

Using Network Science to Assess Particle Swarm Optimizers

Marcos Oliveira · Carmelo J. A. Bastos-Filho · Ronaldo Menezes

Received: date / Accepted: date

Abstract Particle Swarm Optimizers (PSO) have been extensively used in optimization problems, but the scientific community still lacks proper mechanisms to analyze the swarm behavior during the optimization (execution) process. We propose in this paper to assess the swarm information flow based on particle interactions. We define the swarm *influence graph* that captures the information exchanges between the particles in a given iteration during the execution of the algorithm. In order to find a fingerprint of the swarm search behavior, we propose the analysis of this graph by means of its number of components and its overall structure. We simulated the PSO algorithm with three different communication topologies and we showed that each topology leads to different communication signatures. Also, we showed that, in the case of a dynamic topology, this signature is related to the stagnation of the swarm.

Keywords swarm intelligence · network science · particle swarm optimization · swarm behavior assessment

1 Introduction

Computational Intelligence is a set of nature-inspired algorithms that can be applied to solve problems in complex and dynamical environments (Engelbrecht 2007). Some examples of computational intelligence paradigms

include: artificial neural networks, fuzzy system, evolutionary computation and swarm intelligence. Swarm Intelligence is a concept that calls for a system in which actors communicate with each other and act locally on the environment. The interactions between these individuals (swarm) lead to the emergence of solutions to hard problems. Many computational techniques motivated by this behavior constitute what we call Computational Swarm Intelligence—a set of bio-inspired algorithms based on populations of simple reactive agents. Among the most famous approaches in swarm techniques, we can mention: Particle Swarm Optimization (PSO) (Kennedy and Eberhart 1995), Ant Colony Optimization (ACO) (Dorigo and DiCaro 1999), Artificial Bee Colony (ABC) (Karaboga 2005) and Fish School Search (FSS) (Bastos-Filho et al 2008).

PSO was proposed by Kennedy and Eberhart and inspired by the social behavior of flocks of birds searching for food (Kennedy and Eberhart 1995). The technique has been widely used to solve optimization problems in hyper-dimensional search spaces with continuous variables. The main idea behind the technique refers to a population of simple reactive agents (particles) that explore the search space by seeking the best solutions. Each particle has a position representing a candidate solution for the problem and stores the best position it has visited so far. This stored information is used by the particles to update their position and to navigate within the search space. Although different equations may be used to update the particles (Engelbrecht 2007), they modify their position based mainly on their own history and on their neighbors information. Therefore, the way the information exchange happens between the particles plays an important role in the swarm behavior during the search (Mendes et al 2004).

Marcos Oliveira
Florida Institute of Technology, USA
E-mail: moliveirajun2013@my.fit.edu

Carmelo J. A. Bastos-Filho
University of Pernambuco, Brazil
E-mail: carmelofilho@ieee.org

Ronaldo Menezes
Florida Institute of Technology, USA
E-mail: rmenezes@cs.fit.edu

The PSO ability to use different swarm topologies is an example of the importance of the communication between particles on the swarm search behavior. This structure defines the particles that can exchange information with each other (*i.e.* the particle's neighborhood) and its characteristics have high impact on the convergence speed and the quality of the solution obtained by the algorithm (Bratton and Kennedy 2007; Kennedy and Mendes 2002). Less-connected topologies slow down the information flow given that the information is transmitted indirectly through intermediary particles (Kennedy and Mendes 2002). Conversely, highly-connected topologies decrease the average distance between individuals. Consequently, they lead the whole swarm to move quickly towards the first local optimum found by any particle. Indeed, these different behaviors motivated the use of dynamic self-adjustable topologies to manage the information flow during the execution of the PSO algorithm (Suganthan 1999; Peram et al 2003; Janson and Middendorf 2005; Mendes et al 2004; Wang and Xiang 2008; Oliveira et al 2013).

Although some researchers used the swarm topology to analyze the swarm behavior, this structure is only a static representation of the boundaries of the particles communication (Kennedy and Mendes 2002; Mendes et al 2004). Moreover, these analyses regarding the impact of the topology on the swarm performance are generally performed using measures that cannot provide a comprehensive information about the swarm behavior. In general, researchers use measures that do not assess the flow of information within the swarm; they just evaluate simple measures, such as the average distance between particles, and the evolution of the fitness of the particles along the iterations. In fact, as we look further into the literature we conclude that there are no measures to appropriately assess the communication process within swarms.

Furthermore, despite the communication between particles being a fundamental aspect of the swarm behavior, many researchers focus solely on the final result of these particles interactions. For example, many studies have used the particles properties (*e.g.* the position of a particle, the velocity of a particle, etc.) to assess the swarm behavior (Zhan et al 2009; Zhang et al 2011; Pontes et al 2011; Zhou and Shi 2011). Although these approaches perform well with their proposed analyses, they are actually analyzing the final results (particles properties) of the particles interactions. For example, the average distance between particles (the density) in the search space can be used to infer if the swarm has stagnated in a certain area. However, all particles in the same place is actually a result of a lack of diversity in the particles communication. This approach also loses

important information about the swarm behavior. For instance, the particles that attracted all the other particles to a region cannot be distinguished by just evaluating the density of particles in this specific region of the search space. Nevertheless, the particles that have attracted many other particles to this region present a higher influence and can be seen as hubs; hence it might be the case that one can help the entire swarm to escape from a local minimum by just influencing these particular hub-particles. In addition to these aforementioned aspects, one may notice that the approach based on particle's properties may be cumbersome because these properties are dependent on the dimension of the problem addressed by the swarm. Hence, for example, in a swarm that is optimizing a function in 1000 dimensions, any particle's property is also defined in 1000 dimensions, and calculations using these properties (*e.g.* Euclidean distance to calculate the swarm barycenter) will be inconvenient.

We propose the analysis of the particles communication as a way to assess the swarm execution behavior because we believe it to be the core mechanism driving the behavior the swarm. First, in order to have this analysis, we need to capture the information flows within the swarm, thus we define the swarm *influence graph*, that is a graph (or network) representing the information exchange between particles in the swarm. Second, we analyze this network to understand the information flow; we propose the analysis of the influence graph by looking at the number of components present and its structural characteristics. We simulated the PSO algorithm with three different communication topologies and we showed that each topology leads to different communication signatures. Also, we showed that, in the case of a dynamic topology, this signature is related to the stagnation of the swarm.

The paper is organized as follows: we review the Particle Swarm Optimization and some Network Science concepts in Section 2. In Section 3 we define the swarm influence graph and describe the measures to analyze it. In Section 4, the simulation setup and results are presented. Finally, we present our conclusions and suggest some future work in Section 5.

2 Background

In this section, we give a brief explanation of the topics related to this paper. In Section 2.1, PSO is defined focusing on the topologies used by the swarm. Some definitions from network science are given in Section 2.2, together with network concepts that are used in this paper.

2.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a stochastic, bio-inspired, population-based global optimization technique (Eberhart and Kennedy 1995; Kennedy and Eberhart 1995). In PSO, each particle i has a position $\mathbf{x}_i(t)$ at iteration t within the search space and each position represents a possible solution for a d -dimensional optimization problem. The particles move through the problem's search space seeking for the best solutions. Each particle updates its position according to the current velocity $\mathbf{v}_i(t)$, the best position $\mathbf{p}_i(t)$ found by itself and the best position $\mathbf{n}_i(t)$ found in its neighborhood during the search so far. The velocity $\mathbf{v}_i(t)$ and the position $\mathbf{x}_i(t)$ of every particle is updated iteratively by applying Equations (1) and (2), respectively.

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + \mathbf{r}_1 c_1 [\mathbf{p}_i(t) - \mathbf{x}_i(t)] + \mathbf{r}_2 c_2 [\mathbf{n}_i(t) - \mathbf{x}_i(t)], \quad (1)$$

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

where \mathbf{r}_1 and \mathbf{r}_2 are vectors containing random numbers generated from a uniform probability density function within the interval $[0,1]$ at each iteration, for all particles, and for every dimension. The learning factors c_1 and c_2 are the cognitive and the social acceleration constants. They are non-negative constants and weigh the contribution of the cognitive and social components, *i.e.* the second and the third terms of Equation (1).

This basic set of particles' update equations can lead the swarm to what is called an *explosion state*. This behavior arises because these equations allow the particles to increase their velocities indefinitely (*i.e.* they "explode"). Some approaches have been proposed to overcome this issue (Eberhart et al 1996; Shi and Eberhart 1998; Clerc and Kennedy 2002). Clerc and Kennedy developed an approach in which the velocities are constricted by a constant χ , referred as the constriction factor (Clerc and Kennedy 2002). They determined a relation based on this factor that avoids the explosion state, defined according to Equation (3). φ is the sum of the acceleration coefficients c_1 and c_2 .

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \quad \varphi = c_1 + c_2. \quad (3)$$

This factor was introduced to adjust the influence of the previous particle velocities to the optimization process. The final update equation (using χ) for the particle's

position is defined as:

$$\mathbf{v}_i(t+1) = \chi \cdot \left\{ \mathbf{v}_i(t) + \mathbf{r}_1 c_1 [\mathbf{p}_i(t) - \mathbf{x}_i(t)] + \mathbf{r}_2 c_2 [\mathbf{n}_i(t) - \mathbf{x}_i(t)] \right\}. \quad (4)$$

The constriction factor χ also helps to regulate the exploitation-exploration balance of the swarm (Clerc and Kennedy 2002; Bratton and Kennedy 2007). This balance is related to the swarm behavior during the swarm search. The exploration mode is the ability of individuals to broadly explore a region in the search space, while exploitation happens when the search is focused on a specific area of the search space (Kennedy and Eberhart 2001).

2.1.1 Particle Swarm Optimization Topologies

The swarm topology defines the boundaries of the particles communication. The particles only share information with others in the neighborhood defined by the swarm topology. Thus, the flow of information within the swarm during the search is impacted by the topology used by the swarm. Kennedy and Mendes showed the topology structure influences the flow of information among the particles (Kennedy and Mendes 2002). They demonstrated that the presence of intermediary individuals slows down the information flow. Conversely, the information moves faster if more pairs of individuals are directly connected. Thus, when the average distance between nodes is too short, the population tends to move quickly towards the best solution found in earlier iterations. This behavior implies a faster convergence to the global optimum in simple uni-modal problems. However, this fast convergence might force the swarm to prematurely reach a local optimum and loose diversity, specially in multimodal problems (Bratton and Kennedy 2007). In such cases, communication topologies with lower number of connections may reach better results, because the information spreads slowly and the swarm explores different regions of the search space.

Figure 1 depicts some well-known communication topologies used in swarms (Bratton and Kennedy 2007). The global topology was the first topology proposed for the PSO (Kennedy and Eberhart 1995). In this topology, shown in Figure 1(a), all the particles of the swarm are neighbors of each other, leading to a social memory shared by the entire swarm. On the other hand, in local topologies the particles do not share the same neighborhood (Bratton and Kennedy 2007). Consequently, the social memory is not the same for all the particles and is topology dependent. The most used local topology

is the Ring topology (Bratton and Kennedy 2007), in which each particle has only two neighbors, as depicted in Figure 1(b). This structure helps to avoid a premature attraction of all particles to a single location of the search space given that the information spreads slowly; this characteristic comes with the caveat of a slow convergence time (Bratton and Kennedy 2007).

These two topologies lead to extreme opposite behaviors in the swarm, therefore many efforts have been made to propose approaches that have fast convergence while avoiding local minima, such as the ones depicted in Figure 1(c) and Figure 1(d) (Kennedy and Mendes 2002; Mendes et al 2003; Bratton and Kennedy 2007). However, all these proposed structures are usually more appropriate to certain problems (*e.g.* Global topology for unimodal problems) (Kennedy and Mendes 2002). The main reason for this problem dependency is that these topologies are based on arbitrary structures, normally static, and consequently they lead to a non-robust search behavior.

In order to achieve more robustness, some dynamic topologies were proposed (Suganthan 1999; Peram et al 2003; Mendes et al 2004; Janson and Middendorf 2005; Wang and Xiang 2008; Godoy and Von Zuben 2009; Mo et al 2010; Oliveira et al 2013). The structures of these topologies are not static, they change during the search process generally based on specific rules. Oliveira et al

proposed an approach that (self-)adapts the communication scheme based on the swarm state. This dynamic topology tries to change the information flow within the swarm when the particles are stagnating in the search space. The structure modification is based on the preferential attachment mechanism present on the Barabási-Albert model (Barabasi and Albert 1999; Albert and Barabasi 2002). Each particle attempts to create a connection with new particles using a roulette wheel based on the particles' fitness. We call this topology as the *Dynamic topology* henceforth in this paper.

The self-adaptation in the Dynamic topology is based on the state of the swarm. The topology changes when the swarm is stagnating, thus a way to assess the swarm state is needed. In order to do that, each particle has a new attribute, called $P_k failures$, to determine if the particle k is improving its fitness during the search process. Therefore, if the fitness of the particle k does not improve after its iterative position update, $P_k failures$ is incremented, otherwise $P_k failures$ is set to zero. The particle is considered stagnated if $P_k failures$ reaches a preset threshold, a parameter of the algorithm.

In the beginning of the algorithm execution, all the particles are connected based on a structure that allows exploration of the search space. The rationale of this initial scheme is that it is desirable for the swarm to explore the search space in the initial stage of the algorithm run, then in the final steps an exploitation behavior is preferable (Bratton and Kennedy 2007). At each iteration, each particle updates its $P_k failures$ and when the value reaches the preset threshold of failures, the particle k searches for better particles to be connected with, as well as to stop being connected to its current neighbors. This selection of new neighbors is based on a roulette wheel with a linear fitness-based ranking. More details about the Dynamic topology can be found in Oliveira et al (2013).

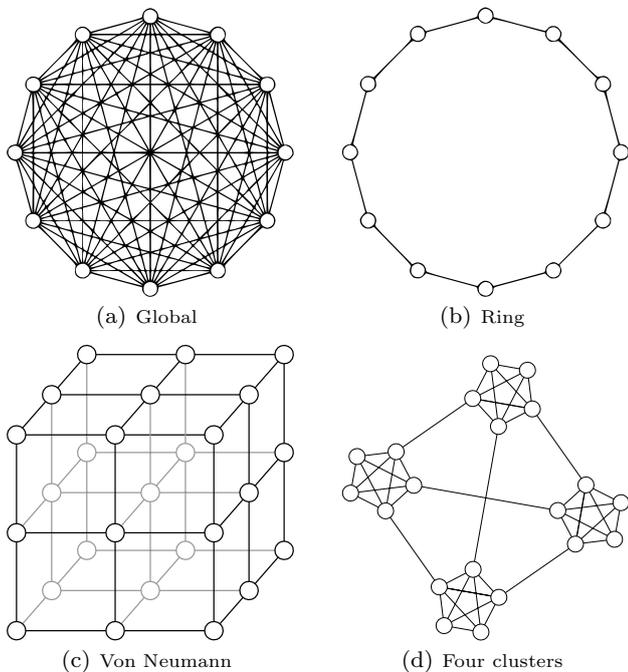


Fig. 1 The swarm topology defines the boundaries of the particles communication. In each network topology, the nodes are the particles and the links represent the possibility of information exchange.

2.2 Network Science Concepts

The analysis made in this paper is a network-based approach to capture the information flow within the swarm and is conducted by modeling the particles' interactions as networks/graphs. A graph G consists of a pair $[V(G), E(G)]$, where $V(G)$ is a set of vertices labelled $1, 2, \dots, n$ and $E(G)$ is a set of edges (*i.e.* pairs of adjacent vertices). Any graph G can be represented by its adjacency matrix $A(G)$, in which its elements are defined as:

$$A_{ij} = \begin{cases} 1, & \text{if vertex } i \text{ and vertex } j \text{ are adjacent,} \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

The spectrum of G consists of the n eigenvalues of $A(G)$, denoted as $\lambda_1, \lambda_2, \dots, \lambda_n$ where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$. The spectral density of a graph can be defined as the density of these eigenvalues and can be stated as a probability density function as follows:

$$\rho(\lambda) = \frac{1}{n} \sum_{j=1}^n \delta(\lambda - \lambda_j), \quad (6)$$

where n is the number of eigenvalues and δ is the Dirac function.

Many characteristics of the graph can be assessed by analyzing the graph spectrum (Cvetković et al 2010). Actually, the spectrum is considered a fingerprint of the networks that can be used to characterize them (Dorogovtsev et al 2003). Farkas et al showed that topological features of some kinds of graphs (uncorrelated random networks, the small-world networks, and the scale-free networks) can be identified by its graph spectral density (Farkas et al 2001). They also proposed some practical tools for the identification of the basic types of random graphs and for classification of real-world networks. These tools are based on the extremal eigenvalues of the adjacency matrix. Farkas et al showed that these eigenvalues contain useful information about the structure of the graph. Depending on the periodicity of this structure, the principal eigenvalue is detached from the rest of the spectrum. Thus, they proposed a quantity named R , defined as:

$$R = \frac{\lambda_1 - \lambda_2}{\lambda_2 - \lambda_n}, \quad (7)$$

that measures the distance of the first eigenvalue from the main part of $\rho(\lambda)$.

The R -value can be used in order to distinguish between some graph-structure features: (i) periodical or almost periodical; (ii) uncorrelated and non-periodical; and (iii) strongly correlated non-periodical. For example, a regular graph has a structure that is periodical, conversely a random graph is non-periodical. However, when the probability in a random graph is not constant and depends on the nodes involved, the result is a correlated structure. In large systems, R -value of the sparse uncorrelated random graph converges to a constant, while R -value in the scale-free model decays as a power-law function of the number of the nodes.

The Laplacian matrix $L(G)$ of a simple graph G is the matrix $D(G) - A(G)$, where $D(G)$ is the degree matrix of graph G . Some properties of a graph can be inferred by using the Laplacian matrix, its eigenvalues $v_1(G) \geq v_2(G) \geq \dots \geq v_n(G)$ relate to many of these properties. For instance, the multiplicity of 0 as an eigenvalue of L is equal to the number of components in G (Cvetković et al 2010). Thus, the second-smallest

eigenvalue of L is equal to zero if and only if G is a non-connected graph. This eigenvalue is called the algebraic connectivity (or Fiedler value) of G and its magnitude reflects how well the graph is connected.

3 Assessing Particle Swarm Optimizers

In this section, a structure called the swarm influence graph is defined in order to capture the flows of information within the swarm. Moreover, some measures are described to analyze these flows by analyzing this proposed structure. In Section 3.1 the aforementioned structure is defined and the measures are presented in Section 3.2.

3.1 The Swarm Influence Graph

Although the swarm communication topology defines which particles can communicate with one another, the topology only bounds the particles communication range. Actually, the information flow changes at each iteration when the particles share information with their best neighbors. That is, at iteration t , each particle i only gets information from its best neighbor $n_i(t)$. Thus, a network of information exchanges between particles in a given iteration can be used to assess the information flow within the swarm. Therefore, we define

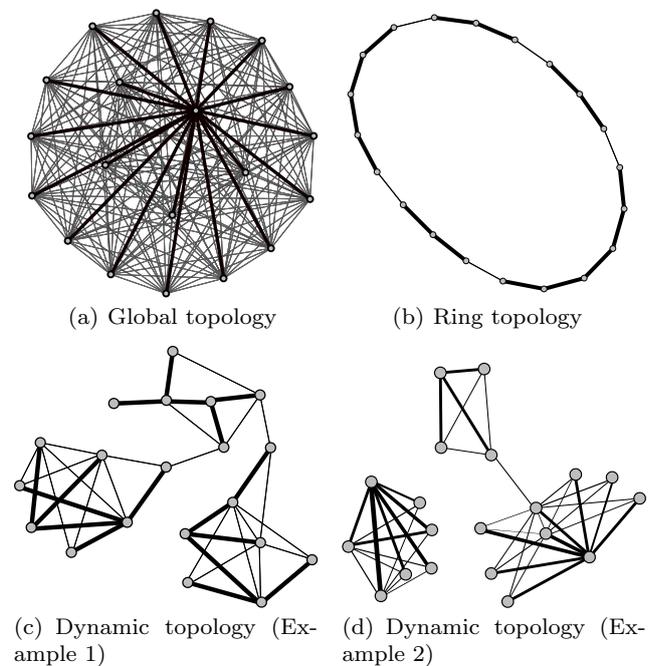


Fig. 2 Examples of the swarm influence graph over the topology for the three communication topologies. These examples suggest that the swarm topology impacts the structure of the influence graph.

the influence graph as the representation of this network. The elements of the swarm influence graph I'_t at iteration t are defined as follows:

$$I'_{t_{ij}} = \begin{cases} 1, & \text{if } n_i(t) = j, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

where $n_i(t)$ is the neighbor of particle i at iteration t .

One can observe that the influence graph I'_t is a directed graph by definition. The direction of the edges is the actual direction of the information exchange between particles. The simplified influence graph I_t is defined by removing the edge direction. The elements of this graph at iteration t are described as follows:

$$I_{t_{ij}} = \begin{cases} 1, & \text{if } n_i(t) = j \text{ or } n_j(t) = i, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

In other words, the simplified influence graph is the network of the particles that shared information between them at a given iteration. Figure 2 depicts examples of simplified influence graphs (in bold) over three different swarm topologies. These examples suggest that the swarm topology impacts the structure of the influence graph. For example, in the case of the Ring and Dynamic topologies, there are some sub-graphs, whereas the Global topology presents only a focal point in the influence graph. One must notice that the existence of two particles in the same component does not imply they share information, actually they only do if they are directed connected, otherwise they only share common particles that are indirectly exchanging information.

The swarm influence graph is a forest, that is a graph in which each connected component is a tree. The components in the influence graph are related to the information flows within the swarm. Therefore, the number of components as well as their structure are associated with the swarm behavior, and the analyses in this paper are made on these two aspects.

The simplified influence graph is the one used to analyze the swarm information flow in this paper because the analysis of the structure of a undirected graph is simpler than a directed one (*e.g.* the eigenvalues of the adjacency matrix of a directed graph may be not real). Although the direction is removed from the edges in the simplified graph, there is still implicit information about the influence of the nodes. This is the case because of the tree structure of the influence graph and the fact that every particle must retrieve information from other particle, once the PSO version being analyzed is a selfless model, that is, the particles always use the neighbors' information to evaluate $\mathbf{n}_i(t)$ (Kennedy 1997). For example, Figure 3 is a small tree that depicts a possibly influence graph of three particles, namely a , b and c .

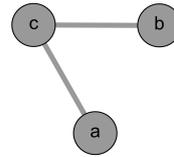


Fig. 3 The structure of the simplified influence graph still provides information about the swarm behavior. In this example, a simplified influence graph of three particles is derived from one of the graphs in Figure 4.

The graph in Figure 3 can be the result of the direction removal process of the follow structures depicted in Figure 4. One must notice that the structures in Figure 4(b) and Figure 4(c) can be seen as the same, the difference between them is just the labels, thus they are treated as the same. Furthermore, once this is a selfless model and a particle must retrieve information from its neighborhood, the structure in Figure 4(a) may also be understood in the same way as the other ones. Therefore, when the labels are disregarded, these structures are the same, as depicted in Figure 5.

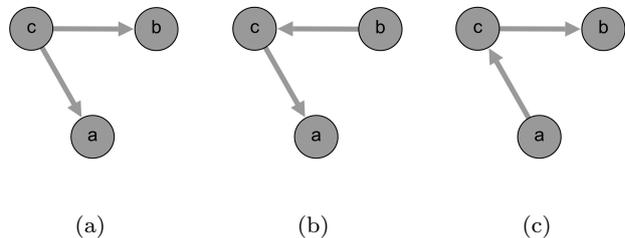


Fig. 4 The possible directed influence graphs of the structure in Figure 3. Due to the tree structure of the influence graph and the fact that every particle must retrieve information from other particle, all of them can be seen as the same.

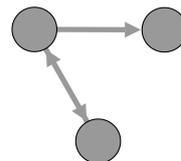


Fig. 5 The origin of the structure in Figure 3 when the labels are disregarded. There is no loss of information when the directions are removed from these trees.

Such understanding of the swarm influence graph can be also easily obtained in a tree of two particles, a case in which there is only one possible configuration: both particles sending and receiving information. Thus, there is no loss of information when the directions are removed from these small-sized trees. On the other hand, bigger trees are more complicated to analyze and indeed there is loss of information. However, the structure of these trees still contains informative aspects of the swarm communication. For example, a particle with degree k in a tree of the influence graph transmits information to at least $k - 1$ other particles (and at most to k particles). Thus, the higher is the degree of a particle, the higher is the impact of this particle on the others. Therefore, although the influence graph is simplified, the analysis of its structure still provides information about the influential role of the particles involved, as well as the information flow.

3.2 Analyzing the Swarm Influence Graph

In this paper, the swarm influence graph is analyzed by its number of components and its structure. In Section 3.2.1, the use of the number of zero-valued eigenvalues of the Laplacian matrix is discussed as a way to count the number of information flows. In order to analyze the structure of these flows, the density spectrum is used and discussed in Section 3.2.2. Once the analysis of the spectrum may be cumbersome, the R -value is used to assess the information flow and is also discussed in Section 3.2.2.

3.2.1 The Number of Zero-valued Eigenvalues in the Laplacian Matrix

The components in the swarm influence graph (*i.e.* the sub-graphs in the case of undirected graphs) are groups of particles that shared information with the same particles; in such a manner that if two particles are not in the same component, they have different sources of information to calculate their velocities. Thus, these components can be seen as different flows of information in the swarm. The number of these flows within the particles is related to the swarm diversity. For instance, a swarm with few information flows does not have different sources of information regarding different regions of the search space. In this case, the swarm may contain low diversity, which drives all the particles towards the same place in the search space. Conversely, a swarm with many different information flows explores the search space as a whole because there are many sources of information. These two distinct search behaviors with different number of information flows also

relate to the exploration–exploitation balance in the swarm. When the particles are exploring broadly a region in the search space (*i.e.* exploration search mode), there are different information flows in the swarm. On the other hand, when the search is focused in an area (*i.e.* exploitation search mode), the swarm diversity is low.

The multiplicity of “0” as an eigenvalue of the Laplacian matrix of the swarm influence graph is equal to the number of components in this graph. Thus, in order to count the number of information flow in the swarm, we analyze the influence graph with this measure.

3.2.2 The Density Spectrum and R -value

The analysis of the structure of the information flow within the swarm may elucidate aspects of the swarm behavior. For instance, although particles in the swarm can share information with the same particles, there is the possibility of a particle being more influential than others in the same flow. This is the case, for example, when particles transmitting information to many others are present in the flow. This is highly related to the diversity of the swarm since the existence of impactful particles may lead the other particles to the same region of the search space. Thus, the way information navigates within the swarm is described by the structure of the information flow, that impacts the swarm behavior. Therefore, the analysis of the structure of the swarm influence graph may explain these intricate behaviors of the particles during the search process.

The spectra of eigenvalues of the graph adjacency matrix is considered a fingerprint of the networks that can be used to characterize them (Dorogovtsev et al 2003). Thus, we use the density spectrum of the swarm influence graph to find the fingerprints of the swarm search behaviors. That is, this analysis can show a signature of the type of the search being performed by the swarm. Still, this analysis may be cumbersome because the density spectrum analysis we perform is mainly a plot analysis. In general, the spectral density can be mainly used to assess the behavior of algorithms or to diagnose the influence of a parameter in the algorithm performance.

Considering this, we also deployed the R -value since it can also be used as a way to understand the characteristics of the network. Besides, the R -value is a single indicator and can be used to control the operation of the algorithm. It is worthy to mention that the R -value can be used together with other measures, such as the evolution of the fitness or the density of particles in a certain region of the search space.

4 Simulation Setup and Results

4.1 PSO Setup

In order to assess the proposed methodology, we used the PSO algorithm to optimize two well known multimodal benchmark functions, named *F6 function* and *F8 function*, that were proposed in 2010 as large-scale optimization problems (Tang et al 2010). The *F6 function* is a single-group shifted and m -rotated Ackley's function and *F8 function* is a single-group shifted and m -dimensional Rosenbrock's function. The former is a multimodal function and the latter is also a multimodal function very dependent on the initial values. We selected these two different functions to have insights of the swarm behavior in two distinct scenarios.

In all experiments, we used 1,000 dimensions and m equal to 50. We used 200 particles in all simulations. For each simulation trial, we used 300,000 fitness function evaluations. We performed simulations for the Global, local and Dynamic topologies. In the case of the Dynamic topology, the threshold of failures for the particles was set to $P_k failures = 50$. The particles were updated according to Equation (4) with $c_1 = 2.05$ and $c_2 = 2.05$ as indicated in (Clerc and Kennedy 2002), which guarantees no explosion state.

4.2 The Number of Information Flows

The evolution of the number of information flows within the swarm during the execution of the PSO optimizing the *F6 function* is shown in Figure 6(a). It shows the

behavior of this quantity as a function of the number of iterations for the three considered topologies.

The influence graph in the Global topology is an one-component star-like graph and keeps its structure during the whole execution of the algorithm; the number of zero eigenvalues is constant and equal to one. Although the value is equal to one in this case, it does not mean that the influence graph is the same along the entire process. The best particle of the swarm can change along the iterations which causes the center of the star-like topology to change.

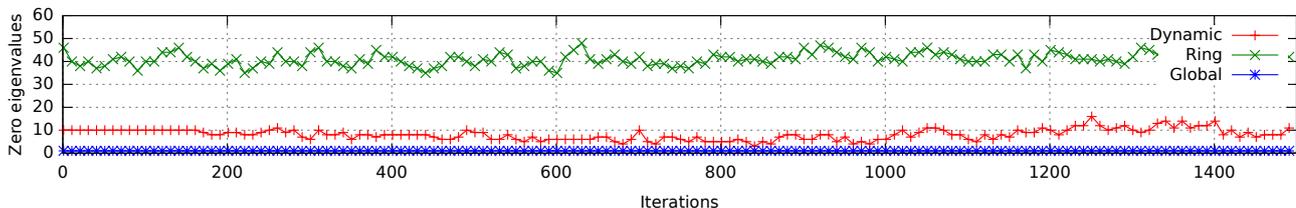
Although the Ring topology is static, the influence graph can present different sub-graphs. One can observe that the number of information flows varies through the iterations, but presents a high average value of 40 along the entire process.

In the Dynamic topology, the algorithm starts with 10 information flows and diminishes in average to 5 information flows along the algorithm execution. One must observe that the Dynamic topology presents a balanced behavior between the two static approaches.

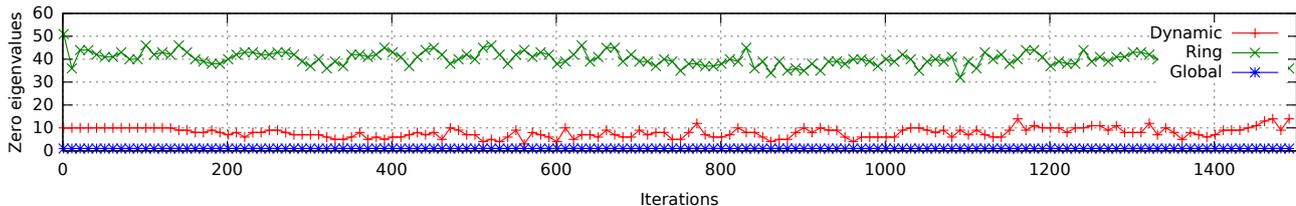
Figure 6(b) depicts the same analysis on the *F8 function*. One can observe that the curves are similar to the ones shown in Figure 6(a). This similarity can also be seen with all benchmark functions in (Tang et al 2010).

4.3 Analyzing the Structure of the Information Flows with the Density Spectrum

In order to analyze the eigenvalue spectra of the swarm influence graph, we divided the analysis in two parts: static topologies and the Dynamic topology. First, we



(a) F6 function with different topologies.



(b) F8 function with different topologies.

Fig. 6 Number of zero eigenvalues of the swarm influence matrix for Dynamic, Ring and Global Topologies for two different benchmark functions.

try to capture the structural characteristics of the static topologies. Thereafter, we try to analyze the performance of the algorithm by its structural characteristics when the swarm has a Dynamic topology.

The static topology analysis is given in Section 4.3.1 and analysis of the Dynamic topology is shown in Section 4.3.2.

4.3.1 Static topologies

The spectral density has the capacity to represent the frequency of the eigenvalues. Therefore, it is interesting to evaluate the characteristics of the topologies as a function of the number of iterations in order to examine the swarm behavior through iterations.

As the Ring and Global topologies present a well known behavior in terms of connectivity and convergence, we first evaluated the evolution of the spectral density of the influence graph as a function of the number of iterations for these two topologies.

The behavior of the spectral density of the influence graph along the iterations of the PSO when using

a Global topology is depicted in Figure 9(b). The snapshots of the spectral density are from iterations 100, 400, 800 and 1200. One can observe that the Global topology presents a perfect uni-modal shape. Moreover, the shape does not vary along iterations.

Figure 9(a) depicts the curve of the spectral density along the iteration for the PSO with local topology. The shape is bi-modal, unlike the density spectrum shape of the swarm with a Global topology. Nevertheless, the curve also does not change along the iterations.

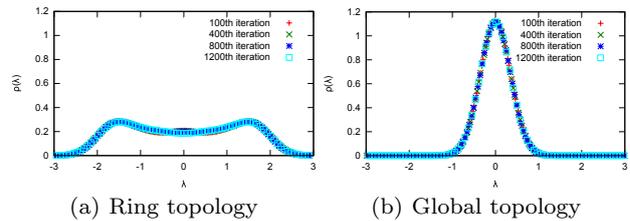


Fig. 9 Density spectrum of the swarm influence graph for (a) the Ring topology and (b) the Global topology (right).

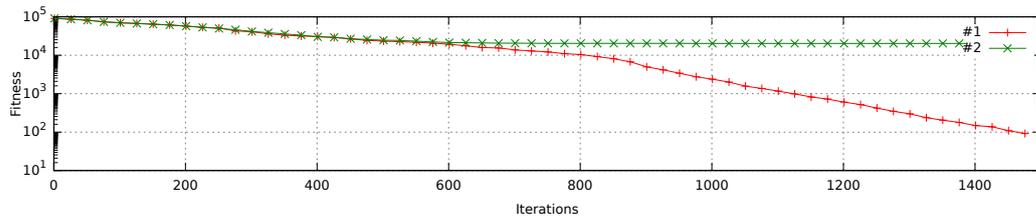


Fig. 7 The performances of two independent runs of the PSO algorithm with the Dynamic topology optimizing $F6$ function.

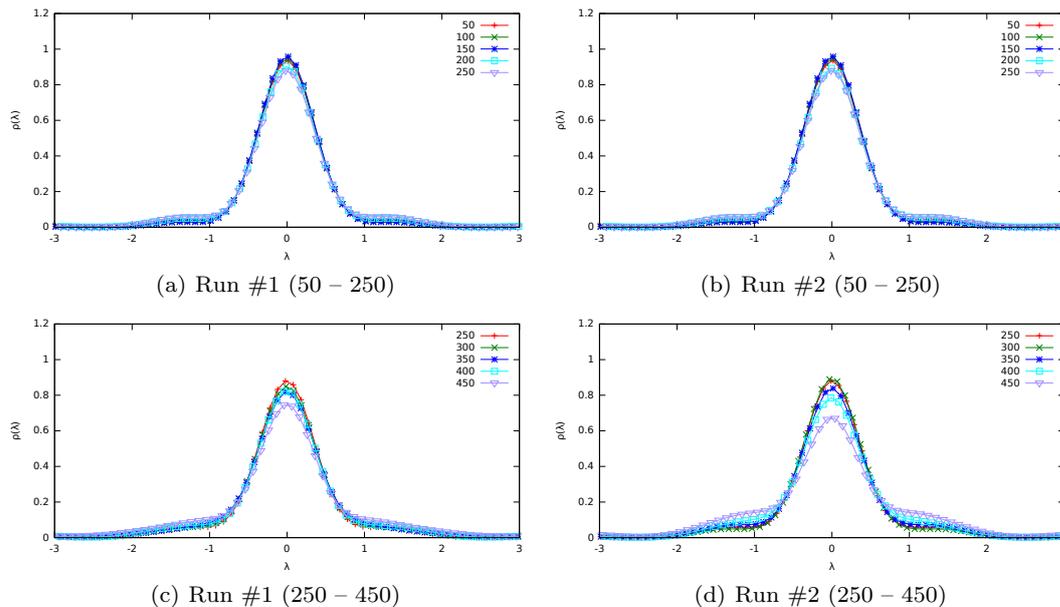


Fig. 8 The density spectrum of the Swarm Influence Graph from different iterations of the two runs with performances depicted in Figure 7. The curve for Run #2 is flattened out more quickly than the curve from Run #1.

4.3.2 Dynamic topology

In order to analyze the information flow of the swarm and to assess the performance of the proposed approach by means of the information disseminated in the swarm, we use two different runs of the PSO algorithm. These different runs optimize one of benchmark functions used before (*F6 function*), but one of these runs gets stagnated along the iterations and the other has success.

Figure 7 shows two different trials of the PSO with the Dynamic topology (Run #1 and Run #2) optimizing *F6 function*. As seen in the figure, the results of these two independent runs are quite different. In Run #1, the algorithm converged, while in the Run #2 the algorithm got stuck in a local minimum. Figure 8 shows evolution of the spectral density of the influence graph for these two runs for the iterations 50 to 450 with increments of 50. In Run #2, the curve is flattened out more quickly than the curve from Run #1. Moreover, in both cases, the structure in the first iterations is similar to the one when the swarm has the

Global topology. However, one can observe that both cases present side lobes, which represents a combination of local and Global behaviors.

Two runs of the PSO with the Dynamic topology optimizing *F8 function* are shown in Figure 10. The fitness values for these two independent runs are different just at the end of the algorithm execution. The fitness is slightly better for Run #1 when compared to Run #2.

Figure 11 shows the evolution of the spectral density of the influence graph for these two runs for the iterations from 50 to 1000, with increments of 50. Again, in Run #2, the curve is flattened out a little more quickly than the curve from Run #1. This might explain the difference in the final fitness. However, one must observe that the difference in the shape of the curve is not so evident as in the former case (*F6 function*).

Hence, although interesting, from our results alone, we still cannot define a rigid rule between the flattening rate of the spectral density and the stagnation level of the swarm. However, the information provided by the

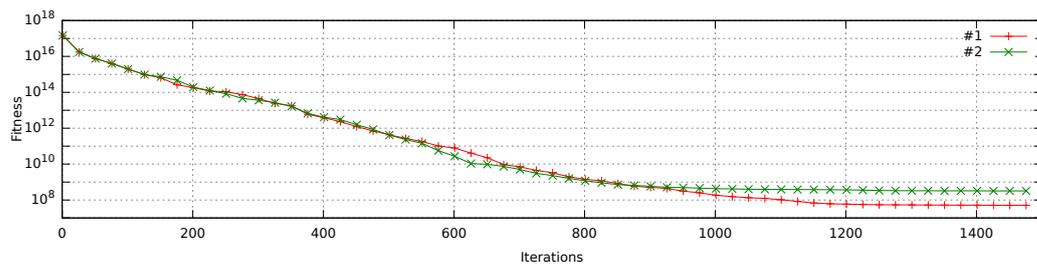


Fig. 10 The performances of two independent runs of the PSO algorithm with the Dynamic topology optimizing F8 function.

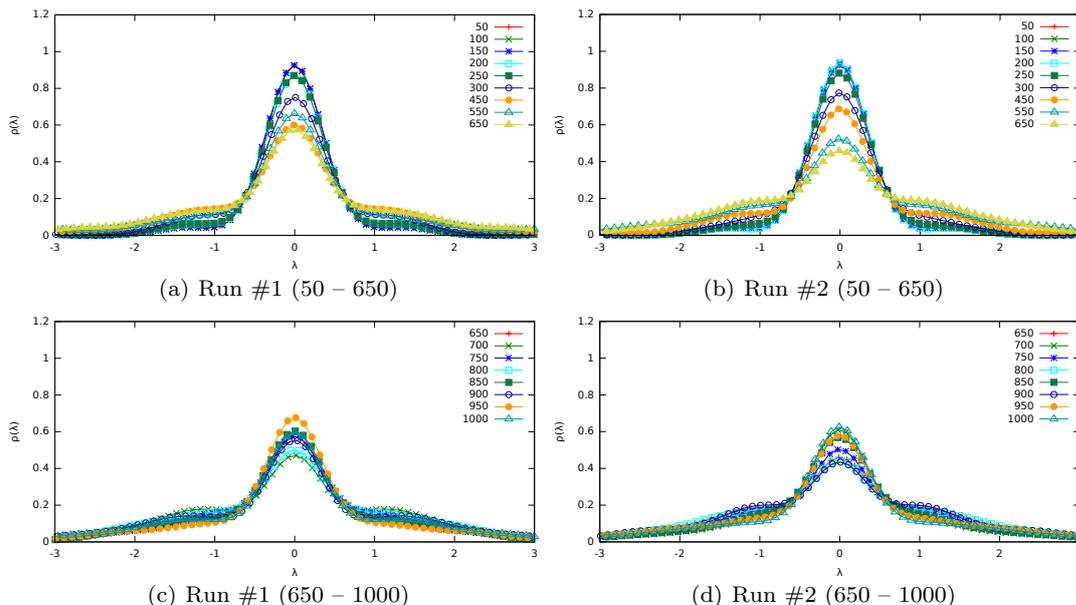


Fig. 11 Density spectrum of the Swarm Influence Graph from different iterations of the two algorithm executions with performance shown in Figure 10. Again, the curve for Run #2 is flattened out faster than the curve from Run #1.

evolution of the spectral density may be used to assess the influence of a parameter in the swarm performance.

4.4 Using the R -value

Two analysis of the swarm influence graph with the R -value were made. First, we compare the structure features through iterations of the information flow of the PSO using different topologies. Thereafter, we analyze the performance of the algorithm by its structural characteristics when the swarm has a Dynamic topology.

4.4.1 Static and dynamic topologies

Figure 12 presents the behavior of R -value of the influence graph as a function of the number of iterations for the three topologies considered in this paper. A low R -value means that λ_1 is not detached from the rest of the spectrum and it can be seen as a consequence of a periodical structure (Farkas et al 2001).

Because of the star-like behavior of influence graph for the Global topology, its R -value is constant and presents the value 1, since the extremal eigenvalues λ_1 and λ_N are opposites, and $\lambda_2 = 0$.

Both the Ring and the Dynamic topologies present small R -values, as they produce influence graphs that display very periodical structures. Again, the Dynamic topology presented a balanced behavior.

4.4.2 Dynamic topology

In Section 4.3.2, Figure 7 and Figure 10 are runs of the PSO algorithm with different performances. Here, we try to assess the performances by means of the information disseminated in the swarm using the R -value to examine if we can learn more about those runs.

Figure 13(a) depicts the evolution of the R -value through the iterations of the PSO when optimizing the

$F6$ function. As the plot is confusing to understand, we use the cumulative R -value, defined as follows:

$$R_{cum_i} = \frac{1}{i} \sum_{j=1}^i R_j, \quad (10)$$

where R_i is the R -value at iteration i .

The evolution of the cumulative R -value through iterations of the swarm influence graph of the PSO when optimizing the $F6$ function using the Dynamic topology is shown in Figure 13(b). As it is possible to see, the behaviors of the curves are very different. In the first iterations, the plots are similar, but at approximately iteration 400, the plots diverge. One can observe that the cumulative R -value is higher for Run #2 and presents a value that is more similar to the one presented by a Global topology, which represents a higher chance to get trapped in a local minima.

Figure 13(c) shows the behavior of the cumulative R -value through iterations of the swarm influence graph of the PSO when optimizing the $F8$ function. Again, the worst run (Run #2) presented higher values for the cumulative R -value.

5 Conclusions and Future Work

We proposed the analysis of the swarm behavior in Particle Swarm Optimizers based on the particles interactions. We defined a network of these interactions named the swarm influence graph and we assessed the swarm behavior by analyzing this graph. The novelty of our approach is that we analyze the fundamental aspect of the swarm intelligence: the particles communication. Due to this characteristic, our proposal allows richer analyses regarding the information flows within the swarm than the usual approaches. Moreover, the particles properties (*e.g.* the position of a particle, the velocity of a particle, etc) are not needed on this swarm analysis, thus our approach is not dependent of the problem dimension.

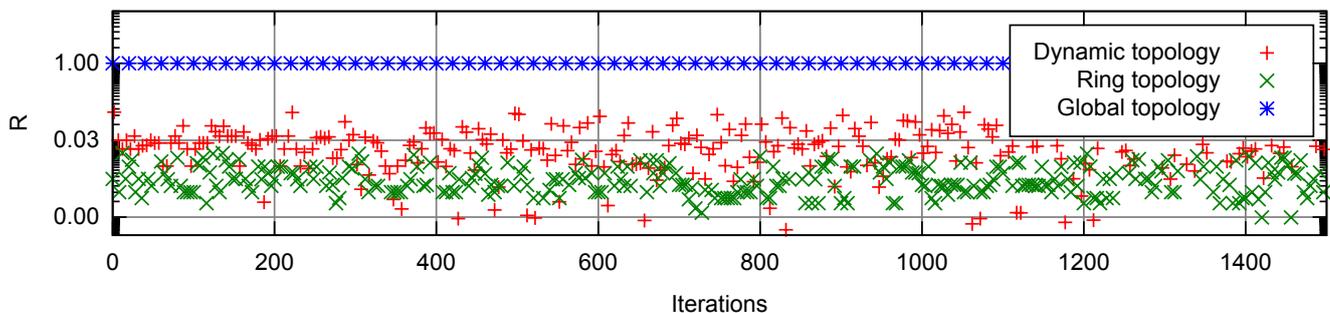
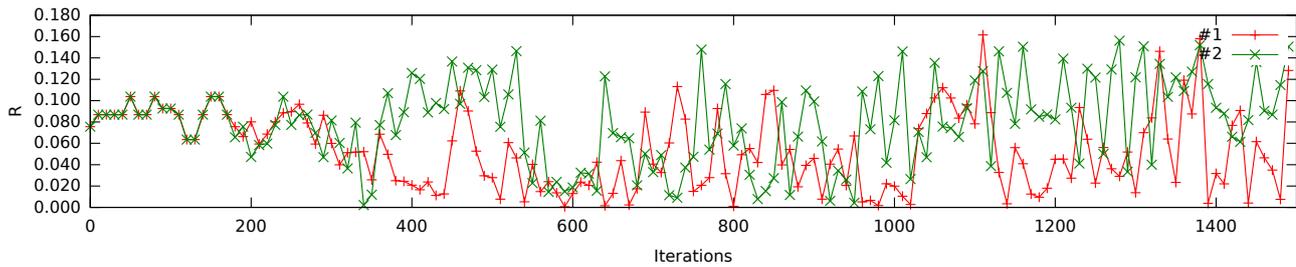
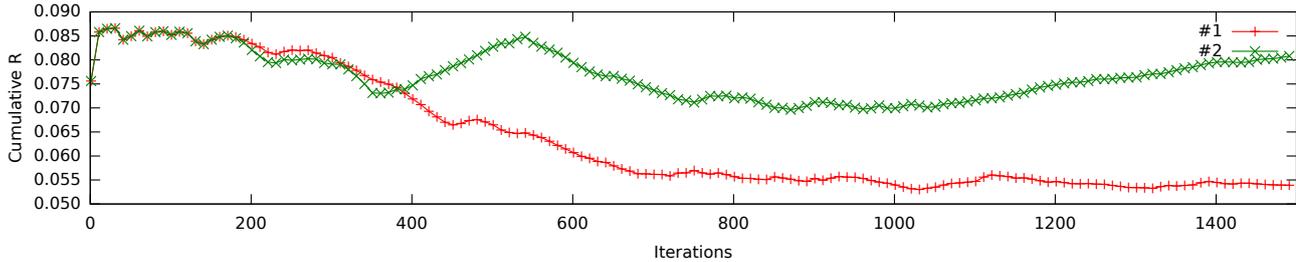


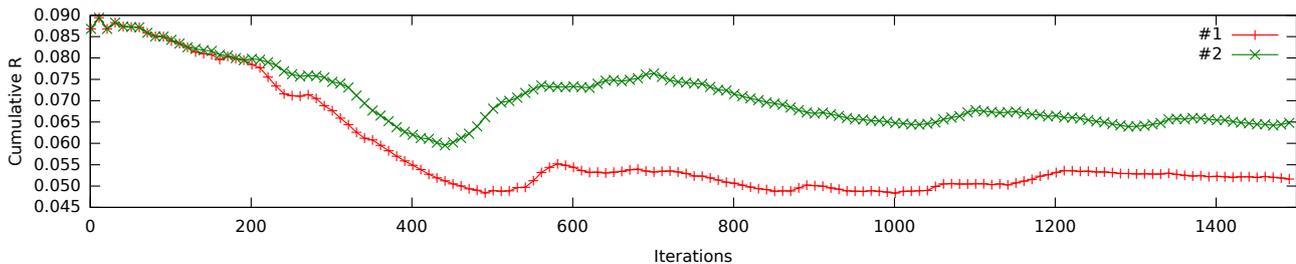
Fig. 12 R -value of the swarm influence graph along the iterations when the swarm has Dynamic, Ring and Global Topologies. The Dynamic topology leads to a balanced behavior between Ring and Global topologies.



(a) R -value of the Influence Graph of the swarm with Dynamic topology optimizing F6 function. Note that the plot is very confusing and hard to be analysed.



(b) Cumulative R -value of the Influence Graph of the swarm with Dynamic topology optimizing F6 function.



(c) Cumulative R -value of the Influence Graph of the swarm with Dynamic topology optimizing F8 function.

Fig. 13 Comparison of the R -value and cumulative R -value of the Influence Graph of the swarm with Dynamic topology. We show the non-cumulative only for the $F6$ function because the result for the $F6$ function is similarly confusing and not used in any analysis. The evolution of the cumulative R -value for runs #1 and #2 diverges at a certain iteration.

In this paper, we proposed the analysis of the swarm influence graph to assess the information flow in order to find the fingerprints of the swarm search behavior. The analyses made on the influence graph consider its number of components and its structure. More specifically, we used: the number of zero-valued eigenvalues in the Laplacian matrix to count the number of components in the graph; the spectral density to characterize the network; and the R -value to numerically characterize the network. The first can be used to assess the number of information flows within the swarm. The second provides a signature of the information flows, which can provide insights on the type of search performed by the swarm in the last iteration. The third one also returns information about the network structure, but is a single value that could be used to control the operation of the algorithm.

We showed that our approach can assess the different search behaviors of the swarm when different

topologies are used. Each topology used by the swarm does have different signatures in the swarm influence graph that leads the swarm to behave in distinct manners. We showed that the swarm with the Dynamic topology has a behavior that is between the two most used static approaches, the Ring and Global topologies. Moreover, regarding the Dynamic topology, the simulation results showed that the measures can be used to indicate a possible stagnation process within the swarm. Again, this can be used to trigger operators, for example, to generate diversity in the swarm.

Although we have presented results using measures from the network science field applied to the PSO algorithm, the main idea presented in this paper is not technique dependent. We believe this is a first attempt to show that this measures can be used to assess the performance and to assist the design of swarm intelligence algorithms. In fact, the analysis of the communication within the swarm can be done with other swarm intel-

ligence approaches (*e.g.* fish school (FSS), ant colony (ACO), etc.). In ACO, the analysis could be performed in the pheromone graph. In the FSS, the idea could be adapted to assess the memory matrix introduced in the dFSS approach.

As future works, we intend to use these measures to design high performance dynamic topologies for PSOs by assessing the information flow within the swarm. We also aim to develop variations of the swarm influence graph to recognize different aspects of the swarm communication. Furthermore, we want to perform analysis on the swarm performance of different swarm intelligence approaches as indicated above.

Acknowledgements We thank the two anonymous reviewers for their useful criticism and suggestions for improvement on our manuscript. Marcos Oliveira would like to thank the Science without Borders program (CAPES Foundation, Brazil) for financial support under grant 1032/13-5.

References

- Albert R, Barabasi AL (2002) Statistical mechanics of complex networks. *Reviews of Modern Physics* 74:47, URL doi:10.1103/RevModPhys.74.47
- Barabasi AL, Albert R (1999) Emergence of scaling in random networks. *Science* 286:509–512
- Bastos-Filho CJA, Lima-Neto FB, Lins AJCC, Nascimento AIS, Lima MP (2008) A novel search algorithm based on fish school behavior. 2008 IEEE International Conference on Systems, Man and Cybernetics pp 2646–2651
- Bratton D, Kennedy J (2007) Defining a standard for particle swarm optimization. *Swarm Intelligence Symposium, 2007 SIS 2007 IEEE* pp 120–127, DOI 10.1109/SIS.2007.368035
- Clerc M, Kennedy J (2002) The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *Evolutionary Computation, IEEE Transactions on* 6(1):58–73, DOI 10.1109/4235.985692
- Cvetković D, Rowlinson P, Simić S (2010) *An introduction to the theory of graph spectra*, 1st edn. Cambridge University Press, New York
- Dorigo M, DiCaro G (1999) Ant colony optimization: A new meta-heuristic. In: *Proceedings of the Congress on Evolutionary Computation, IEEE Press*, pp 1470–1477
- Dorogovtsev SN, Goltsev AV, Mendes JFF, Samukhin AN (2003) Spectra of complex networks. *Phys Rev E* 68:046,109, DOI 10.1103/PhysRevE.68.046109, URL http://link.aps.org/doi/10.1103/PhysRevE.68.046109
- Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp 39–43, DOI 10.1109/MHS.1995.494215, URL http://dx.doi.org/10.1109/MHS.1995.494215
- Eberhart R, Simpson P, Dobbins R (1996) *Computational intelligence PC tools*. Academic Press Professional, Inc.
- Engelbrecht AP (2007) *Computational Intelligence: An Introduction*. Wiley Publishing
- Farkas IJ, Derényi I, Barabási AL, Vicsek T (2001) Spectra of "real-world" graphs: beyond the semicircle law. *Phys Rev E Stat Nonlin Soft Matter Phys* 64(2 Pt 2), URL http://view.ncbi.nlm.nih.gov/pubmed/11497741
- Godoy A, Von Zuben F (2009) A complex neighborhood based particle swarm optimization. In: *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pp 720–727, DOI 10.1109/CEC.2009.4983016
- Janson S, Middendorf M (2005) A hierarchical particle swarm optimizer and its adaptive variant. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 35(6):1272–1282, DOI 10.1109/TSMCB.2005.850530
- Karaboga D (2005) *An idea based on honey bee swarm for numerical optimization*. Tech. rep., Erciyes University, Engineering Faculty, Computer Engineering Department
- Kennedy J (1997) The particle swarm: social adaptation of knowledge. In: *Evolutionary Computation, 1997., IEEE International Conference on, IEEE*, pp 303–308
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *IEEE International Conference on Neural Networks, vol 4*, pp 1942–1948 vol.4, DOI 10.1109/ICNN.1995.488968, URL http://dx.doi.org/10.1109/ICNN.1995.488968
- Kennedy J, Eberhart RC (2001) *Swarm Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA
- Kennedy J, Mendes R (2002) Population structure and particle swarm performance. In: *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, vol 2, pp 1671–1676, DOI 10.1109/CEC.2002.1004493
- Mendes R, Kennedy J, Neves J (2003) Watch thy neighbor or how the swarm can learn from its environment. In: *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, pp 88–94, DOI 10.1109/SIS.2003.1202252
- Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: simpler, maybe better. *Evolutionary Computation, IEEE Transactions on* 8(3):204–210, DOI 10.1109/TEVC.2004.826074
- Mo S, Zeng J, Tan Y (2010) Particle swarm optimization based on self-organizing topology driven by fitness. In: *Proceedings of the 2010 International Conference on Computational Aspects of Social Networks, IEEE Computer Society, Washington, DC, USA, CASO-N '10*, pp 23–26, DOI 10.1109/CASoN.2010.13, URL http://dx.doi.org/10.1109/CASoN.2010.13
- Oliveira M, Bastos-Filho CJA, Menezes R (2013) Using network science to define a dynamic communication topology for particle swarm optimizers. In: *Menezes R, Evsukoff A, González MC (eds) Complex Networks, Studies in Computational Intelligence, vol 424, Springer Berlin Heidelberg*, pp 39–47, DOI 10.1007/978-3-642-30287-9_5
- Peram T, Veeramachaneni K, Mohan C (2003) Fitness-distance-ratio based particle swarm optimization. In: *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, pp 174–181, DOI 10.1109/SIS.2003.1202264
- Pontes MR, Neto FBL, Bastos-Filho CJ (2011) Adaptive clan particle swarm optimization. In: *Swarm Intelligence (SIS), 2011 IEEE Symposium on, IEEE*, pp 1–6
- Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pp 69–73, DOI 10.1109/ICEC.1998.699146
- Suganthan P (1999) Particle swarm optimiser with neighbourhood operator. In: *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol 3, pp 3 vol. (xxxvii+2348), DOI 10.1109/CEC.1999.785514

- Tang K, Li X, Suganthan PN, Yang Z, Weise T (2010) Benchmark Functions for the CEC'2010 Special Session and Competition on Large-Scale Global Optimization. Tech. rep., University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL): Héfěi, Ānhuī, China, URL <http://goo.gl/kz5P7d>
- Wang YX, Xiang QL (2008) Particle swarms with dynamic ring topology. In: Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on, pp 419–423, DOI 10.1109/CEC.2008.4630831
- Zhan ZH, Zhang J, Li Y, Chung HH (2009) Adaptive particle swarm optimization. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 39(6):1362–1381
- Zhang J, Zhan Zh, Lin Y, Chen N, Gong Yj, Zhong Jh, Chung HS, Li Y, Shi Yh (2011) Evolutionary computation meets machine learning: A survey. Computational Intelligence Magazine, IEEE 6(4):68–75
- Zhou Z, Shi Y (2011) Inertia weight adaption in particle swarm optimization algorithm. In: Advances in Swarm Intelligence, Springer, pp 71–79