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 nature inspiration. In this work, we employed the concept of *interaction networks* to capture the interaction patterns that take place in 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 make use of *portrait divergence*, a newly-proposed state-of-the-art metric, to assess structural similarities between our interaction networks. Using this approach to analyse the cat swarm optimization (CSO) algorithm, we were able to identify some of the algorithm's characteristics, assess the impact of one of the CSO's parameters, and compare this algorithm to two other well-known methods (particle swarm optimization and artificial bee colony). Lastly, we discuss the relationship between the interaction network and the performance of the algorithms assessed.

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 *Genetic and Evolutionary Computation Conference (GECCO '20), July 8–12, 2020, Cancún, Mexico.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3377930.3390201

Permission to make digital or hard copies of all or part 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 components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
GECCO '20, July 8–12, 2020, Cancún, Mexico
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7128-5/20/07…$15.00
https://doi.org/10.1145/3377930.3390201

1 INTRODUCTION

Several published studies indicate that the number of proposals of "novel" optimisation methods has been increasing in recent years [14, 24]. These are 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 [30]. 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, 10, 22, 23].

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, except for some intial works by Oliveira et al. [17, 19], currently, this 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 exchange positional information of the global/local best particle in the swarm; in the artificial bee colony (ABC) [12] 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. his idea of creating a network to capture the flow of information or interactions patterns in a swarm-based algorithm was explored differently by Clerc [8] and Oliveira et al. [19]. The main difference between the approach of Clerk and Oliveira et al. is that the former has one network where the particles are linked to all its neighbours and another one that models the flow of the update information. On the other hand, the proposal of Oliveira et al. has only one type of network where links are created to connect the particles to the neighbours which provide the information used to update the particles' position. This proposal, known as the *influence graph* and later renamed to *interaction network* and henceforth called as such in this paper, was applied to assess the workings of particle swarm optimisers. 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 interaction 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 interaction network could be used to analyse the search behaviour of the particles and diagnose stagnation.

In 2014, the authors introduced the concept of interaction network with history [16]. The main motivation for the introduction of the concept of history was to have a method to record all the information exchange that occurred within 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 and, by consequence, the performance of PSO.

A new metric which could be applied over the interaction network to obtain information on the behaviour of the algorithm was introduced in 2016 [17]. 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 [18].

Besides the applications of the interaction network to analyse PSO, in 2019 this method was applied on the ABC [26] and the ant colony optimisation (ACO) [11]. These examples showed that the interaction network approach have the potential to be a more general approach 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 [13, 20, 21, 27, 29].

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, called *portrait divergence*, 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 ring topologies) 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 (CSO), the interaction networks and the portrait divergence metric.

2.1 Cat Swarm Optimisation

The cat swarm optimisation is a bio-inspired meta-heuristic 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 (candidates solution), a velocity, and a fitness value which quantifies the quality of this candidate 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* (counts of dimensions to change). 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:

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

where $\vec{c}_{i,d}$ is the position of the i^{th} copy in the d^{th} dimension of the problem, t is the current iteration, r 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 topology. Hence, this mode is guided by the current best cat in the swarm and the position is updated according to Equation 2:

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

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

$$\vec{v}_i(t+1) = \omega\vec{x}_i(t) + c_1\vec{r}_1[\vec{x}_{\text{best}}(t) - \vec{x}_i(t)], \tag{3}$$

in which ω is the inertia factor, $\vec{r}_1$ is a vector containing d random numbers 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* (mixture ratio) parameter. This parameter is defined by the user as a number between 0.0 and 1.0 and does not change during the execution. Also, it is quite important since it controls the exploration/exploitation balance of the swarm. The CSO is summarised in Algorithm 1.

Algorithm 1: CSO Algorithm

Initialise all cats positions and velocity randomly;
while *stop criterion is not reached* **do**
  Evaluate each cat and update $\vec{x}_{\text{best}}$;
  Choose randomly *MR%* of the cats to perform the tracing and the seeking mode;
  for *each cat* **do**
    if $\vec{x}_i(t)$ *is in seeking mode* **then**
      Make *SMP* copies of the current cat;
      for *each of the SMP* -1 *copies* **do**
        Select *CDC* dimensions and update them according to Equation 1;
      end
      Evaluate the fitness of the $SMP - 1$ copies;
      Apply roulette wheel method using the *SMP* copies to select the candidate to replace the original cat;
    end
    if $\vec{x}_i(t)$ *is in tracing mode* **then**
      Update cat's velocity using Equation 3;
      Apply Equation 2 to update the cat's position;
    end
  end
end
Return the $\vec{x}_{best}$ as the final solution.

2.2 Interaction Networks

The interactions among elements within swarm is a key factor for swarm-based algorithms; the interaction network emerge 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[16] shows the definition of the interaction networks (I)

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

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 networks from the algorithm's interactions. Note that, the small is the size of the time window, the more similar are the networks in Figure 1 (2) to the networks in Figure 1 (3).

Previous works 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 [11, 16, 19, 26], exploration/exploitation balance [16, 18], stagnation analysis and communication diversity [16, 17].

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.

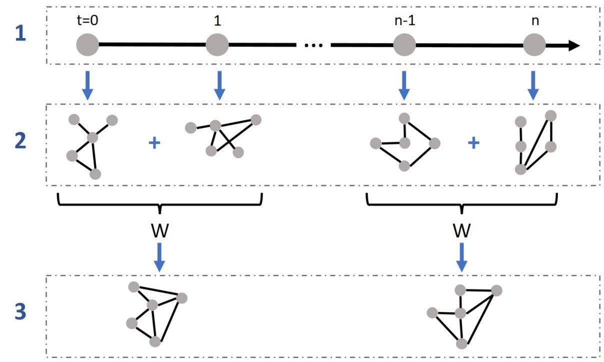

Figure 1: The process to create the interaction network. Where 1 represents the iterations of the algorithms, 2 illustrates the networks generated for each iteration and 3 shows the networks that result from the application of the time window w.

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 [25], 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 performs the comparisons based on the idea of *portraits* of complex networks [2]. A *network portrait* is the name given to a matrix (B-matrix) that encodes the structural information of a given network (e.g. the first row of the matrix contains the number o nodes in the network and the second rows encodes the degree distribution). The B-matrix is a unique, label-independent representation the network and it is calculated as indicated by Equation 5:

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

where $B_{\ell,k}$ is the ℓ row of the B-matrix and represents the number of nodes which have k neighbours at a distance ℓ. It is worth noting that the network portrait is independent of the network labels, which means that networks with the same structure will have the same matrix. Also, for weighted networks, the distance is based on the weight of the links and the B-matrix is calculated using a binning strategy (i.e., instead of counting the number of neighbours that distant are exactly ℓ links, we count the number of neighbours which has a distance between ℓ and $\ell + threshold$).

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 [25]. It produces values between 0 and 1, where 0 means that the two networks are identical and 1 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 using three different models: Erdös–Rényi (ER), Random (RND) and Barabási–Albert (BA) Networks. To determine if there will be a connection between nodes in a Erdös–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 model produces networks with n nodes where the links are defined randomly. Lastly, the Barabási–Albert model generates networks in which new nodes are inserted with k edges which follows the rules of the preferential attachment (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.

It should be noticed that when comparing the networks generated from the same model (Figure 2 (A) and (B)) the portrait divergence (PD) 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.

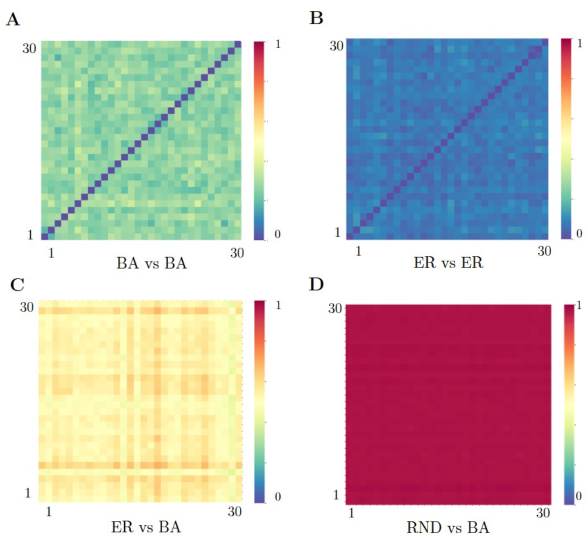

Figure 2: Application of the portrait divergence to measure the distance (difference) between Barabási–Albert (BA), Erdös–Rényi (ER) and Random (RND) Networks. The BA networks were generated with $n = 100$, $k = 50$ and $p = 0.25$, the ER had $n = 100$ and $p = 0.25$, and the RND used $n = 100$ and $d = 5$. (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.

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 individuals in 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 [16, 17]. This strategy consists of creating a link between the elements in the swarm and the best element of the current iteration. For the CSO, only the cats performing the tracing operation will be linked to the best cat of the swarm. We will not model a network for the seeking mode because there is no interaction between the cats in this phase.

3.1 Analysis of the CSO Network

In order to execute the CSO, we selected the parameters so that $MR = 20\%$, $SMP = 5$, $CDC = 0.8$, $PMO = 0.2$, $C_1 = 2.0$, and ω decreases linearly from 0.9 to 0.4. The population size was set to 100 and the stopping criteria is limited to 500 iterations. These values were defined based on previous works in the literature [6, 28]. The benchmark problem selected was the standard Sphere function (no shift nor translation applied) with 100 dimensions and we performed 30 independent simulations of the algorithm.

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. Conversely, a uniform degree distribution in the interaction network can be an indication of swarm convergence.

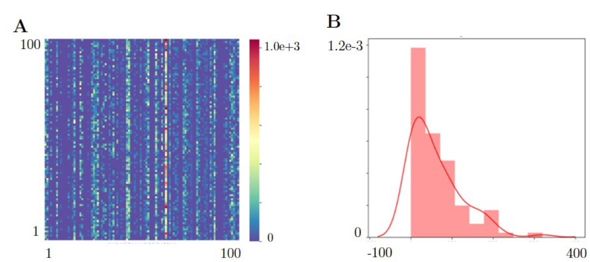

Figure 3: Adjacency matrix (A) and network degree distribution histogram (B) of the network created combining the last 10 iterations (w=10) of the all the 30 executions of the CSO.

As can be noticed in Figure 3 (A), 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 connections 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 the same 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.

In order to measure the impact of the stochastic components of the algorithm on the interaction network, we used the network portrait metric to compare thirty different executions of the CSO and the result is presented in Figure 4 (A).

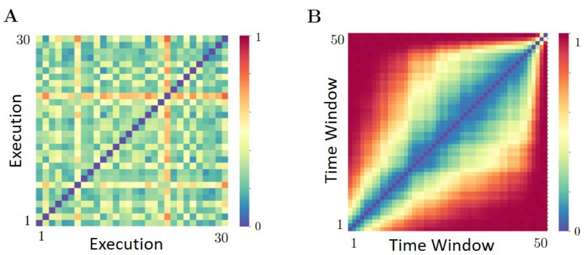

Figure 4: Characteristics of the CSO's interaction network. Using the portrait divergence to compare different executions of the CSO (A) and to compare the evolution of the networks throughout the iterations (B).

As can be seen in Figure 4 (A), 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 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 despite the random components of the meta-heuristic. The existence of such signature, as demonstrated in a previous work [9], can be used to compare the characteristics of 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.

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 [12] and two approached of the particle swarm optimisation: with one local best ring topology (LPSO) and global best topology (GPSO) [4]. We highlight that these two topologies were selected due to the similarities that they share with the CSO. Nevertheless, other topologies could also be used as illustrated in [16, 19]. The network for the ABC was modelled as described in [26] and the PSO follow the same modelling as the one adopted for the CSO. The improvements *trials* parameter of the ABC was set as 100 and the colony has 100 bees. For the PSO, we defined swarm with 100 particles and both the cognitive coefficient (C_1) and the social coefficient (C_2) were set as 1.49. We used the same problem (Sphere with 100 dimensions) used in the experiments with the CSO and these two algorithms were executed 30 times with stop criteria of 500 iterations.

Figure 5 shows the adjacency matrices and the degree histograms 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 Figures 5(A)–(D), the network of the CSO (A) is similar to the GPSO (B) 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 (E) is also similar to the GPSO (F), supporting the argument that both networks have a similar structure. On the other hand, the network for the ABC (D) 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 (C), the network does not experience much change and the interactions are limited to the particle's neighbours. For both LPSO (G) and ABC (H) we can see that the degree distribution is more regular (the majority of the nodes posses 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 (C) is the one with less variation due to its fixed communication topology. The ABC (D) and the GPSO (B) presents a similar degree of variation, while the CSO (A) falls between them.

Figure 7 depicts the network comparison as a function of the optimisation process. As can be observed, the pattern displayed on the CSO x GPSO comparison (B) is similar to the one displayed for the CSO x CSO (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.

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 but, as the iterations progress, their structure becomes increasingly different. The highest degree of similarity was $PD = 0.2959$ and it was significantly smaller than the biggest value for the comparison with the GSPO (B).

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 very 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

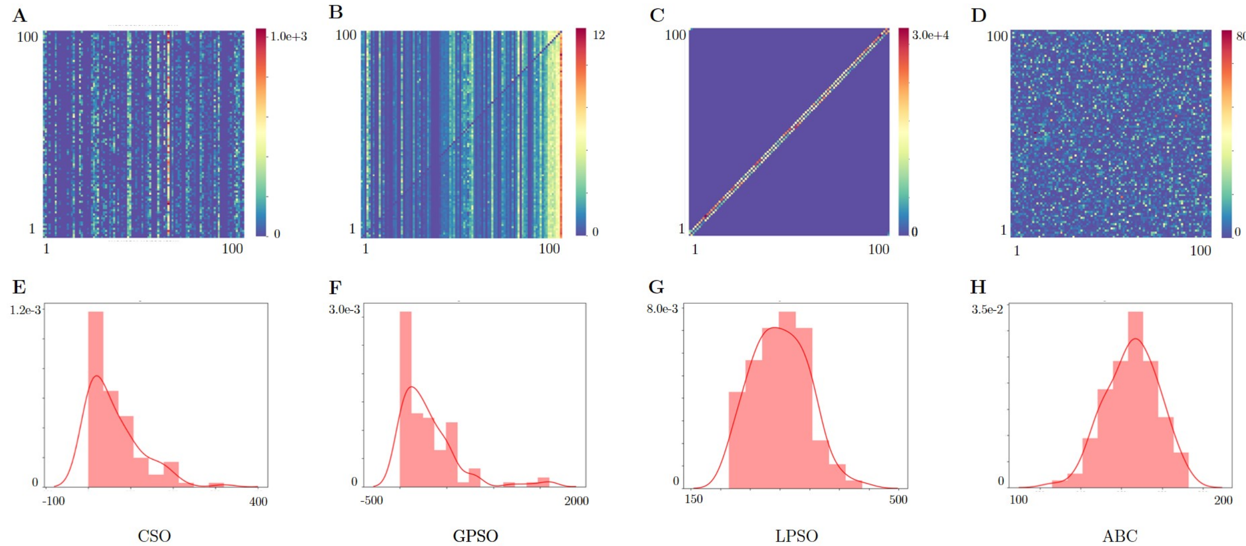

Figure 5: Comparison between the adjacency matrix and degree distribution of the final networks of the CSO (A)(E), GPSO (B)(F), LPSO (C)(G) and ABC (D)(H).

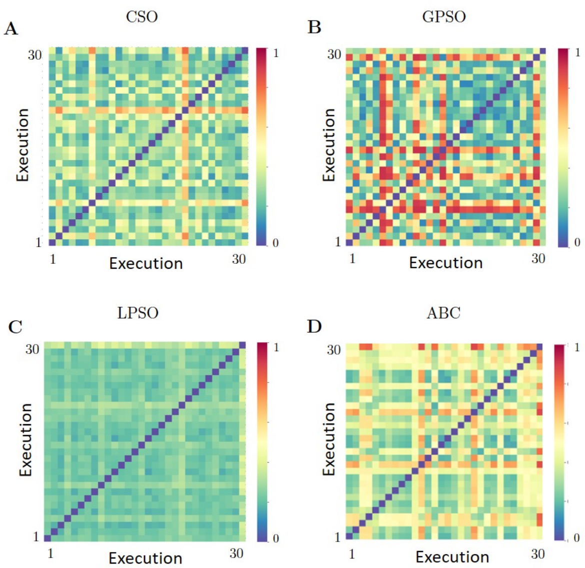

Figure 6: Comparison between the 30 different executions of the CSO (A), GPSO (B), LPSO (D), and ABC (E).

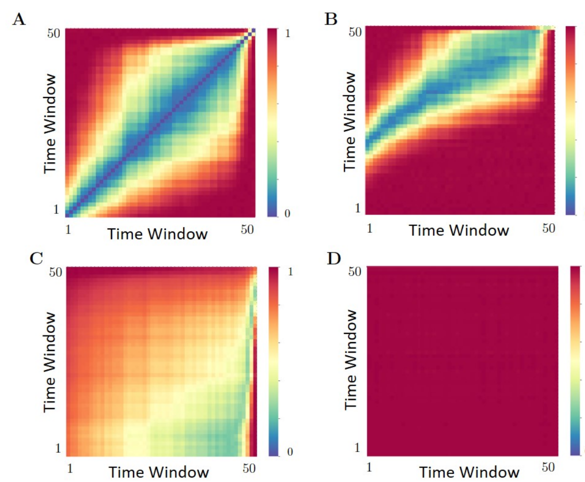

Figure 7: Network comparison over the optimisation process for CSO x CSO (A), CSO x GPSO (B), CSO x LPSO (C), CSO x ABC (D).

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; Figure 8 presents the network produced from CSO executions with values of *MR* equals to 10%, 40%, 60%, and 100%. Not that as the value of *MR* increases the adjacency matrix of the CSO (Figure 8 (A)–(D)) becomes more similar to the GPSO matrix. Nonetheless, for *MR* = 100%, the degree histogram (Figure 8 (H)) displays a different characteristic. A possible explanation for this difference is that for *MR* = 100% the seeking mode

Table 1: Best interaction network match between the CSO and the GPSO for a $w = 50$.

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

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 more similar to each other, which in turn, could make the alternation of the global best more frequent. As a result, the degree of the nodes in the network become more similar, like in the ABC or the LPSO cases. 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.

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 CONCLUSION

The lack of widely-adopted methods to classify and compare algorithms 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. 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.

The usage of the portrait divergence also allowed us to perform comparisons between the CSO and ABC and the PSO. The comparison results indicated an 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 among executions. In these cases, the more similar the networks, the smaller 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) influence the structure of the network;
- Analyse the impact of external factors to the network characteristics. For example, how the number of dimensions of the problem and multi-modal functions can affect the network;
- Experiments with other problems and swarm-based algorithms such as the Fish School Search [3], Grey Wolf Optimiser [15] and The Firefly Algorithm [31];
- 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.

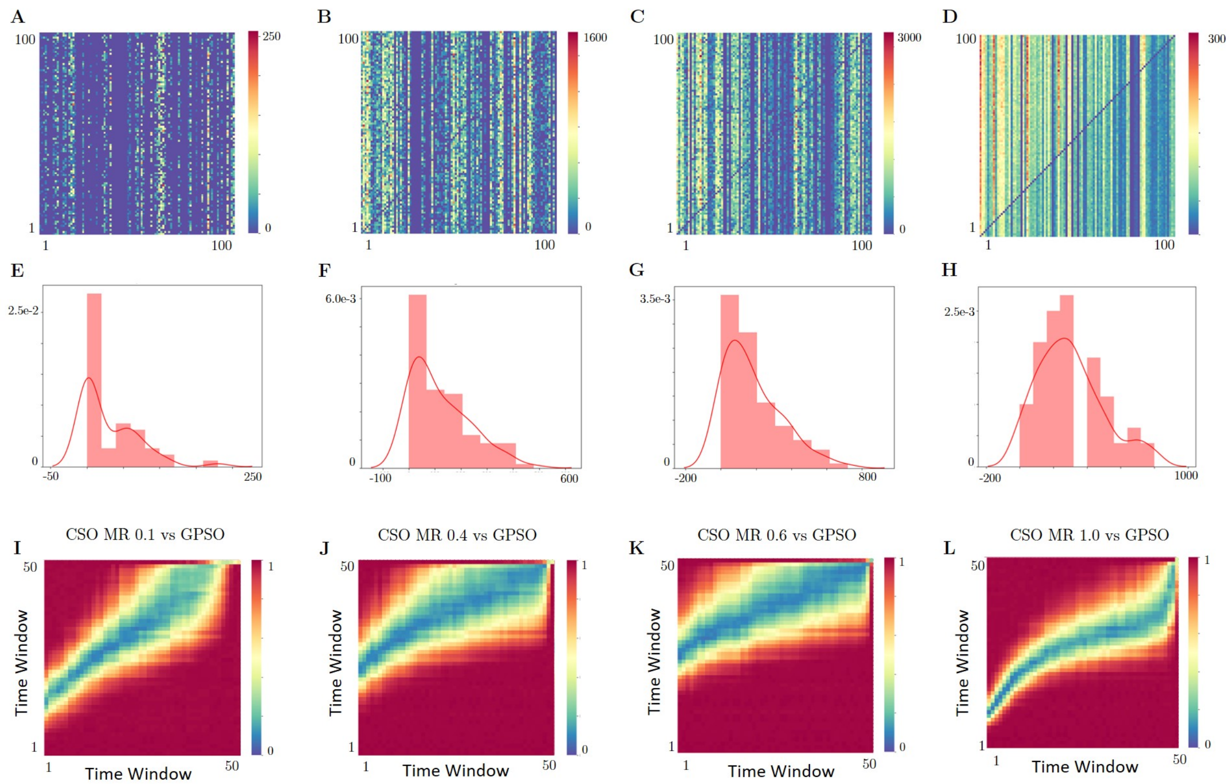

Figure 8: Experiments with different values for the *MR* parameter. (A)–(D) shows, respectively, the adjacency matrix for the CSO network with $MR = [10\%, 40\%, 60\%, 100\%]$, (E)–(H) shows the degree histogram of these networks and (I)–(L) compares the w of those networks to the GPSO.

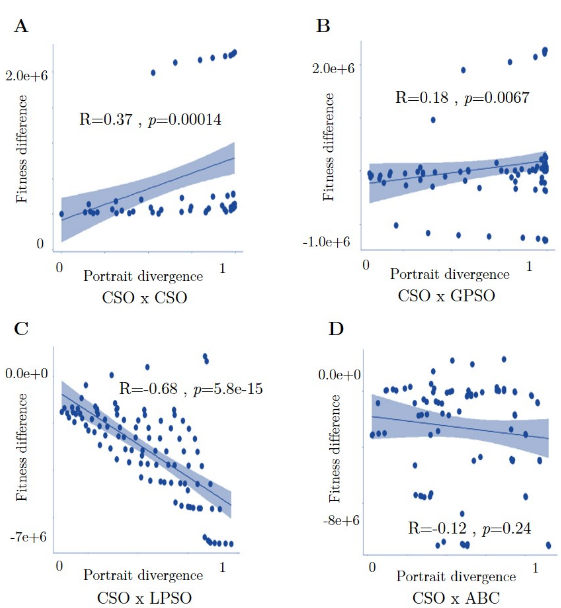

Figure 9: Correlation between portrait divergence and fitness for the CSO, GPSO, LPSO and ABC.

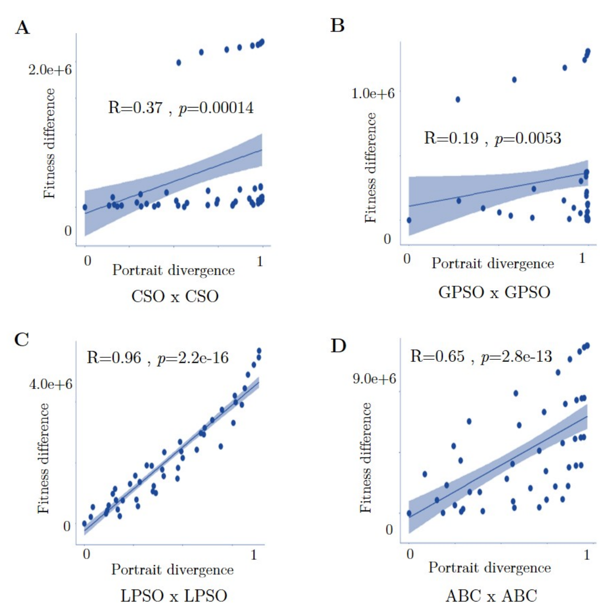

Figure 10: Auto-correlation between portrait divergence and fitness for the CSO, GPSO, LPSO and ABC.

[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] Maurice Clerc. [n. d.]. Particle swarm optimization. 2006. *ISTE, London* ([n. d.]).

[9] Maurice Clerc. 2015. *Guided randomness in optimization*. Vol. 1. John Wiley & Sons.

[10] 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.

[11] Nishant Gurrapadi, 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.

[12] 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.

[13] 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.

[14] Michael A Lones. 2020. Mitigating metaphors: A comprehensible guide to recent nature-inspired algorithms. *SN Computer Science* 1, 1 (2020), 49.

[15] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. 2014. Grey wolf optimizer. *Advances in engineering software* 69 (2014), 46–61.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] Kenneth Sörensen. 2015. Metaheuristics—the metaphor exposed. *International Transactions in Operational Research* 22, 1 (2015), 3–18.

[25] Mattia Tantardini, Francesca Ieva, Lucia Tajoli, and Carlo Piccardi. 2019. Comparing methods for comparing networks. *Scientific reports* 9, 1 (2019), 1–19.

[26] Lydia Taw, Nishant Gurrapadi, 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).

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] Xin-She Yang. 2010. Firefly algorithm, stochastic test functions and design optimisation. *arXiv preprint arXiv:1003.1409* (2010).