

Characterizing the Social Interactions in the Artificial Bee Colony Algorithm

Lydia Taw[†], Nishant Gurrupadi[‡], Mariana Macedo^{§*}, Marcos Oliveira[¶],
Diego Pinheiro^{||}, Carmelo Bastos-Filho^{††}, and Ronaldo Menezes[§]

[†]Department of Computer Science, George Fox University, USA

[‡]Department of Computer Science, University of Texas at Dallas, USA

[§]BioComplex Laboratory, Department of Computer Science, University of Exeter, UK

[¶]Computational Social Science, GESIS–Leibniz Institute for the Social Sciences, Germany

^{||}Department of Internal Medicine, University of California, Davis, USA

^{††}Polytechnic School of Pernambuco, University of Pernambuco, Brazil

Abstract—Computational swarm intelligence consists of multiple artificial simple agents exchanging information while exploring a search space. Despite a rich literature in the field, with works improving old approaches and proposing new ones, the mechanism by which complex behavior emerges in these systems is still not well understood. This literature gap hinders the researchers’ ability to deal with known problems in swarms intelligence such as premature convergence, and the balance of coordination and diversity among agents. Recent advances in the literature, however, have proposed to study these systems via the network that emerges from the social interactions within the swarm (i.e., the interaction network). In our work, we propose a definition of the interaction network for the Artificial Bee Colony (ABC) algorithm. With our approach, we captured striking idiosyncrasies of the algorithm. We uncovered the different patterns of social interactions that emerge from each type of bee, revealing the importance of the bees variations throughout the iterations of the algorithm. We found that ABC exhibits a dynamic information flow through the use of different bees but lacks continuous coordination between the agents.

Index Terms—Swarm Intelligence, Network Science, Social Interaction, Artificial Bee Colony

I. INTRODUCTION

Since its introduction, Computational Swarm Intelligence has proven to be a valuable approach to numerical optimization and in particular to finding solutions to problems involving high-dimensional search spaces [1]–[3]. Swarm-based algorithms are inspired by the collective behavior exhibited in nature by ants [4], bees [5], fireflies [6], and fish [7], to name a few. In swarm-intelligent techniques, multiple simple artificial reactive agents are released onto a limited search space to find the best possible positions [2], [3]. Though several algorithms share this same principle, the community is yet to explain the mechanisms that turn these algorithms intelligent [8].

Swarm-intelligent algorithms lack *explainability*. We still cannot explain why they work well and—maybe more

importantly—why they sometimes do not work [8]. The field lacks a unified classification of the plethora of algorithms proposed thus far as well as the understanding of their relationships and use cases [8]–[10]. Bonabeau et al. argued that a more systematic comparison of the existing swarm-based algorithms is crucial and could lead to a better understanding of why they work [9]. After almost two decades since this claim, there has not been much progress made towards a unified framework of swarm intelligence algorithms.

Bratton and Blackwell proposed a straightforward way to tackle the problem of understanding swarm algorithms [11]. They simplified a swarm-based algorithm, namely the Particle Swarm Optimization (PSO) [12], to study the parts of the system and understand their role in the complex swarm behavior. For this, they removed the randomizing factors from the equations of the algorithm. They demonstrated that by removing components of the algorithm and seeing how the removal affects the performance, we gain insights into what makes PSO an effective method. However, this was still far from showing why social behavior emerges from the simple rules.

A more concrete effort into understanding why swarm-based methods work uses an approach inspired by Network Science proposed by Oliveira et al. [8]. Their approach looks closer at the characteristics of the social interactions within a swarm. This framework can identify the *interaction diversity* throughout the iterations and trends towards premature convergence of PSO [13]–[17].

In this paper, we design and analyze the interaction network for the case of Artificial Bee Colony (ABC). We not only show that ABC can be modeled and evaluated by the interaction network approach but also demonstrate the potential of the framework in mapping swarm intelligence techniques to a common space.

The paper is structured as follows: Section II describes the ABC algorithm laying the foundation for understanding our approach. Section III defines what interaction networks are and why they are relevant to the field of swarm intelligence. In Section IV, we propose how to model the interaction network

*Corresponding author: mmacedo@biocomplexlab.org.

©20xx IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

for the ABC algorithm. Section V experimental setup applied to our ABC experiments leading to Section VI where we discuss the results and the insights obtained from analyzing the social interactions within ABC. Section VII concludes the paper with some final thoughts regarding directions for future research.

II. ARTIFICIAL BEE COLONY

Artificial Bee Colony (ABC) is an optimization algorithm inspired by the foraging behavior of honeybees; it consists of agents (i.e., bees) that change their roles over time [18]. Each role defines (a) how bees interact with other bees, and (b) how bees search for the solution of a problem. In the algorithm, a *food source* represents a possible solution to an optimization problem which is evaluated using a fitness function. The roles are defined as follows:

- (i) *Employed Bees* exploit food sources to which they are currently assigned (or “employed” at) and carry information about the profitability of these food sources. An employed bee interacts with a randomly-chosen employed bee to determine the direction of its movement.
- (ii) *Onlooker Bees* interact with employed bees and are more likely to interact with bees that are employed at the best-known regions of the search space.
- (iii) *Scout Bees* are an employed bee that abandoned its food source after exhaustively exploiting its food source. Scout bees do not interact with other bees and randomly explore the search space seeking for a new food source. Despite the lack of direct interaction, scouts find places that will be occupied by employed bees which in turn will interact with other bees. Hence they lead to future interactions.

Each cycle of the algorithm consists of bees changing roles, rotating from employed, to onlooker, and then scout. The pseudocode of ABC is described in Algorithm 1.

Algorithm 1 Pseudocode for the ABC algorithm.

Initialize the swarm on random positions in the search space
while the stopping criteria is not met **do**
 Place employed bees on the known food sources
 Place onlooker bees on the known food sources
 Send scouts to new food sources

Each employed bee performs a greedy search for a position near to its assigned food source and updates its food source to the newly found position if it has better fitness than the current food source. Equation 1 defines the new position \vec{v}_i of the bee i :

$$v_{i_k} = x_{i_k} + \phi_{i_k}(x_{i_k} - x_{j_k}) \quad (1)$$

where x_{i_k} is the previous position of the bee i at dimension k , ϕ_{i_k} is a randomly generated number from a uniform distribution in the interval $[-1, 1]$, and x_{j_k} is the position of a randomly chosen food source j at dimension k .

The onlooker bee is recruited to move towards the food source where another bee is depending on the quality associated with that food source, p_i , calculated using Equation 2:

$$p_i = \frac{f(i)}{\sum_{k=1}^N f(k)}, \quad (2)$$

where $f(i)$ is the fitness value of the food source i , and N represents the total number of food sources. That is, the bees select a food source using a roulette wheel mechanism that gives better food sources higher chances of being selected by the onlooker bee. Similar to the employed bees, the onlooker bees perform a greedy search in a nearby position within the search space.

Once a food source has been fully exploited (following a limiting parameter), the employed bee associated with that food source becomes a scout and selects a new source. The scout appears at a random new position and as the algorithm enters the next iteration, where the scout turns into an employed at the newly discovered food source.

III. INTERACTION NETWORK

Interactions are at the core of any swarm-based algorithm. The ability of the swarm to coordinate and adapt comes from the rules that define how agents interact among themselves and with the environment [2]. Depending on the inspiration of the algorithm, the interaction may be indirect or direct. Social interactions happen when individuals communicate, but more importantly, they also have the potential to change the individuals [2]. They are the reason for the emergence of complex behavior within the system, a consequence of agents being able to influence and be influenced by one or several other agents [1].

Oliveira et al. [8] introduced the perspective of analyzing swarm intelligence from the social interaction point of view by modeling them using a framework called *interaction network*. In the interaction network \mathbf{I} , nodes represent the agents, and the links between the nodes represent the presence (and intensity) of influence between the agents. This network enables the capture of social behavior within the swarm at different points in time during the algorithm execution.

The authors have used the interaction network to provide the means to understand different aspects of the Particle Swarm Optimization (PSO) such as the search mode and the impact of social dynamics on performance [13]–[17]. In the case of PSO, Oliveira et al. defined a network $\mathbf{I}'(t)$ at iteration t as an adjacency matrix where each element of the matrix is defined by the presence or the absence of influence between the agent i and j as follows:

$$\mathbf{I}'_{ij}(t) = \begin{cases} 1, & \text{if } i \text{ interacted with } j \text{ at the iteration } t, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Note that $\mathbf{I}'(t)$ tells us whether two artificial agents in a swarm interacted with each at a given iteration t and neglects the past of the swarm. To keep track of the history of information

exchanges, Oliveira et al. defines $\mathbf{I}^{t_w}(t)$ using a time window t_w as the following:

$$\mathbf{I}^{t_w}(t) = \sum_{t_i=t-t_w+1}^t \mathbf{I}'(t_i) \quad (4)$$

with $t \geq t_w \geq 1$. The time window t_w defines the frequency–recency balance in the analysis [14].

The weighted interaction network $\mathbf{I}^{t_w}(t)$ allows us to analyze the history of interactions of each agent in a swarm and determine if any particular agents have significantly more interactions and exerted more influence on the other agents. Moreover, we can analyze different time windows through time to identify the characteristics that define swarm-based techniques.

In order to capture the structure of the information flow within the swarm, Oliveira et al. also proposed to measure how quickly the interaction network can be destroyed by removing the edges from the network [8], [16]. As we remove edges based on their weight, isolated components start to emerge in the network [8]. Let $C(w)$ be the number of components in the network \mathbf{I}^{t_w} when we remove each edge $\mathbf{I}^{t_w}_{ij}$ that satisfies $\mathbf{I}^{t_w}_{ij} \leq w$. Note that the maximum value of $C(w)$ is the number of nodes in the network. The area under the curve of $C(w)$, namely A , gives us a measure of the pace that the components emerge and can be seen as a measure of diversity in the information flow. In order to account different networks (i.e., different time windows), Oliveira et al. defined the *Interaction Diversity* (ID) as:

$$\text{ID}(t) = 1 - \frac{1}{|S||T|} \sum_{t_w \in T} A_{t_w=t_w}(t), \quad (5)$$

where S is the size of the swarm, A_{t_w} is the area under the destruction curve of removal edges of \mathbf{I} with time window $t_w \in T$, and T is a set of time windows. With this definition, high ID values indicate exploration behavior in swarms, while low ID is an indicator of exploitation.

IV. AN INTERACTION NETWORK FOR ABC

Looking at ABC from a social interactions perspective, we describe how we model the interaction network for ABC in order to capture what is taking place between the various types of bees and consequently understand the dynamics of the algorithm. Because each type of bee has a different role in the algorithm, we model an interaction network for each type of bee as a layer in a multi-layer network and use the aggregated projection of the layers as a representation of the entire swarm. Figure 1 illustrates this multi-layer representation for ABC. The first layer represents the aggregation of all bee types (a projection of the multi-layer network). The second layer represents the behavior of employed bees who are interacting with each other randomly and with similar interaction strength. The third layer represents onlooker bees and the different strengths between the social interactions which lead to the presence of hubs (bees interacting with many bees) displayed by the size of nodes. The fourth layer represents that scouts are

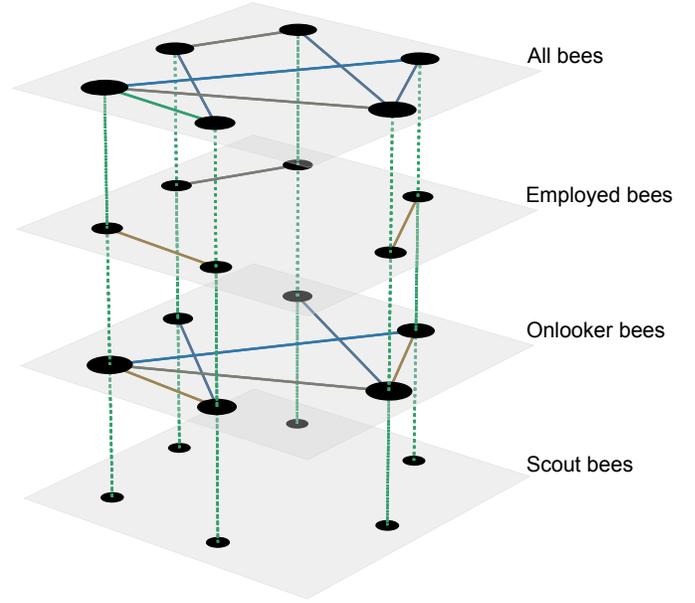


Figure 1: Multi-layer network capturing the social interactions in ABC. The nodes represent the bees with their respective degree strength shown by their size. The edges colors represent the strength of the link between two bees. The brown color is the weakest link, followed by blue and then green. We observe that employed bees display more random influences with similar strengths, onlooker bees have a more diverse social interaction network because some bees are more influential, and the scouts bees only influence themselves. The single layer projection of the multi-layer network of all networks is a simple sum of all edges from every pair of nodes.

only influencing themselves and not each other. A four-layer model allows capturing the movement and interaction for each type of bee at each iteration of the algorithm. There may be other alternatives to build the model, but this particular one provides a comprehensive way to analyze the abstraction of the core of the social interactions.

A. Interaction Network for the Employed Bees $\mathbf{I}_E^{t_w}$

In the employed stage, an edge can be drawn between bee i and j if bee i influenced bee j or vice-versa during a successful greedy search. It is important to note that we are representing interactions in such a way that an edge is only drawn if the bee actually moved, indicating the movement was successful.

$$\mathbf{I}'_{Eij}(t) = \begin{cases} 1, & \text{if } i \text{ moved using information from bee } j \\ & \text{at the iteration } t, \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

B. Interaction Network for the Onlooker Bees $\mathbf{I}_O^{t_w}$

In the onlooker stage, the bee is recruited to go to a food source and is once again influenced by random bees at that food source. Because bees are more likely to be recruited to promising locations, we would expect to see strong recurrent

influences between bees at those food sources leading to networks having hubs (i.e., highly-connected nodes).

$$\mathbf{I}'_{O_{ij}}(t) = \begin{cases} 1, & \text{if } i \text{ moved using information from bee } j \\ & \text{at the iteration } t, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

C. Interaction Network for the Scout Bees \mathbf{I}'_S

An employed bee turns into a scout when the greedy search performed at a food source has been unsuccessful for a predetermined number of trials. The bee jumps to an entirely new, random position on the search space. The scout stage represents the exploration of the algorithm. Excluding it from the interaction network would have resulted in an inaccurate representation of overall influence among the bees. When a bee changes its role to a scout, a self-loop link of influence can be drawn, because the bee is influencing its new position.

$$\mathbf{I}'_{S_{ij}}(t) = \begin{cases} 1, & \text{if } i \text{ moved using information from bee } j \\ & \text{at the iteration } t, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

It is worth noticing that in our current model, \mathbf{I}'_S is disconnected meaning that it does not contribute directly to the aggregated network. However, we still chose to represent it in the general model because others could choose to capture the indirect influence of scouts in the future. As we mentioned before, scouts lead to future interactions because their random jumps will cause employed and onlooker bee interactions. We discuss this in a little more detail in Section VII.

D. Aggregation of the Interaction Networks \mathbf{I}'_A

We use the same Eq. 3 and Eq. 4 developed by Oliveira et al., but, in our case, we capture each stage both separately (each layer), and together as a sum of the stages to form the interaction network representing the full run of the ABC algorithm.

$$\mathbf{I}'_A = \mathbf{I}'_E + \mathbf{I}'_O + \mathbf{I}'_S \quad (9)$$

Note that this may not be the only way of building an interaction network for ABC, but we believe that our model demonstrates the network can capture some idiosyncrasies of ABC which suffices for understanding the algorithm.

V. EXPERIMENTAL SETUP

We selected the Rastrigin function to analyze and understand the behavior of bees in ABC. This multimodal function is widely applied as a benchmark of optimization algorithms. Because of its many peaks, it is a hard function to solve with a high number of dimensions. The optimal solution for the Rastrigin function is at $\vec{x} = \vec{0}$ with $F(\vec{x}) = 0$. To determine whether the complexity of Rastrigin impacts the behavior bees in ABC, we tested it with two different configuration setups: using 100 and 1,000 dimensions. It is important to highlight that the study in this unique benchmark does not allow to

analyze the behavior of the swarm in all situations, but enable to validate the proposal as a tool to analyze artificial bee colony algorithms.

All bees were spread uniformly in the search space at the beginning of each of the 30 executions. We used 50 bees in our experiments. Initially, we run the experiments with 200,000 evaluations per execution. However, it became evident that more time was necessary in order for patterns to be detected. Specifically, we needed to detect if the behavior found at the beginning of the execution would happen again later in the execution. Consequently, the number of evaluations per execution was increased to 1,000,000 resulting in around 9,950 iterations per execution. Because the scouting stage results in a non-deterministic number of evaluations, the exact number of iterations varies slightly between executions.

In our results, we aim to show that employed and onlooker bees display different behaviors because onlookers are more likely to go to better regions. Also, we want to show that each bee has the opportunity to influence the swarm at different points in the execution, hence the necessity for having all three mechanisms modeled. To demonstrate this necessity, we evaluate the temporal dynamic behavior of each network. We also show that the aggregation of all bees into one network hides the peculiarities of each type of bee, but reveals the overall effects of the bees working together in each iteration.

VI. RESULTS

We first look at the evolution of the fitness value through the interactions. Figure 2 shows the executions for 100 and 1,000 dimensions. We noticed that the evolution of fitness varied among the executions. All the executions converged prematurely as the best solution would be zero. Some simulations converged before 3,000 iterations, others would continue improving through more iterations. The influence between bees dynamically changes over time due to the nature of the algorithm, so we cannot observe a complete stagnation of the swarm on the search space even though we notice a fitness stagnation. The scout stage allows for new areas to be explored, but ABC does not have mechanisms to prevent bees from exploiting areas similar to their old food sources nor remembers multiple good regions explored in the past.

Figure 3 depicts the degree distribution of the interaction network at the last iteration. If the bees were interacting randomly, we would see a distribution with a well-behaved expected average degree (normal distribution). Instead, we observe a slow decay that allows bees to attain a high weighted degree—these bees are influential in the swarm. Moreover, the number of dimensions does not change the behavior expected from the interaction network, so we detect similar characteristics from both experiments.

Considering the communication flow in the interaction networks, we analyzed the evolution of Interaction Diversity (ID) from 100 and 1,000 dimensions perspectives in Figures 4 and 5 using a cumulative time window of one iteration. Looking at both images, we notice that the difficulty of the function being optimized does not impact the diversity of the swarm,

so we conclude that the bees interact similarly independent of the number of dimensions.

The interaction diversity of ABC indicates that exploitation is the main focus of the algorithm, and the little exploration from the lack of the coordination perspective can corroborate ABC being stagnated from the fitness perspective. Even though the scouts bring to ABC new search space areas, they do not have the possibility of high exploration because the scouts have no memory to avoid previously-explored locations or strategies to better spread on the search space.

In the first 100 or so iterations, there is a sharp decrease in Interaction Diversity. This indicates that the swarm is exploiting more over time. Even though the scouts affect the search space, interaction diversity is not impacted. Consequently, the scouts seem to be not relevant to the interaction network point of view.

If we calculate the Interaction Diversity using a time window of 5 iterations, we identify that the diversity does not

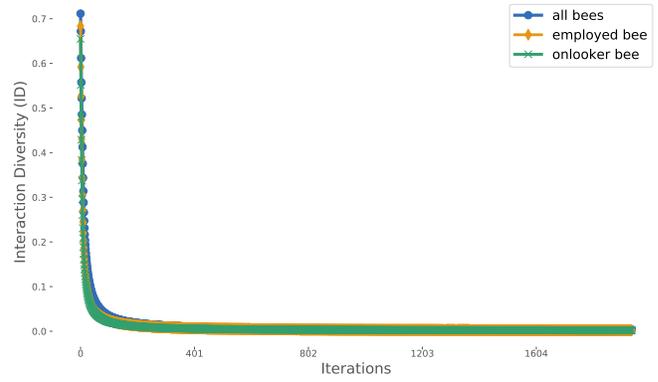


Figure 4: Interaction Diversity evolution through the iterations performed by ABC on the Rastrigin function (100 dimensions) shows a smooth decrease in the first 1,000 iterations indicating steady exploitation.

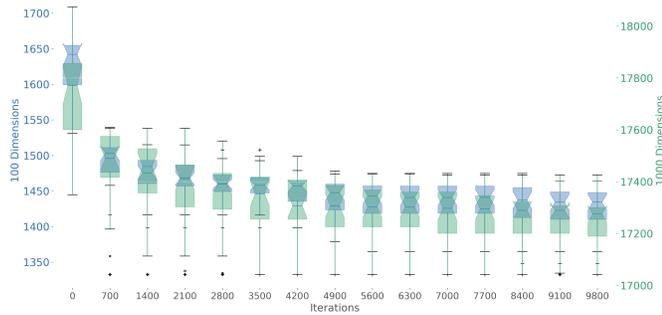


Figure 2: Evolution of the fitness as a function of the number of iterations with around 9500 iterations. The double y-axis represents the value of the fitness at each execution time. This compares the behavior of two experiments run with two different configurations. The blue indicates the experiment run with 100 dimensions, and the green indicates 1000 dimensions.

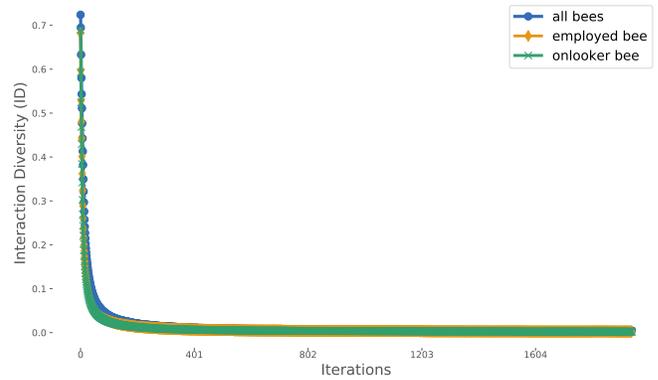


Figure 5: Interaction Diversity evolution through iterations performed by ABC on the Rastrigin function (1,000 dimensions). We observe a behavior similar to the case with 100 dimensions.

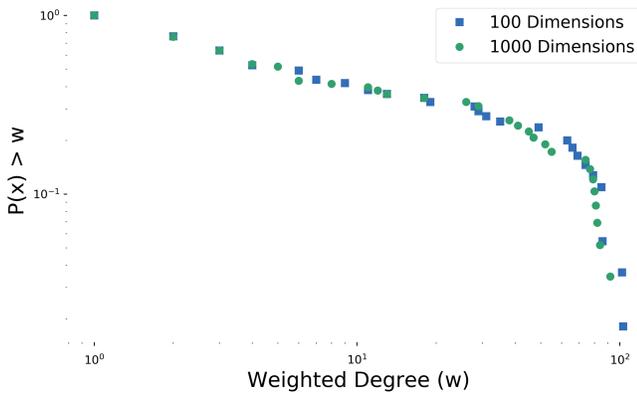


Figure 3: Complementary Cumulative Distribution Function (CCDF) of the final interaction network of 100 and 1,000 dimensions. Note that there are very few nodes with a high weighted degree (hubs) Note that the plot is on a log-log scale..

change much over time because the rules also do not change; we only identify a tiny difference from the values (approx 0.025). The creation of adaptive versions of ABC could be the solution to recover the behavior found at the beginning of the execution and to improve the lack of exploration found at later stages of the execution. However, as ABC does not utilize previous information as much as PSO [12] and FSS [19], the convergence is the real limitation of the algorithm.

Figure 6 shows the number of nodes in the giant component of interaction graph from the best execution using 1,000 dimensions; it also shows how it changes depending on the size of the time window used. The giant component represents the biggest sub-swarm with the most number of nodes concerning the other components. Presumably, the number of components is the same as the number of bees in the swarm upon initialization as no interactions have occurred yet. Throughout the iterations, as the bees start interacting with each other, the number of components decreases as new connections are

established and lost. Similarly, the number of nodes in the giant component increases as the time window increases as a result of the cumulative interactions among the bees.

When the time windows are less than 10, we observe that the ABC works on separate sub-swarms. This excess of diversity causes a lack of coordination among the bees, so even though they help each other to move around some food source, they also do not influence each other in a coordinated way. If we analyze what is happening on the search space, they look like they are coordinated in some food sources, but what they are doing is being selfish regarding the neighbors.

At the same time window point, we see that despite the existence of several components, the giant component seems to be large compared to the other components. The scouts provoke the presence of more sub-swarms. However, the onlookers display a high impact through time enhancing the relationship between some particular bees, and the employed bees make the bees widely connected because they rarely interact with the same bee in the same time window. They are the mechanism by which the swarm tries to coordinate and make bees go to better regions. Without the onlooker role, the convergence of the algorithm would be impacted because they would lack coordination leading to a near random behavior.

Figure 7 represents the value of sum of all edge weights and number of edges while increasing the time window from the aggregated network in ABC. We display both because they provide a better idea of the number of edges and the total influence exerted by them. The influence of bees (edge weight) and the number of connections (number of edges) are small when analyzing small time windows. Moreover, the increase of the standard deviation shows that the influence fluctuates more throughout the iterations when analyzing bigger time windows. One thing to point out is that even though the greedy movement constrains the bees to move only when successful, the influence by time window remains generally small. Several bees move and are unsuccessful which demonstrates the waste of computational time. In the first 10 iterations, the bees easily influence each other and move successfully, but after about 10 iterations the movement is less likely to take place.

In Figure 8, we show the interaction network (as an adjacency matrix) of each bee type. The interaction network of the employed bees is similar to the onlooker because they differ only in strength and not in the pattern of social interactions. The scout interaction network has only diagonal values, which is precisely what was expected due to the nature of the scouting stage. The cumulative networks are scale-free, which means the influence is spread dynamically among the bees, but some bees are hubs (influencers in the swarm). To demonstrate the difference between employed and onlookers, we provide Figure 9; we diminished the color scale, allowing us to observe the intensity of influence between onlookers and employed. The employed stage displays green lines which indicate the presence of hubs in the network because some employed bees start in good regions, while others start in bad regions. The onlooker stage displays green, yellow, and red which represents a higher level of influence than the employed bees.

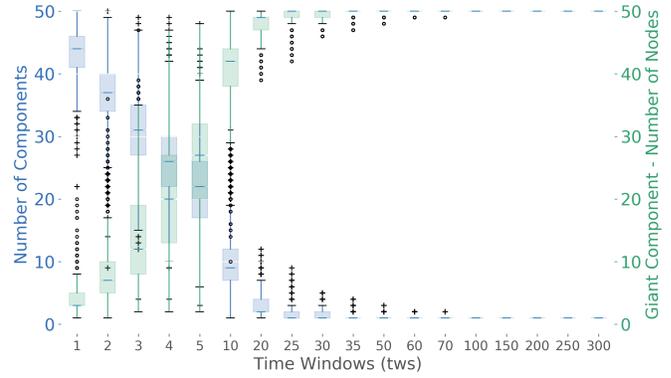


Figure 6: Number of components in the interaction network and the number of nodes in the giant component with increasing sizes of sliding time windows (1,000 dimensions).

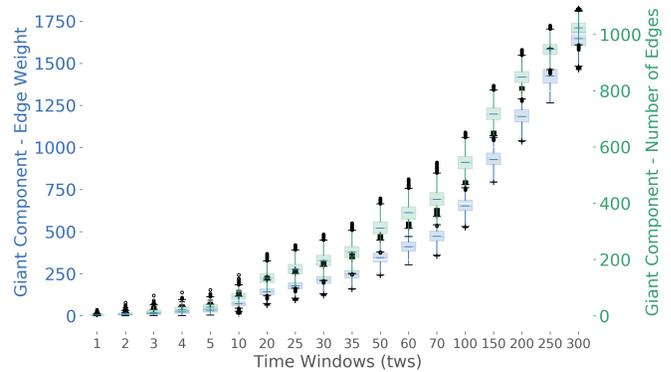


Figure 7: Number of edges and sum of the edge weights within the giant component as we vary the size of the time windows (1,000 dimensions).

This is because onlooker bees have a higher probability of going towards good regions due to the nature of the algorithm allowing them to have more movement. Furthermore, the bees are more likely to choose the same good locations. These heat maps allow us to differentiate between the nuances of influence between employed and onlooker. We can also see that some bees are hubs in both stages, and the onlooker stage is more effective in the greedy search than the employed stage. This is expected because the best food sources always attract more onlooker bees.

Looking at the time evolution of the employed and onlooker bees in Figure 10, we observe that a majority of the influence happens at the beginning of the simulation rather than the end. As more iterations happen, it becomes harder for bees to improve and move successfully. Despite having scouts who are randomly assigned to new food sources, the swarm does not seem to have strong enough coordination that allows them to find better regions and avoid worse regions. The algorithm does not provide a strategy for the bees to make wise decisions about where scouts can appear. Onlooker bees have a slightly

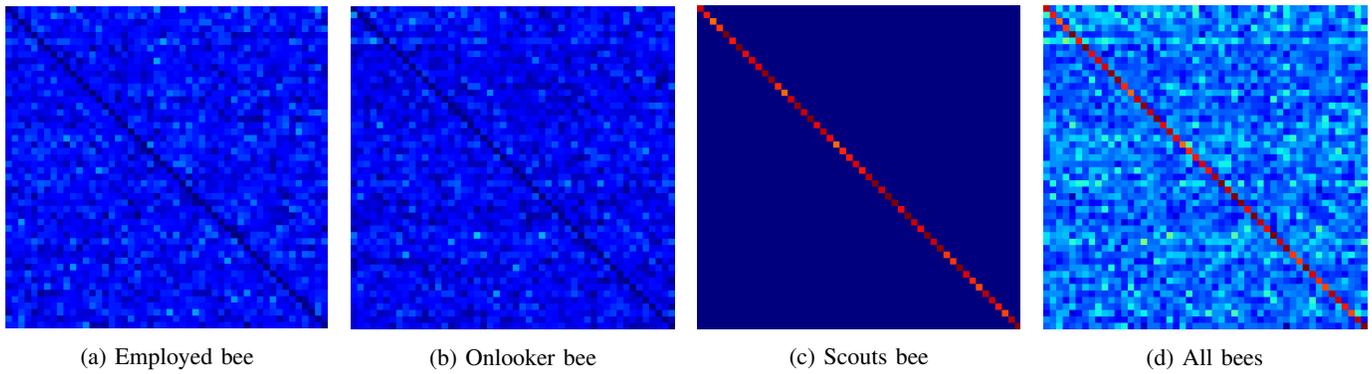


Figure 8: Interaction Network of the middle of iterations for each bee. The degree of influence is indicated on a color scale of dark blue to dark red, with the dark blue indicating a low level of interaction and dark red a high level of interaction. As expected, employed and onlooker bees never influence themselves, and the scout bees only influence themselves.

more controlled way of exploiting a food source. Because previously successful food sources most influence them, there is a small amount of coordination between them. However, even though they are attracted to the same place, they still do not have enough time to converge in a food source.

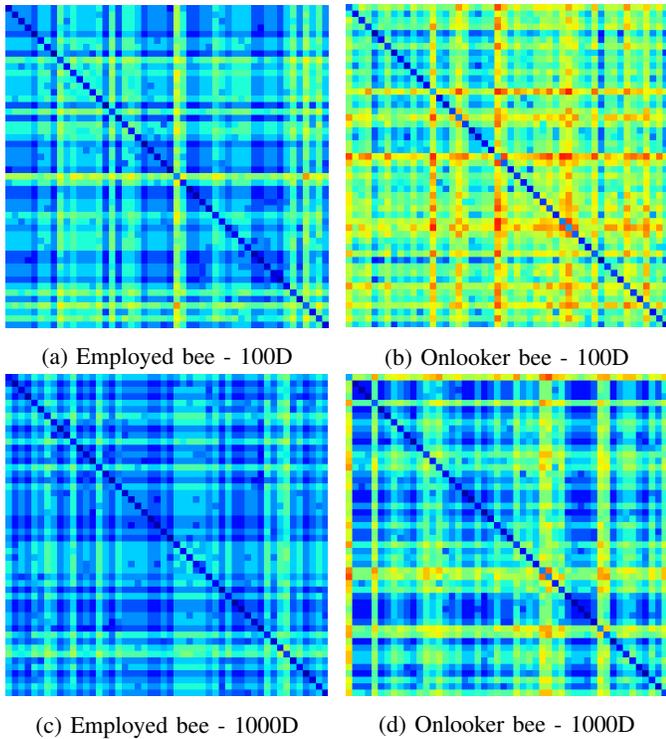


Figure 9: Interaction network from the employed and onlooker bees for 100 and 1,000 dimensions. The color scheme represents the strength of influence between bees, from blue, the weakest influence until red, the strongest influence.

VII. CONCLUSIONS

The understanding of swarm-based algorithms is still challenging; few have attempted to explain *why* they work. In

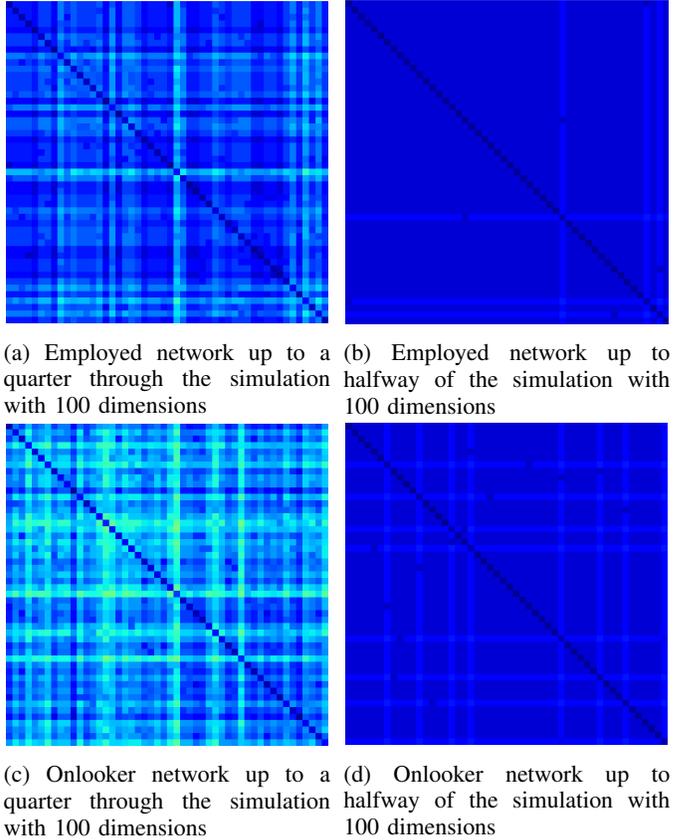


Figure 10: Influence between bees over time. The color scheme represents the strength of influence between bees from blue as the weakest influence to green as the most substantial influence. Earlier iterations have stronger influence activity, and influence becomes less intense as iterations happen. Onlooker bees display higher levels and more concentrated areas of influence. The employed bees show a little more random influence, which is what we would expect to see due to the nature of the algorithm.

this paper, we contribute to the area with new insights about the general usefulness of the interaction network framework to evaluate the social interactions of the swarm. We expanded upon Oliveira et al.'s [17] experiment by taking their network-model approach to the PSO and applying it to the ABC. We demonstrate that ABC can be modeled and analyzed by this framework as a multi-layer network in which the aggregated version captures the general behavior within the swarm. We also present that each type of bee or stage provides different influences over time, and the absence of a layer prevents a complete understanding of the algorithm.

Interestingly, ABC displays a lack of coordination within the swarm which is not present on several configurations of other swarm-based techniques such as PSO [12] and FSS [19]. The non-cooperation is dictated by the excess of dynamic influence through the network supplying diversity on the search space and also on the social behavior.

Moreover, we can identify that employed and onlooker bees influence the swarm at different levels, but they have similar patterns of influence. The employed bees contribute to the network with a random influence between the bees which adds diversity in the influence patterns. The onlooker bees reveal a higher strength of influence which is expected by the higher probability of successful movements arising from their transition rules. At first glance, scouts may seem unimportant to the process because they generally affect themselves in the multi-layer network. However, without them, the entire ABC algorithm would fail due to lack of diversity. The network that emerges from the two layers (employed and onlooker) is insufficient for ABC to discover other areas of the search space. Note that from Algorithm 1 that scouts will assume other roles in the algorithm, so the discovery of new locations is fundamental to the success of ABC. This recognition of different bee patterns expresses the richness of our modeling.

Lastly, this paper does not argue that our model is the only way that social interactions may be modeled in the ABC. Even previous works on the modeling of PSO do not make such a claim. There may well be other possible representations, but the point of our work is to show that an interaction network modeling is also useful in the case of ABC.

Future works may attempt to discover alternate ways of modeling the ABC as our goal in this paper was not provide the best way to model the interaction network for ABC, instead, show that using a network-based model can be used to explain what is happening within the algorithm. For instance, one alternative approach to the interaction network could use the food sources as nodes in a bipartite network composed of bees and food sources. Bees move to food sources and interactions could be captured by bees that find themselves in the same location within a time window. Such an approach could highlight the influence of scouts a little more directly. Another future work is to analyze how the dynamics of the swarm is affected in constrained optimization problems.

We believe that more work is necessary for the use of interaction network modeling in ABC and other swarm algorithms and that such an approach can unveil similarities between

many of the swarm approaches currently in the literature.

ACKNOWLEDGMENT

The authors acknowledge support from National Science Foundation (NSF) grant No. 1560345 and Florida Institute of Technology for providing the computing facilities. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. This work used the Extreme Science and Engineering Discovery Environment (XSEDE) Bridges at the Pittsburgh Supercomputing Center through allocation TG-IRII80008, which is supported by National Science Foundation grant number ACI-1548562 [20].

REFERENCES

- [1] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*. Oxford university press, 1999, no. 1.
- [2] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. Morgan Kaufmann Publishers Inc., 2001.
- [3] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, 2006.
- [4] M. Dorigo and T. Stützle, *Ant Colony Optimization*. The MIT Press, 2004.
- [5] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (abc) algorithm," *Applied soft computing*, vol. 8, no. 1, pp. 687–697, 2008.
- [6] X.-S. Yang, "Firefly algorithm, lévy flights and global optimization," in *Research and Development in Intelligent Systems XXVI*, 2009, pp. 209–218.
- [7] C. J. Bastos Filho, F. B. de Lima Neto, A. J. Lins, A. I. Nascimento, and M. P. Lima, "A novel search algorithm based on fish school behavior," in *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*. IEEE, 2008, pp. 2646–2651.
- [8] M. Oliveira, D. Pinheiro, M. Macedo, C. Bastos-Filho, and R. Menezes, "Unveiling swarm intelligence with network science—the metaphor explained," *arXiv preprint arXiv:1811.03539*, 2018.
- [9] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, no. 6791, p. 39, 2000.
- [10] K. Sörensen, "Metaheuristics—the metaphor exposed," *International Transactions in Operational Research*, vol. 22, no. 1, pp. 3–18, 2013.
- [11] D. Bratton and T. Blackwell, "Understanding particle swarms through simplification: a study of recombinant pso," in *Proceedings of the 9th annual conference companion on Genetic and evolutionary computation*. ACM, 2007, pp. 2621–2628.
- [12] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*. IEEE, 1995, pp. 39–43.
- [13] M. Oliveira, C. J. A. Bastos-Filho, and R. Menezes, "Assessing particle swarm optimizers using network science metrics," in *Complex Networks IV*, ser. Studies in Computational Intelligence, 2013, vol. 476, pp. 173–184.
- [14] M. Oliveira, C. J. Bastos-Filho, and R. Menezes, "Towards a network-based approach to analyze particle swarm optimizers," in *Swarm Intelligence (SIS), 2014 IEEE Symposium on*. IEEE, 2014, pp. 1–8.
- [15] M. Oliveira, C. J. A. Bastos-Filho, and R. Menezes, "Using network science to assess particle swarm optimizers," *Social Network Analysis and Mining*, vol. 5, no. 1, pp. 1–13, 2015.
- [16] M. Oliveira, D. Pinheiro, B. Andrade, C. Bastos-Filho, and R. Menezes, "Communication diversity in particle swarm optimizers," in *International Conference on Swarm Intelligence*. Springer, 2016, pp. 77–88.
- [17] M. Oliveira, D. Pinheiro, M. Macedo, C. Bastos-Filho, and R. Menezes, "Better exploration-exploitation pace, better swarm: Examining the social interactions," in *Computational Intelligence (LA-CCI), 2017 IEEE Latin American Conference on*. IEEE, 2017, pp. 1–6.
- [18] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, Tech. Rep., 2005.

- [19] C. J. Bastos Filho, F. B. de Lima Neto, A. J. Lins, A. I. Nascimento, and M. P. Lima, "Fish school search," in *Nature-inspired algorithms for optimisation*. Springer, 2009, pp. 261–277.
- [20] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr, "Xsede: Accelerating scientific discovery," *Computing in Science and Engineering*, vol. 16, no. 5, pp. 62–74, Sept.-Oct. 2014.