2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

A Hybrid Artificial Bee Colony Algorithm with Simulated Annealing for Enhanced Community Detection in Social Networks

Narimene Dakiche*, Karima Benatchba*, Fatima Benbouzid-Si Tayeb*, Yahya Slimani†, Mehdi Anis Brahmi*
*Laboratoire des Méthodes de Conception de Systèmes (LMCS),
Ecole nationale Supérieure d'Informatique (ESI), BP 68M - 16270 Oued Smar, Alger, Algérie.
{an_dakiche, k_benatchba, f_sitayeb, cm_brahmi }@esi.dz
†Computer Science Department, ISAMM Institute of Manouba, Manouba University, Manouba, Tunisia
yahya.slimani@fst.rnu.tn

***Abstract*—In this paper, we propose a hybrid Artificial Bee Colony algorithm with Simulated Annealing (ABC-SA) to address the community detection problem. SA enhances the exploitation by searching the most promising regions located by ABC algorithm. Besides, in order to accommodate the characteristics of social networks, we use locus-based adjacency encoding scheme, in which communities are identified as a graph connected components and Pearson's correlation as structural information to guide the solutions' construction. Results obtained on synthetic and real-word networks show that the proposed algorithm can discover communities more successfully in comparison with traditional ABC algorithm and other state-of-the-art algorithms.**

I. Introduction

The problem of detecting communities in social networks according to many interesting objective functions, like maximizing modularity, has been proved to be NP-hard. Consequently, several metaheuristics are proposed in the literature including greedy optimization, genetic algorithms, multiobjective optimization, and so forth [1]. In recent decades, the popularity of swarm intelligence algorithms has increased. Thus, the artificial bee colony (ABC) algorithm inspired by the intelligent, self-organizing and aggregated behavior of bee colonies [2], was adopted to solve the community detection problem.

Abu Naser and Alshattnawi [3] used ABC to detect communities by maximizing the modularity measure. They adopted a string-based representation for food sources (solutions i.e community structures) which assume that the number of communities is already known. Their results showed that ABC is very successful for community detection problem. Hafez et al. [4] used ABC more efficiently to detect communities. They used locus-based adjacency encoding scheme to represent a food source which has the advantage of enabling the algorithm to detect communities' number automatically. Besides, they used different popular communities' quality measures as an objective function and their results revealed that optimizing modularity yields the best community structure. Wang et al. [5] proposed an improved ABC for community detection. Their algorithm modifies the number of initial food sources and dynamically adjusts search scope to prevent falling into local optimization. Experimental results showed that their algorithm outperformed traditional ABC in complex networks. Guo et al. [6] proposed a Heuristic ABC based community detection algorithm. They redefined the searching process of ABC and used an agglomerate probability of two neighbor communities as heuristic function.

Nonetheless, ABC algorithm can easily be trapped in a local optima. In order to prevent this shortcoming for community detection problem, we combine it with Simulated Annealing (SA) [7] aiming to enhance the exploitation by searching the most promising regions located by ABC algorithm. SA is a metaheuristic inspired by the concept of annealing solids and the crystallizing behavior of chemical substances [8]. Indeed, combining various features of more than one metaheuristic that perform together and complement each other may result in more robust combinatorial optimization tools [9]. The main features of our proposed hybrid algorithm, hereinafter **ABC-SA**, are as follows. First, ABC-SA ensures a good balance between intensification and diversification of the research space. Besides, we adopt the locus-based adjacency representation to record each solution representing a community structure. Moreover, we employ Pearson's correlation, a similarity measure between nodes, to construct these solutions. The algorithm reaches better modularities compared to traditional ABC and other algorithms from the literature.

The rest of this paper is organized as follows: Section II presents the detailed description of the proposed ABC-SA algorithm. The results of experiments conducted on synthetic and real-world networks are presented in Section III. The conclusion and future works are given in Section IV.

II. Proposed ABC-SA Algorithm

Before we introduce the proposed ABC-SA, we would like to briefly introduce some basic network concepts. A social network can be modeled as a graph $G = (V, E)$, which includes $N = |V|$ nodes and $m = |E|$ edges. V represents the set of nodes and E represents the set of edges in G. A is an $N \times N$ adjacency matrix of the network, where A_{ij} represents

IEEE/ACM ASONAM 2022, November 10-13, 2022
978-1-6654-5661-6/22/$31.00

the element at the i^{th} row and j^{th} column of A. If there is an edge from node i to node j, $A_{ij} = 1$; otherwise $A_{ij} = 0$. For a final community structure $P = \{C_1, C_2, ..., C_k\}$ of a network, each element $C_l (l = 1, 2, ..., k)$ is a proper subset of V, and k is the total number of communities.

In this paper, we design a hybrid algorithm **ABC-SA** for community detection in which we introduce SA as a step after each ABC cycle to balance global exploration and local exploitation during the evolutionary process. Thus, ABC-SA involves two steps, each one is designed to fulfill a specific aim. The process of ABC-SA is described in Algorithm 1. In what follows, we discuss its details.

Algorithm 1 ABC-SA for static community detection

```
Input: A network G = (V, E)
Output: Community structure of the network C = {C_1, C_2, ..., C_k}
Initialize the algorithm' parameters: PS (population size), limit (abandonment
  criterion), itermax (maximum number of iterations), T (initial temperature), and
  α (cooling factor)
  // FoodSourcesInitialization(PS), see subsection II-D1
for i ← 1 to PS do
  EBee[i] ← GenerateInitialSolution();
end for
Compute the fitness value
Find the current optimal solution (BestModularity)
Set iter = 0
while iter < itermax do
    // EmployedBeesPhase (PS), see subsection II-D2
  for i ← 1 to PS do
    NeighborEBee ← GenerateNeighborEBee(EBee[i]);
    if NeighborEBee > EBee[i] then
      EBee[i] ← NeighborEBee;
    end if
  end for
    // OnlookerBeesPhase(PS), see subsection II-D3
  for i ← 1 to PS do
    OBee[i] ← ProbabilisticSelectionEBee();
    NeighborOBee ← GenerateNeighborBee(OBee[i]);
    if NeighborOBee > OBee[i] then
      OBee[i] ← NeighborOBee;
    end if
  end for
    // ScoutBeesPhase(limit, PS), see subsection II-D4
  for i ← 1 to PS do
    CheckAbandon(EBee[i], limit);
     CheckAbandon(OBee[i], limit);
  end for
  UpdateOptimalSolution (BestModularity)
    // SimulatedAnnealingPhase, see subsection II-E
  SimulatedAnnealing (BestModularity, T, α)
  UpdateOptimalSolution (BestModularity)
  iter ← iter + 1;
end while
Decode the optimal solution to obtain the communities of network t, C =
  {C_1, C_2, ..., C_k}
```

A. Solution Representation

We use the locus-based adjacency representation [4] to record solutions as follows: for a given network with n nodes, every individual g in the population is expressed as $\langle g_1, g_2, ..., g_n\rangle$ with n genes, and each g_i can take the value j in the range of $\langle 1, 2, ..., n\rangle$. A value j assigned to the i^{th} gene is interpreted as an edge between the nodes i and j of V. Thus, in the detected community structure, nodes i and j will be in the same community. A decoding step is necessary to identify all the connected components of the graph, so that nodes of the same component are assigned to the same community. It has been shown that this decoding schema is very effective for community detection and can be efficiently done in linear time [10]. Fig. 1(a) shows a network with 9 nodes, 14 edges, and three communities visualized by different colors. A solution of locus-based representation is shown in Fig. 1(b), that corresponds to the graph division given in Fig. 1(c). The final community structure result is given in Fig. 1(d).

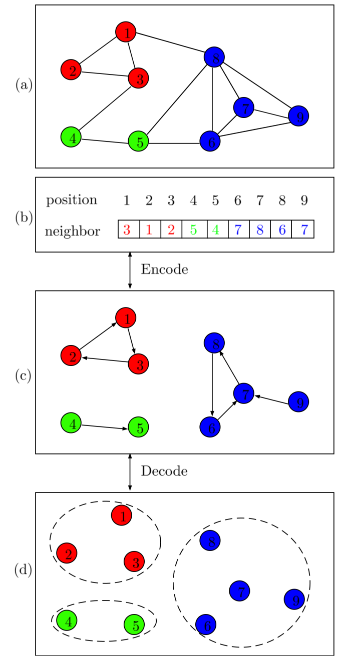

Fig. 1. Illustration of the locus-based adjacency representation steps.

B. Objective Function

The objective function plays an important role in the optimization process. It is the "steering wheel" in the process that leads to good solutions. Modularity Q is by far the most used and best known quality function for measuring the quality of a partition of a network [11], and many community detection algorithms are developed to optimize it [12]. For each graph G, with n nodes and m edges, modularity is computed as follows:

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

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. High modularity denotes that connections between nodes of the same community are dense and connections between nodes belonging to different communities are sparse [1].

C. Pearson's correlation

Based on the fact that the nodes belonging to the same community are similar to each other, we used Pearson's correlation similarity measure [1] to lead the generations of neighbor solutions by the bees. The Pearson correlation $C(i, j)$ between nodes v_i and v_j is defined as follows:

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

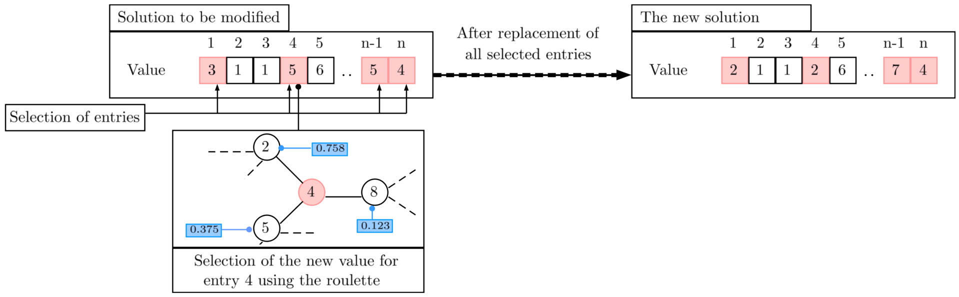

Fig. 2. Pearson's-based neighbor procedure.

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. Since the Pearson correlation may take a negative value, we apply the logistic function to the correlation defined above to make the outputs positive.

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

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.

D. Exploration step

The ABC process includes four steps that we detail in what follows.

1) Population Initialization: Solutions are initialized by employed bees. However, a randomly generated solution could contain a value j in the i^{th} position, but no edge (i,j) exists in the network. Therefore, to obtain a population with certain quality and diversity, for each node i, a value j in the i^{th} position is randomly chosen among its neighbor nodes. This strategy provides a diversity in the population and improves the convergence of the method.

2) Employed bees: Employed bees look, within their neighborhood, for new better solutions. To obtain a neighbor solution, we propose a Pearson's-based neighbor procedure (Fig. 2) which modifies several entries of the solution vector. The scope of the search is defined by the percentage of entries to be modified δ. Different percentages are used to discover new regions of the search space. The modification of the selected entries is done using the roulette method where the probability that a stop (i,j) is taken as a component of the new solution is $p = n_{ij}/\sum_{k\in v(i)} n_{ik}$, where n_{ij} is the Pearson correlation between nodes i and j, and $v(i)$ is the set of neighbor nodes of i. If it has a greater modularity than the current solution, then it replaces the current solution for this employed bee.

3) Onlooker bees : Onlooker bees select their food sources based on solutions' fitness information shared by employed bees. We apply the roulette wheel selection, thus onlooker bees will select better solutions. The probability $p(x^i)$ with which a solution x^i is selected is computed by Eq. (4).

$$p(x^i) = \frac{fitness(x^i)}{\Sigma_{m=1}^{PS} fitness(x^m)} \tag{4}$$

Once an onlooker bee selects probabilistically a solution x^i, it generates a neighbor solution. If the latter has a greater modularity then it replaces the selected solution for this onlooker bee.

4) Scout bees : The scout bees perform a random restart and search for new solutions. Therefore, employed and onlooker bees whose solutions cannot be improved, after a predetermined number of attempts called limit, become scout bees. In our case, the new solution of a scout bee is generated randomly from scratch or by greatly disturbing the current one. The obtained scout bee is improved by using Pearson's-based neighbor procedure.

E. Exploitation step

The SA step starts with the best solution located by the bees in the current iteration. The behavior of SA depends on two important parameters which are the initial temperature T and the cooling factor α. SA goes through two phases: when T is large, all visited solutions using Pearson's-based neighbor procedure are accepted with a high probability. However, as T decreases, only good solutions are accepted.

III. Empirical study

In order to study the effectiveness of ABC-SA, we conducted many experiments on both synthetic and real networks. For synthetic networks, we used the two current gold standard benchmark networks: GN and LFR [13]. The former type has well defined community structures. We used, for experiments, its mixing parameter μ up to 0.8 with an interval of 0.05. When the mixing parameter increases, the border between communities becomes more ambiguous which makes their detection more difficult. LFR can control the distribution of nodes' degree and communities' size in a proper way. By varying its parameters, we generated two groups of benchmark networks with small (A) and big (B) community sizes. As for real networks, we used a set of 15 networks extracted from various domains with different scales and degree distributions.

We compared our ABC-SA with six algorithms [13], namely GN, CNM, SCAN, Louvain, BSO [14], and ABC without

SA. We adopted two widely used criteria to evaluate the accuracy of community detection algorithms: (1) normalized mutual information (NMI) to measure the similarity between the ground truth community structure and the detected one in synthetic networks; (2) and modularity to measure the quality of the detected community structure in real networks for which no ground truth is available.

We have undertaken a sensitive analysis of performance for ABC-SA by varying its different parameters. Then, we set each parameter to its most interesting level. Table I resumes these parameters, their meaning and values used.

TABLE I
PARAMETERS AND NOTATIONS OF ABC-SA

Parameter	Meaning	Value
Colony size (CS)	number of candidate solutions	100
Maximum cycles (C)	Number of iterations	250
Scout limit	limit to abandon a solution	15
Temperature (T)	value for accepting worse solutions	1000
Cooling rate (α)	mechanism of temperature reduction	0.8

A. ABC-SA performance analysis on synthetic networks

Figure 3 illustrates the experimental results on GN benchmarks. Most algorithms get $NMI = 1$ when the mixing parameters μ is smaller than 0.15, indicating that detected communities match the ground-truth perfectly. However, as the mixing parameter μ increases, this performance decreases gradually. When μ is no larger than 0.3, both Louvain and ABC-SA can uncover the true communities. Until $\mu = 0.40$ Louvain algorithm maintains its performance while ABC, ABC-SA and CNM still give acceptable results. When μ increases further, the quality of algorithms' results decreases because the border between communities becomes more ambiguous. As one can see, when $\mu = 0.45$, NMI score of ABC-SA is slightly lower than the one of Louvain and higher than the one of CNM. In addition, ABC performs better than CNM when $\mu > 0.45$. It even gets best NMI results than Louvain when $\mu > 0.60$ while ABC-SA still outperforms all the algorithms when $\mu > 0.5$.

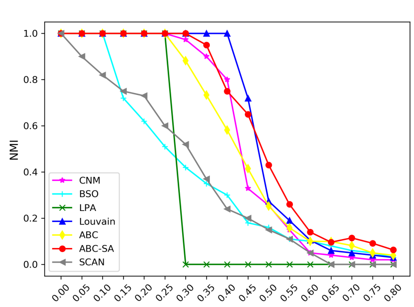

Fig. 3. NMI results on GN benchmark networks.

Figure 4 illustrates the NMI scores on LFR benchmarks. Plots (A) and (B) correspond to the same network size with two different community sizes (small $20 \sim 50$ and big $30 \sim 100$). As shown in both plots, ABC-SA performs better than other algorithms when $\mu < 0.55$ except for Louvain. Yet, its results are still acceptable (> 0.8) for $\mu < 0.40$ when Louvain gets $NMI = 1$. As shown in Fig.4(A), ABC-SA seems more successful in finding community structures when these are complicated and ambiguous ($\mu >= 0.55$). This can be explained by the diversification of our algorithm provided during the generation of the initial population of solutions and the generation of scout bees. The same result is obtained in Fig.4(B) except for $\mu = 0.65$ in which Louvain shows its advantage in identifying big communities. Yet, we find that ABC-SA still has an advantage over all the other algorithms.

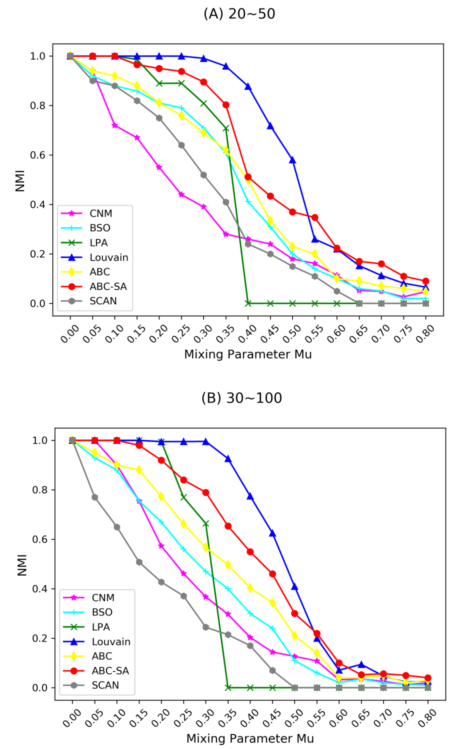

Fig. 4. NMI results on LFR benchmarks with $n = 500$. (A) Small size communities. (B) Big size communities.

B. ABC-SA performance analysis on Real networks

Table II shows the experimental results of modularity on real-world networks. Numbers in boldface indicate the highest modularity values in the corresponding row. As one can notice, ABC-SA algorithm gets optimal modularity results for two-thirds of the 15 real-world networks (66.67%), compared to other algorithms, while Louvain is the only one obtaining the best results for 4 networks (26.67%). This result shows that ABC-SA is effective to detect community structures of real-world networks. In addition, it achieves better results on 12 networks (86.67%) compared to GN, SCAN, and BSO which makes it a competitive method for community detection problem.

In order to ascertain the statistical significance of the obtained results, we have performed Friedman test [15] with Li

TABLE II
COMPARISON OF MODULARITY RESULTS ON REAL-WORLD NETWORKS

Network	GN	CNM	SCAN	Louvain	BSO	ABC	ABC-SA
Karate	0.4013	0.3807	0.3409	**0.4198**	**0.4198**	**0.4198**	**0.4198**
Contig-USA	0.5958	0.5824	0.5387	0.5902	0.5882	0.5935	**0.5970**
Dolphin	0.5194	0.4955	0.2887	0.5185	0.5253	0.5285	**0.5290**
Lesmis	0.5381	0.5006	0.2258	0.5548	0.5430	0.5594	**0.5600**
Polbooks	0.5168	0.5020	0.4045	0.5270	0.5189	0.5116	**0.5272**
Word	0.0805	0.2947	0.1130	0.2904	0.2231	0.2933	**0.3072**
Football	0.5996	0.5497	0.5143	0.6043	0.5146	0.6009	**0.6054**
Jazz	0.4051	0.4389	0.2689	0.4438	0.3125	0.4384	**0.4448**
Neural	0.3010	0.3728	0.2256	**0.3926**	0.2319	0.3137	0.3870
08blocks	0.8599	**0.8750**	**0.8750**	**0.8750**	0.8104	**0.8750**	**0.8750**
Metabolic	0.4048	0.4172	0.3078	**0.4405**	0.3128	0.3664	0.4194
Polblogs	0.4180	**0.4270**	0.3269	0.4269	0.3642	0.3702	0.4259
Netscience	0.9579	0.9551	0.8957	**0.9592**	0.9006	0.9086	0.9476
Facebook	0.8079	**0.8087**	0.7902	**0.8087**	0.8086	**0.8087**	**0.8087**
Collaboration	0.8492	0.8119	0.6945	**0.8624**	0.7054	0.7028	0.7479

http://konect.uni-koblenz.de/networks/
http://networkrepository.com/networks

post-hoc [16], a nonparametric statistical tests on the modularity values. The test was rejected with a p-value of 0.05, which means that indeed, all the algorithms behave differently when tested on multiple real-word networks. Moreover, the post-hoc indicated that ABC-SA significantly outperforms SCAN, BSO and GN with the same confidence interval, and even CNM when p-value is equal to 0.1. Despite the fact that the test indicated that ABC-SA and Louvain cannot be distinguished, ABC-SA can still be considered as effective since Louvain algorithm is seen as a reference in the field.

IV. CONCLUSION

In this paper, we proposed a hybrid ABC with SA algorithm (**ABC-SA**) for the community detection problem in social networks. The locus-based adjacency representation is adopted to record solutions, and Pearson's correlation is employed to construct them. Our experiments on both synthetic and real-world networks shown that ABC-SA can accurately and effectively uncover community structures. It leads to better modularities than basic ABC and other existing algorithms. In this paper, we focused on static networks; however, most real networks evolve over time and community structures as well [17]. Therefore, our focus in the future will be to propose algorithms that can dynamically detect community structure.

REFERENCES

[1] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[2] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, Nov 2007.

[3] A. M. Abu Naser and S. Alshattnawi, "An artificial bee colony abc algorithm for efficient partitioning of social networks," *Int. J. Intell. Inf. Technol.*, vol. 10, no. 4, pp. 24–39, Oct. 2014.

[4] A. I. Hafez, H. Zawbaa, A. E. Hassanien, and A. Fahmy, "Networks community detection using artificial bee colony swarm optimization," in *Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications*, 2014, pp. 229–239.

[5] Z. Wang, X. Zhao, P. Wen, J. Xue, and C. Hu, "Community detection in complex networks using improved artificial bee colony algorithm," 2016.

[6] Y. Guo, X. Li, Y. Tang, and J. Li, "Heuristic artificial bee colony algorithm for uncovering community in complex networks," *Mathematical Problems in Engineering*, vol. 2017, 2017.

[7] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[8] C. Wang, D. Mu, F. Zhao, and J. W. Sutherland, "A parallel simulated annealing method for the vehicle routing problem with simultaneous pickupâĂŞdelivery and time windows," *Computers & Industrial Engineering*, vol. 83, pp. 111 – 122, 2015.

[9] G. R. Raidl, "A unified view on hybrid metaheuristics," in *Hybrid Metaheuristics*, F. Almeida, M. J. Blesa Aguilera, C. Blum, J. M. Moreno Vega, M. Pérez Pérez, A. Roli, and M. Sampels, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to algorithms second edition," 2001.

[11] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[12] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, oct 2008.

[13] P. Bedi and C. Sharma, "Community detection in social networks," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 6, no. 3, pp. 115–135, 2016.

[14] Z. Beldi and M. Bessedik, "A new brainstorming based algorithm for the community detection problem," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, June 2019, pp. 2958–2965.

[15] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the american statistical association*, vol. 32, no. 200, pp. 675–701, 1937.

[16] J. D. Li, "A two-step rejection procedure for testing multiple hypotheses," *Journal of Statistical Planning and Inference*, vol. 138, no. 6, pp. 1521 – 1527, 2008.

[17] N. Dakiche, F. B.-S. Tayeb, Y. Slimani, and K. Benatchba, "Tracking community evolution in social networks: A survey," *Information Processing & Management*, vol. 56, no. 3, pp. 1084 – 1102, 2019.