BSODCS: Bee Swarm Optimization for Detecting Community Structure

Narimene Dakiche[a]
Laboratoire des Methodes de Conception de Systèmes (LMCS), Ecole Nationale Supérieure d'Informatique (ESI), BP 68M - 16270 Oued Smar, Alger, Algeria

Keywords: Social Networks, Community Detection, Bee Swam Optimization, Modularity.

Abstract: This paper presents, **BSODCS**, a Bee Swarm Optimization for detecting community structure within networks. It employs artificial bees to explore a search space and construct solutions for community detection. To accommodate the specific features of networks, we adopt a locus-based adjacency encoding scheme. Each bee makes decisions regarding its neighboring solutions and shares information through a dance. To explore the neighborhood of each bee, we use Pearson's correlation as the heuristic information. The modularity of the bees' solutions serves as a metric for evaluating their quality. The algorithm is tested on well-known real-world networks, and the experimental findings demonstrate that BSODCS outperforms other existing swarm-based methods, delivering higher-quality results.

1 INTRODUCTION

Modern network science has made significant progress in modeling complex real-world systems. One key characteristic of these networks is the presence of community structure. Over the past few years, numerous algorithms for community detection have been proposed to uncover the structural properties and collective behaviors in networks (Bedi and Sharma, 2016).

Community detection refers to the procedure of dividing a network into groups of interacting nodes depending upon their structural properties (Yang et al., 2013). It has been proven to be an optimization problem, considered NP-Hard, which involves maximizing various interesting objective functions (Brandes et al., 2007).

Modularity (Newman and Girvan, 2004) is by far the most used and best known quality function for measuring the quality of a partition of a network, even thought it presents a resolution limit problem in certain realistic situations (Fortunato and Barthélemy, 2007). Consequently, several community detection algorithms are proposed to optimize it.

Swarm Intelligence is the field of studying and designing well-organized computational intelligent interactive solutions, in which complex problems are being solved by using the behavior of living swarms such as birds, reptiles, fish, and ants (Nayyar et al., 2018). Many swarm based optimization algorithms have been adopted to tackle the community detection problem.

Cai et al. (2014) introduced a discrete Particle Swarm Optimization (PSO) algorithm specifically designed for detecting communities in signed networks. The same corresponding author further extends this research in (Cai et al., 2015), focusing on the clustering of large-scale social networks.

Ant Colony Optimization (ACO) has gained prominence as an effective global search meta-heuristic for graph-related problems. Various adaptations of the ACO algorithm have been proposed for community detection. Noteworthy instances of these adaptations can be found in the works cited as (Chen et al., 2012) and (He et al., 2011).

In recent times, researchers have begun to investigate the applicability of contemporary nature-inspired meta-heuristics for community detection. One of these methods is Bat algorithm (BA), as demonstrated in studies such as (Hassan et al., 2015) and (Song et al., 2016). These works shed light on the effectiveness of BA in tackling the community detection problem in graphs.

Another notable adaptation for community detection problems is the utilization of Firefly Algorithm (FA). Jaradat et al. (2018) present an interesting study where a FA-based solver outperforms other bio-inspired solvers, such as Genetic Algorithm (GA) and Ant Colony Optimization (ACO), when applied to a

[a] https://orcid.org/0000-0003-2371-616X

Dakiche, N.
BSODCS: Bee Swarm Optimization for Detecting Community Structure.
DOI: 10.5220/0012209300003584
In *Proceedings of the 19th International Conference on Web Information Systems and Technologies (WEBIST 2023)*, pages 422-428
ISBN: 978-989-758-672-9; ISSN: 2184-3252

small set of real-life networks.

Artificial Bee Colony (ABC) is another contemporary bio-inspired solver that has been successfully applied to community detection problem. In (Hafez et al., 2014), an ABC based algorithm is introduced, which autonomously determines the optimal number of partitions in the network. Moreover, (Dakiche et al., 2022) is an additional work that demonstrates the effectiveness of ABC for community detection challenges and offers insights into its potential applications in this domain.

Over the past decade, numerous algorithms inspired by the behavior of bees have been proposed and applied to various problems (Karaboga and Akay, 2009). One such algorithm is Bee Swarm Optimization (BSO), introduced as a metaheuristic by Drias et al. (2005), which draws inspiration from the foraging behavior of real bees.

BSO is an effective approach that has demonstrated promising results in several issues, such as satisfiability (Djeffal and Drias, 2013), association rules (Djenouri et al., 2013), clustering (Sadeg and Drias, 2007), and feature selection (Sadeg et al., 2015). Due to its effectiveness for those problems, we are interested in investigating its capabilities for community detection problem.

To the best of our knowledge, only one study purported at proposing an algorithm based on BSO for community detection problem (Belkhiri et al., 2017). In their algorithm, the authors take modularity Q as objective function and choose a string based schema to represent the solutions. No explicit local search was adopted to visit the solutions' neighborhood except for a flipping algorithm to determine the search area.

In our study, we propose to adapt the BSO algorithm for community detection problem by using the locus-based adjacency representation to record each solution representing a community structure, which has the advantage of enabling the algorithm to detect communities' number automatically, contrarily to the string based representation. Moreover, while exploring each bee's neighborhood, we propose to use the network knowledge to select the most appropriate neighbor. For that, we employ Pearson's correlation, which is a similarity measure between nodes. The algorithm, we call BSODCS (BSO for Detecting Community Structure), reaches better modularities compared to the first-proposed BSOCD (Belkhiri et al., 2017) and other swarm based algorithms from the literature.

The rest of the paper is organized as follows: In sections 2, the commuity detection as an optimization problem is described. In section 3, the general algorithm of BSO is presented and its application for community detection is described. Section 4 shows the results of BSODCS and compares them to the results of other methods before we conclude with final remarks and directions for future works in section 5.

2 COMMUNITY DETECTION PROBLEM

A complex network can be represented as a graph $G = (V,E)$, where V denotes the set of nodes and E denotes the set of edges. The graph consists of $N = |V|$ nodes and $m = |E|$ edges. To describe the network, an adjacency matrix A of size $N \times N$ can be used. Each element A_{ij} in the matrix corresponds to the presence ($A_{ij} = 1$) or absence ($A_{ij} = 0$) of an edge between the i^{th} and j^{th} nodes. The final community structure $P = \{C_1,C_2,...,C_k\}$ of the network is defined, where each $C_l (l = 1,2,...,k)$ is a subset of V, and k represents the total number of communities.

Community detection as an optimization problem requires the use of an objective function to guide the search for optimal solutions. The objective function acts as the "steering wheel" in the process, leading to favorable solutions. The most widely used and well-known quality function for evaluating the quality of a network partition is modularity Q (Newman and Girvan, 2004). In the context of a graph G with n nodes and m edges, modularity is computed in the following manner:

$$Q = \frac{1}{2m}\Sigma_{ij}^{n}\left(A_{ij} - \frac{d_i \times d_j}{2m}\right)\delta(C_i,C_j) \tag{1}$$

where A is the corresponding adjacency matrix, $d_i(d_j)$ denotes the degree of node $i(j)$, $C_i(C_j)$ is the community to which node $i(j)$ belongs to. $\delta(C_i,C_j) = 1$, if $C_i = C_j$, otherwise, 0. A high modularity indicates that there is a dense concentration of connections within nodes of the same community, while connections between nodes from different communities are sparse (Fortunato, 2010).

3 PROPOSED BSODCS ALGORITHM

BSO, inspired by the cooperative behavior of bees, is a search process that uses a population of artificial bees to solve optimization problems. It begins with a bee called BeeInit, which generates an initial solution as the reference one (RefSolultion). Then, additional solutions are generated from RefSolultion, using a flipping strategy to form the search space. Each solution is assigned to a bee, serving as a starting point for exploring its neighborhood. After producing their respective solutions, the bees communicate and share their best solution through the Dance table.

In order to maintain a balance between exploitation and exploration, BSO uses intensification and diversification mechanisms. Intensification improves the best global solution by selecting from the dance table, while diversification explores new areas. The choice is based on progress, and the selected solution becomes the new RefSolultion for the next iteration.

Algorithm 1: BSO for detecting community structure.

```
Data: A network G = (V,E)
Result: Community structure of the network C = {C1,C2,...,Ck}
Initialize the algorithm parameters: N (number of bees), itermax
  (maximum number of iterations), flip (search space creation
  parameter), and limit (abandonment criterion);
// Initial reference solution, see subsection 3.2;
RefSolution ← GenerateInitialSolution() ;
Set TabuList = newlist();
Set iter = 0;
while iter < itermax do
    // Save RefSolutio in Tabu list;
    TabuList ← TabuList + RefSolutio;
    // Determine search space, see subsection 3.3;
    for i ← 1 to N do
        ApplyFlippingStrategies;
    end
    Set nbChances = limit;
    // Neighborhood exploration (N), see subsection 3.4;
    for i ← 1 to N do
        NeighborBee ← GenerateNeighborBee(Bee[i]);
        // Save best solution in Dance table;
        DanceTable ← DanceTable + BestSol(Bee[i],NeighborEBee)
    end
    // Select the new RefSolution , see subsection 3.5;
    if
      BestModularity(DanceTable) > BestModularity(TabuList)
      then
        RefSolution ← BestModularity(DanceTable);
        nbChances ← limit;
    else
        if nbChances > 0 then
            nbChances ← nbChances − 1;
            // Intensification;
            RefSolution ← RefSolution;
        else
            // Diversification
              RefSolution ← GenerateInitialSolution();
        end
    end
    iter ← iter + 1;
end
BestModularity ← BestModularity(TabuList) ;
Decode the optimal solution to obtain the communities of
  network G, C = {C1,C2,...,Ck}
```

To prevent repetition, the latter is stored in a taboo list. The algorithm continues until an optimal solution is found or the maximum of iterations is reached.

In this paper, we adapt the BSO algorithm for detecting community structure in networks, which involves specific considerations. The process is given in Algorithm 1, and in what follows, we outline the encoding of solutions and the evaluation of their quality, the generation of the initial solution and the determination of the search region, the local search conducted by the bees and the selection of the reference solution.

3.1 Solution Encoding

In order to fit the characteristics of networks, we employ the locus-based adjacency representation (Shi et al., 2009) to encode solutions representing community structures. This encoding scheme offers a significant advantage by automatically detecting the number of communities. For a particular network with n nodes, every individual g in the population is represented as $\langle g_1, g_2, ..., g_n \rangle$ with n genes, and each g_i can take the allele value j in the range of $\langle 1, 2, ..., n \rangle$. Assigning a value j to the i^{th} gene implies an edge between nodes i and j in V, resulting in the nodes i and j belonging to the same community in the detected structure. To extract the community structure from this representation, a decoding step is required to identify all connected components of the graph, ensuring that nodes within the same component are assigned to the same community. It has been demonstrated that this decoding method is highly effective for community detection and can be performed efficiently in linear time (Cormen et al., 2001). For illustration, Fig. 1 shows a network, its corresponding solution using the locus-based representation, and the final community structure result.

3.2 Initial Reference Solution

Initially, the algorithm generates a solution to represent the BeeInit's food source. This solution is generated randomly or via a heuristic. For community detection problem, a randomly generated solution may not be feasible. Indeed, it could contain a value j in the i^{th} position, but no edge (i, j) exists in the network. Therefore, in order to obtain a solution with certain quality, for each node i, a value j in the i^{th} position is randomly chosen among its neighbor nodes. This means that the edge (i, j) exists in the network. This strategy provides a good starting point for the algorithm and improves the convergence of the method.

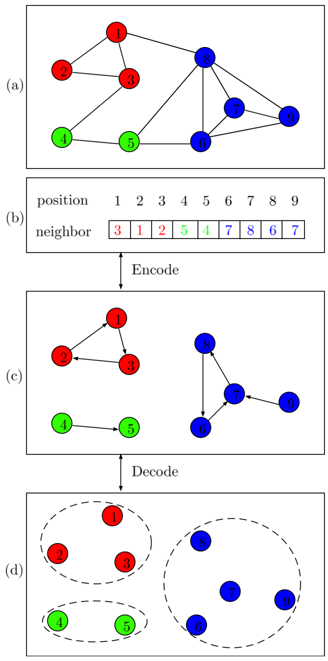

Figure 1: Example of a solution representation (Dakiche et al., 2022).

3.3 Search Space

The search space consists of a collection of N solutions, where N corresponds to the number of bees in the swarm, since each solution will be assigned to one bee as a starting point of its local search. These solutions within the search space are generated by flipping a number of bits equal to n/flip from the reference solution (Refsolution), flip being an empirical parameter that significantly affects the search process. A small Flip value implies a local optimum for Refsolution in the exploitation region, limiting potential improvement. Conversely, a high Flip value may cause the swarm to move away from Refsolution's region, potentially overlooking valuable solutions.

To ensure diverse solutions within the search space, we employ two strategies (Sadeg et al., 2015). In the first strategy, the k^{th} solution is generated by flipping variables in Refsolution with a separation of flip bits, starting from the k^{th} position. While in the second one, the k^{th} solution is obtained by flipping n/flip contiguous bits, starting from the k^{th} position.

If we consider the example of Fig.1 in which $N = 9$ and we take $flip = 3$. With the positions labeled from 1 to 9, the solutions we obtain by using the first strategy are generated by flipping the following positions: (1,4,7), (2,5,8), and (3,6,9), while using the second strategy produces the solutions generated by flipping the following positions: (1,2,3), (4,5,6), and (7,8,9) as illustrated in Fig.2. The positions given by both strategies are changed by randomly selecting a neighbor of the corresponding node from its list of neighbors.

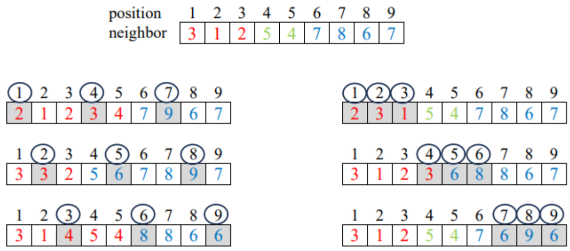

Figure 2: (left) solutions generated by the first strategy, (right) solutions generated by the second strategy.

If the number of generated solutions is insufficient (less than the defined number of bees), a random approach can be used. These strategies aim to maintain distinct solutions while exploring the search space effectively.

3.4 Neighborhood Procedure

The bees explore their neighborhood for new eventually better solutions. To obtain a neighbor solution, we propose to take advantage of the prior knowledge inferred from the network structure, and use a Pearson's-based neighbor procedure. It modifies various entries of the solution vector using the roulette method. This method determines the probability of selecting a pair (i, j) as a component of the new solution. Specifically, the value j is chosen as the new value for the input index i with a probability of $p = n_{ij}/\sum_{k\in v(i)} n_{ik}$, where n_{ij} represents the Pearson correlation between nodes i and j, and $v(i)$ denotes the set of neighboring nodes of i. Fig. 3 illustrates the generation of neighboring solutions.

The Pearson's correlation similarity measure (Fortunato, 2010) is based on the fact that the nodes belonging to the same community are similar to each other. The Pearson correlation $C(i, j)$ between nodes v_i and v_j is defined as follows:

$$C(i, j) = \frac{\sum_{vl\in V}(A_{il} - \mu_i)(A_{jl} - \mu_j)}{n\sigma_i\sigma_j} \tag{2}$$

where A_{il} is the l^{th} element of the i^{th} row in the adjacency matrix, $\mu_i = \sum_l A_{il}/n$ is the average, and $\sigma_i = \sqrt{\sum_l (A_{il} - \mu_i)^2/n}$ is the standard deviation. $C(i, j)$ takes values between -1 and 1. A value close to 1 means that nodes v_i and v_j are very similar in a structural perspective. Otherwise, $C(i, j)$ would be close to -1. As the Pearson correlation can have negative values, it cannot be directly used in BSODCS algorithm. To address this, we apply the logistic function to the

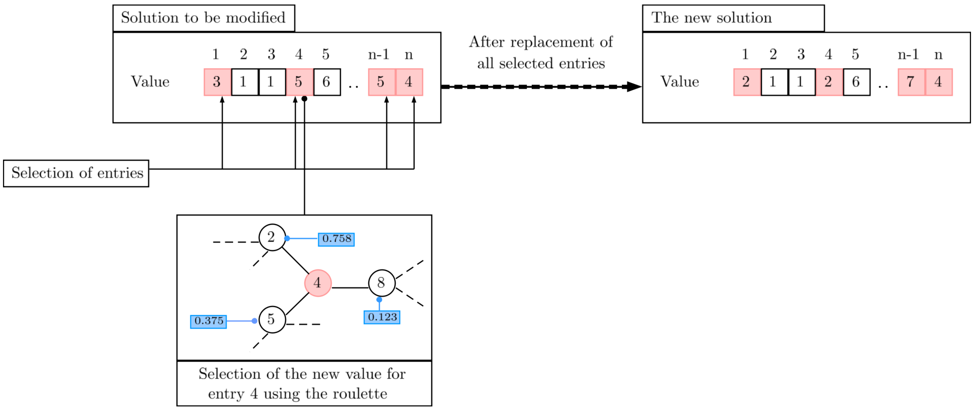

Figure 3: Pearson's-based neighbor procedure (Dakiche et al., 2022).

aforementioned Pearson correlation, resulting in positive outputs. These outputs are incorporated into the algorithm as heuristic information:

$$n_{ij} = \frac{1}{1 + e^{-C(i,j)}} \tag{3}$$

The more similar nodes v_i and v_j are the larger value n_{ij} would take, and the more probably edge (i, j) is to be taken as a solution component.

3.5 Selection of Reference Solution

The selection of the reference solution, for next iteration, initially relies on the modularity of solutions in the dance table. However, if the swarm observes a lack of progress in terms of modularity over a certain period of time, it incorporates a restart strategy. This restart enables the swarm to escape from a potentially stagnant region and explore alternative regions in the search space. By combining both quality and diversity considerations, the swarm ensures a more comprehensive and effective search process.

4 EXPERIMENTAL RESULTS

In this section, our algorithm was tested on a selection of commonly used networks in the community detection problem: American college football, Zachary's karate club, the Dolphin social network, and Books about US politics. To assess the quality of the resulting community structures, we used modularity as a quality measure, and compared the results with five other swarm based optimization algorithms from literature: PSO (Chen et al., 2012), ACO (Chen et al., 2012), BA (Song et al., 2016), ABC (Dakiche et al., 2022), and BSOCD (Belkhiri et al., 2017). In what follows, we present the used datasets and discuss the results.

4.1 Datasets

The American college football network, initially proposed by Girvan and Newman (Girvan and Newman, 2002), it represents the matches played between college football teams in the United States during the 2000 season. This network consists of 115 nodes representing teams and 616 edges representing the games played, organized into 12 teams. The Zachary's karate club network (Zachary, 1977) describes the members of a karate club, comprising 34 nodes, and the relationships between them, represented by 78 edges. Due to internal disagreements within the club, the members eventually split into two groups. The Dolphins network (Lusseau et al., 2003), constructed by Lusseau, focuses on a group of 62 dolphins observed over a period of seven years. The network exhibits a division of the dolphins into two distinct groups, with 159 connections. The Books about US politics network (Newman, 2006) consists of 105 nodes, representing books on American politics acquired from Amazon.com. The edges between books denote frequent co-purchasing of books by the same buyer. The books were categorized into two groups based on their political alignment (liberal or conservative), with the exception of 13 books that did not have a clear affiliation.

4.2 Results Discussion

The performance of BSODCS, like most metaheuristics, is heavily influenced by its parameter values. In our experiments, we determined the optimal parameter values through extensive runs using various parameter combinations. Consequently, we set the parameters as follows: $flip = 5$, $limit = 3$, the number of bees $N = 20$, and the number of iterations 50.

Table 1 provides the modularities obtained from our algorithm, as well as those from other comparative methods. Our algorithm outperforms the other algorithms for all networks, while ABC algorithm obtains the same results for Karate and Dolphins networks. This demonstrates the effectiveness of our BSODCS in real-world network analysis.

Table 1: Modularity results for all datasets.

Methods	Football	Karate	Dolphins	Books
PSO	0.5630	0.3690	-	0.4700
ACO	0.6031	0.4165	0.5628	0.5262
BA	0.5960	0.3920	-	0.4790
ABC	0.6009	**0.4198**	**0.5285**	0.5116
BSOCD	0.6040	0.4197	0.5140	-
BSODCS	**0.6043**	**0.4198**	**0.5285**	**0,5265**

If we consider the Zachary's karate club, which naturally consists of two communities of equal size, our BSODCS algorithm splits the network into four communities as shown in Fig.4, yielding the highest modularity value of Q=0.4198 given in Table 1.

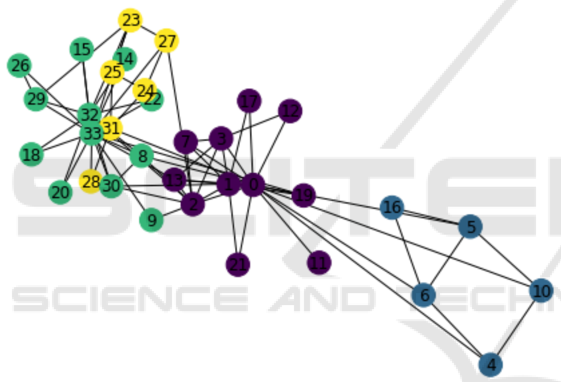

Figure 4: Karate Club communities using BSODCS.

5 CONCLUSIONS

In this paper, we investigated the applicability of bee swarm optimization for community detection problem. The proposed algorithm, BSODCS, uses the modularity Q as objective function and starts with an initial reference solution. A search space is created from this reference solution, then a group of bees collaboratively works to maximize the global function Q. Each bee operates independently within its neighborhood and communicates its findings through a dance. To assess the effectiveness of our algorithm, we conducted experiments on four real-world networks. The results obtained demonstrate the validity and efficiency of our method for community detection problem. In future work, we aim to extend our approach to address community detection problem in dynamic networks, aiming to further enhance the quality of the obtained results as well as tracking communities' evolution.

REFERENCES

Bedi, P. and Sharma, C. (2016). Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(3):115–135.

Belkhiri, Y., Kamel, N., Drias, H., and Yahiaoui, S. (2017). Bee swarm optimization for community detection in complex network. In *Recent Advances in Information Systems and Technologies: Volume 2 5*, pages 73–85. Springer.

Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2007). On modularity clustering. *IEEE transactions on knowledge and data engineering*, 20(2):172–188.

Cai, Q., Gong, M., Ma, L., Ruan, S., Yuan, F., and Jiao, L. (2015). Greedy discrete particle swarm optimization for large-scale social network clustering. *Information Sciences*, 316:503–516.

Cai, Q., Gong, M., Shen, B., Ma, L., and Jiao, L. (2014). Discrete particle swarm optimization for identifying community structures in signed social networks. *Neural Networks*, 58:4–13.

Chen, B., Chen, L., and Chen, Y. (2012). Detecting community structure in networks based on ant colony optimization. In *Proceedings of the International Conference on Information and Knowledge Engineering (IKE)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer ....

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). Introduction to algorithms second edition.

Dakiche, N., Benatchba, K., Tayeb, F. B.-S., Slimani, Y., and Brahmi, M. A. (2022). A hybrid artificial bee colony algorithm with simulated annealing for enhanced community detection in social networks. In *2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 104–108. IEEE.

Djeffal, M. and Drias, H. (2013). Multilevel bee swarm optimization for large satisfiability problem instances. In Yin, H., Tang, K., Gao, Y., Klawonn, F., Lee, M., Weise, T., Li, B., and Yao, X., editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2013*, pages 594–602, Berlin, Heidelberg. Springer Berlin Heidelberg.

Djenouri, Y., Drias, H., and Chemchem, A. (2013). A hybrid bees swarm optimization and tabu search algorithm for association rule mining. In *2013 World Congress on Nature and Biologically Inspired Computing*, pages 120–125. IEEE.

Drias, H., Sadeg, S., and Yahi, S. (2005). Cooperative bees swarm for solving the maximum weighted satisfiability problem. In *International Work-Conference on Artificial Neural Networks*, pages 318–325. Springer.

Fortunato, S. (2010). Community detection in graphs. *Physics reports*, 486(3-5):75–174.

Fortunato, S. and Barthélemy, M. (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41.

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826.

Hafez, A. I., Zawbaa, H. M., Hassanien, A. E., and Fahmy, A. A. (2014). Networks community detection using artificial bee colony swarm optimization. In *Proceedings of the fifth international conference on innovations in bio-inspired computing and applications IBICA 2014*, pages 229–239. Springer.

Hassan, E. A., Hafez, A. I., Hassanien, A. E., and Fahmy, A. A. (2015). A discrete bat algorithm for the community detection problem. In *Hybrid Artificial Intelligent Systems: 10th International Conference, HAIS 2015, Bilbao, Spain, June 22-24, 2015, Proceedings 10*, pages 188–199. Springer.

He, D., Liu, J., Liu, D., Jin, D., and Jia, Z. (2011). Ant colony optimization for community detection in large-scale complex networks. In *2011 Seventh International Conference on Natural Computation*, volume 2, pages 1151–1155. IEEE.

Jaradat, A. S. et al. (2018). Community structure detection using firefly algorithm. *International Journal of Applied Metaheuristic Computing (IJAMC)*, 9(4):52–70.

Karaboga, D. and Akay, B. (2009). A survey: algorithms simulating bee swarm intelligence. *Artificial intelligence review*, 31:61–85.

Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E., and Dawson, S. M. (2003). The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: can geographic isolation explain this unique trait? *Behavioral Ecology and Sociobiology*, 54:396–405.

Nayyar, A., Le, D.-N., and Nguyen, N. G. (2018). *Advances in swarm intelligence for optimizing problems in computer science*. CRC press.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582.

Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113.

Sadeg, S. and Drias, H. (2007). A selective approach to parallelise bees swarm optimisation metaheuristic: Application to max-w-sat. *Int. J. Innov. Comput. Appl.*, 1(2):146–158.

Sadeg, S., Hamdad, L., Benatchba, K., and Habbas, Z. (2015). Bso-fs: Bee swarm optimization for feature selection in classification. In Rojas, I., Joya, G., and Catala, A., editors, *Advances in Computational Intelligence*, pages 387–399, Cham. Springer International Publishing.

Shi, C., Wang, Y., Wu, B., and Zhong, C. (2009). A new genetic algorithm for community detection. In Zhou, J., editor, *Complex Sciences*, pages 1298–1309, Berlin, Heidelberg. Springer Berlin Heidelberg.

Song, A., Li, M., Ding, X., Cao, W., and Pu, K. (2016). Community detection using discrete bat algorithm. *IAENG International Journal of Computer Science*, 43(1):37–43.

Yang, J., McAuley, J., and Leskovec, J. (2013). Community detection in networks with node attributes. In *2013 IEEE 13th international conference on data mining*, pages 1151–1156. IEEE.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473.