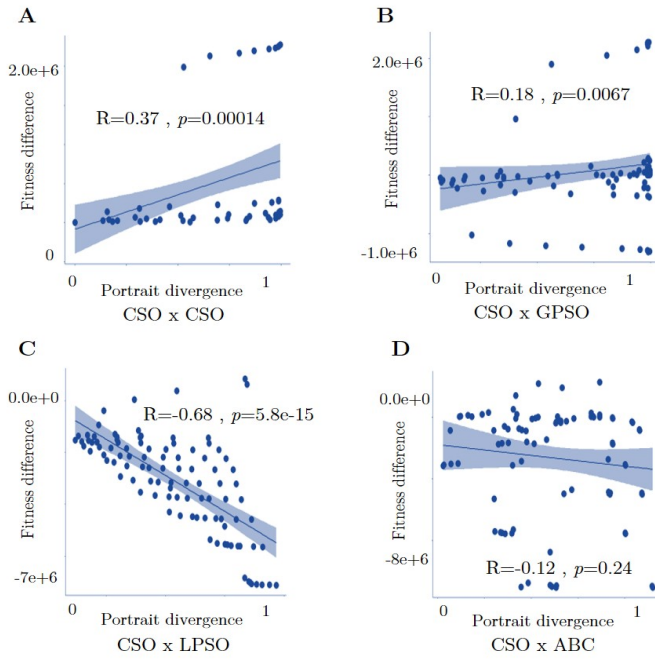
Figure 10 depicts the autocorrelation analysis for the CSO, GPSO, LPSO and ABC algorithm. As in the previous experiments, we can see a degree of similarity between the CSO (Figure 10 (A)) and the GPSO (Figure 10 (A)). Furthermore, Figure 10 reinforces the results illustrated by Figure 6 and the highest correlation was achieved by the algorithm with the lowest variation in its network structure in different executions.

## 4 CONCLUSIONS

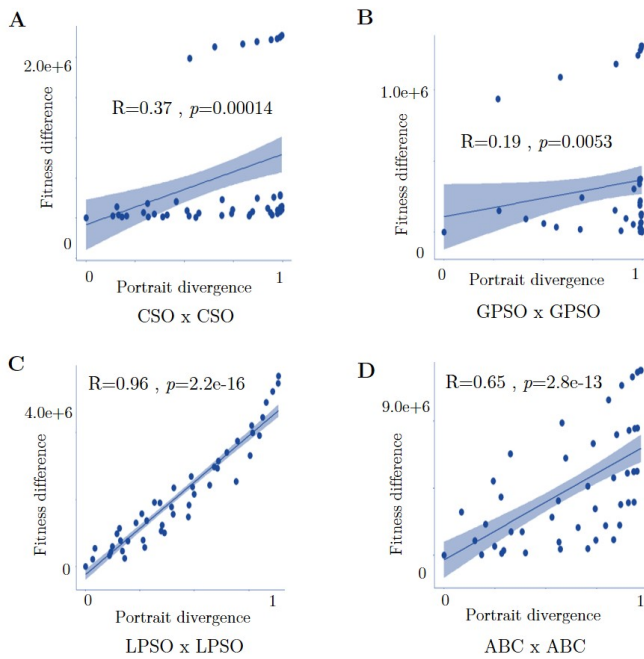
The lack of widely-adopted methods to classify and compare algorithm within the swarm intelligence field is one of the main causes behind the increasing number of proposals of algorithms with a questionable level of novelty. Furthermore, despite the effort made by researchers to better understand these algorithms, there is still a gap in this area.

In this paper, we applied the interaction network to assess the Cat Swarm Optimisation. Using the networks we were able to assess the influence of the  $MR$  parameter, convergence capabilities, information flow in the swarm and, using the Portrait Divergence to compare the structural similarities between the networks, we were able to analyse the stability of the CSO and the evolution of its networks over the iterations.

Furthermore, the usage of the Portrait Divergence also allowed us to perform comparisons between the CSO and ABC and the Particle Swarm Optimisation. The comparison results indicated an



**Figure 9: Correlation between Portrait Divergence and fitness for the CSO, GPSO, LPSO and ABC.**



**Figure 10: Auto-correlation between Portrait Divergence and fitness for the CSO, GPSO, LPSO and ABC.**

elevated degree of similarity between the CSO and the PSO with global best topology, which was expected hence the tracing mode of the CSO is analogous to position update rule of the PSO.

Finally, the experiments performed to study the relation between the difference between the algorithm’s networks and their fitness difference showed that there is a positive correlation between these metrics when the network of the algorithm does not present drastic differences in different executions. In these cases, the more similar the networks the smaller is the difference between their performance in terms of fitness. Moreover, the results indicate that the correlation between two distinct algorithms for those metrics tends to negative.

As future directions we plan the following activities:

- Investigate how other parameters (population size and SRD) influences the structure of the network;
- Analyse the impact of external factors on the network characteristics. For example, how the number of dimensions of the problem and multi-modal functions can affect the network;
- Further assess the relation between the performance of the algorithm and the network structure;
- Experiments with other swarm-based algorithms such as the Fish School Search [3], Grey Wolf Optimiser [13] and The Firefly Algorithm [29];
- Investigate the similarities between variations of the same algorithm;
- Try to adapt the idea of interaction networks for genetic algorithms.

## 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 [1].

## REFERENCES

- [1] James P Bagrow and Erik M Bollt. 2019. An information-theoretic, all-scales approach to comparing networks. *Applied Network Science* 4, 1 (2019), 45.
- [2] James P Bagrow, Erik M Bollt, Joseph D Skufca, and Daniel Ben-Avraham. 2008. Portraits of complex networks. *EPL (Europhysics Letters)* 81, 6 (2008), 68004.
- [3] Carmelo JA Bastos Filho, Fernando B de Lima Neto, Anthony JCC Lins, Antonio IS Nascimento, and Marilia P Lima. 2008. A novel search algorithm based on fish school behavior. In *2008 IEEE international conference on systems, man and cybernetics*. IEEE, 2646–2651.
- [4] Daniel Bratton and James Kennedy. 2007. Defining a standard for particle swarm optimization. In *2007 IEEE swarm intelligence symposium*. IEEE, 120–127.
- [5] Christian Leonardo Camacho-Villalón, Marco Dorigo, and Thomas Stützle. 2018. Why the intelligent water drops cannot be considered as a novel algorithm. In *International Conference on Swarm Intelligence*. Springer, 302–314.
- [6] Shu-Chuan Chu, Pei-Wei Tsai, et al. 2007. Computational intelligence based on the behavior of cats. *International Journal of Innovative Computing, Information and Control* 3, 1 (2007), 163–173.
- [7] Shu-Chuan Chu, Pei-Wei Tsai, and Jeng-Shyang Pan. 2006. Cat swarm optimization. In *Pacific Rim international conference on artificial intelligence*. Springer, 854–858.
- [8] Melvin Gauci, Tony J Dodd, and Roderich Groß. 2012. Why ‘GSA: a gravitational search algorithm’ is not genuinely based on the law of gravity. *Natural Computing* 11, 4 (2012), 719–720.
- [9] Nishant Gurrupadi, Lydia Taw, Mariana Macedo, Marcos Oliveira, Diego Pinheiro, Carmelo Bastos-Filho, and Ronaldo Menezes. 2019. Modelling the Social Interactions in Ant Colony Optimization. In *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 216–224.



- [10] Dervis Karaboga and Bahriye Basturk. 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization* 39, 3 (2007), 459–471.
- [11] Eka N Kencana, Nyoman Kiswanti, and Kartika Sari. 2017. The application of cat swarm optimisation algorithm in classifying small loan performance. In *Journal of Physics: Conference Series*, Vol. 893. IOP Publishing, 012037.
- [12] Michael A Lones. 2020. Mitigating metaphors: A comprehensible guide to recent nature-inspired algorithms. *SN Computer Science* 1, 1 (2020), 49.
- [13] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. 2014. Grey wolf optimizer. *Advances in engineering software* 69 (2014), 46–61.
- [14] Marcos Oliveira, Carmelo JA Bastos-Filho, and Ronaldo Menezes. 2014. Towards a network-based approach to analyze particle swarm optimizers. In *2014 IEEE Symposium on Swarm Intelligence*. IEEE, 1–8.
- [15] Marcos Oliveira, Diego Pinheiro, Bruno Andrade, Carmelo Bastos-Filho, and Ronaldo Menezes. 2016. Communication diversity in particle swarm optimizers. In *International Conference on Swarm Intelligence*. Springer, 77–88.
- [16] Marcos Oliveira, Diego Pinheiro, Mariana Macedo, Carmelo Bastos-Filho, and Ronaldo Menezes. 2017. Better exploration-exploitation pace, better swarm: Examining the social interactions. In *2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*. IEEE, 1–6.
- [17] Marcos AC Oliveira-Júnior, Carmelo JA Bastos-Filho, and Ronaldo Menezes. 2013. Assessing particle swarm optimizers using network science metrics. In *Complex Networks IV*. Springer, 173–184.
- [18] Ganapati Panda, Pyari Mohan Pradhan, and Babita Majhi. 2011. IIR system identification using cat swarm optimization. *Expert Systems with Applications* 38, 10 (2011), 12671–12683.
- [19] Lakshman Pappula and Debalina Ghosh. 2014. Linear antenna array synthesis using cat swarm optimization. *AEU-International Journal of Electronics and Communications* 68, 6 (2014), 540–549.
- [20] Adam P Piotrowski, Jaroslaw J Napiorkowski, and Pawel M Rowinski. 2014. How novel is the “novel” black hole optimization approach? *Information Sciences* 267 (2014), 191–200.
- [21] Dan Simon, Rick Rarick, Mehmet Ergezer, and Dawei Du. 2011. Analytical and numerical comparisons of biogeography-based optimization and genetic algorithms. *Information Sciences* 181, 7 (2011), 1224–1248.
- [22] Kenneth Sörensen. 2015. Metaheuristics—the metaphor exposed. *International Transactions in Operational Research* 22, 1 (2015), 3–18.
- [23] Mattia Tantardini, Francesca Ieva, Lucia Tajoli, and Carlo Piccardi. 2019. Comparing methods for comparing networks. *Scientific reports* 9, 1 (2019), 1–19.
- [24] Lydia Taw, Nishant Gurrupadi, Mariana Macedo, Marcos Oliveira, Diego Pinheiro, Carmelo Bastos-Filho, and Ronaldo Menezes. 2019. Characterizing the Social Interactions in the Artificial Bee Colony Algorithm. *arXiv preprint arXiv:1904.04203* (2019).
- [25] Pei-Wei Tsai, Jeng-Shyang Pan, Shyi-Ming Chen, and Bin-Yih Liao. 2012. Enhanced parallel cat swarm optimization based on the Taguchi method. *Expert Systems with Applications* 39, 7 (2012), 6309–6319.
- [26] Pei-Wei Tsai, Jeng-Shyang Pan, Shyi-Ming Chen, Bin-Yih Liao, and Szu-Ping Hao. 2008. Parallel cat swarm optimization. In *2008 international conference on machine learning and cybernetics*, Vol. 6. IEEE, 3328–3333.
- [27] Zhi-Hui Wang, Chin-Chen Chang, and Ming-Chu Li. 2012. Optimizing least-significant-bit substitution using cat swarm optimization strategy. *Information Sciences* 192 (2012), 98–108.
- [28] Dennis Weyland. 2010. A rigorous analysis of the harmony search algorithm: How the research community can be misled by a “novel” methodology. *International Journal of Applied Metaheuristic Computing (IJAMC)* 1, 2 (2010), 50–60.
- [29] Xin-She Yang. 2010. Firefly algorithm, stochastic test functions and design optimisation. *arXiv preprint arXiv:1003.1409* (2010).