Hindawi
Complexity
Volume 2021, Article ID 6699130, 14 pages
https://doi.org/10.1155/2021/6699130

Research Article

Social Network Community Detection by Combining Self-Organizing Maps and Genetic Algorithms

Mehdi Ellouze

Sfax University, Department of Computer Science, Faculty of Economic Sciences and Management of Sfax, Airport Road Km 4, 3018 Sfax, Tunisia

Correspondence should be addressed to Mehdi Ellouze; mehdi.ellouze@ieee.org

Received 16 December 2020; Accepted 13 September 2021; Published 21 October 2021

Academic Editor: Ning Cai

Social networks have become an important source of information from which we can extract valuable indicators that can be used in many fields such as marketing, statistics, and advertising among others. To this end, many research works in the literature offer users some tools that can help them take advantage of this mine of information. Community detection is one of these tools and aims to detect a set of entities that share some features within a social network. We have taken part in this effort, and we proposed an approach mainly based on pattern recognition techniques. The novelty of this approach is that we do not directly tackle the social networks to find these communities. We rather proceeded in two stages; first, we detected community cores through a special type of self-organizing map called the Growing Hierarchical Self-Organizing Map (GHSOM). In the second stage, the agglomerations resulting from GHSOM were grouped to retrieve the final communities. The quality of the final partition would be under the control of an evaluation function that is maximized through genetic algorithms. Our system was tested on real and artificial databases, and the obtained results are really encouraging.

1. Introduction

Social networks involve such a wealth that various people from different fields try to exploit for valuable information. Information coming from social networks is used in different areas such as marketing, politics, economy, statistics, and education [1]. Community detection drew the attention of a lot of researchers over the last few years [1, 2]. Knowing the structure of communities of individuals inside a social network helps target suitable people when achieving marketing campaigns for instance or when trying to understand the opinion of a given social category. Social networks are made of individuals called nodes, like profiles on Facebook or LinkedIn. Two kinds of features are characterizing every node: topological features and semantic features. The topological features are based on the links existing between nodes [1]. Nodes belonging to the same community are densely linked. However, semantic features are related to information proper to each node such as age, family, education, and comments. Nodes belonging to the same community generally share some common information. As it is complex to extract semantic information, most works only rely on links to extract the communities.

Communities denote a collective behavior of nodes and involve nodes that are strongly linked. Within one community, nodes do not have the same importance. Some of them represent the core of the community and attract all the community nodes, whereas others are peripheral nodes. They are located on the border of the community. Detecting communities in a social network is a complex task because nothing is known about the structure of the communities, their size, their core nodes, and so on.

In the literature [2], a social network is considered as a graph $G = (V, E)$, where V is a set of nodes or vertices and E is a set of links, called edges that connect two elements of V. Detecting communities means detecting subgraphs of nodes with strong interactions between them and little interaction with the other subgraphs. However, the main challenge is the lack of a clear quantitative criterion that can be used to delimit these subgraphs. This explains why most of the works

deal with the extraction of communities in a sequential way. They compare nodes two by two. This paper proposed a different approach because it considered that grouping nodes into communities has to be achieved to relatively all the nodes and not through finding out direct similarities between them two by two. This study, therefore, proposed to detect agglomerations of nodes as a first step in the process of communities' detection. Agglomerations are not necessarily the final communities. They represent the cores of the final communities. Once detected, these agglomerations will compete to attract each other to produce the eventual partition of communities. This new vision ensures the important level of scalability, especially when dealing with big social networks.

The proposed approach can help explore social media to detect political communities and help predict political elections. Moreover, social networks are becoming the most important market in the world. The proposed approach may also be useful for business purposes. By detecting social network communities, this approach can be used to target a particular type of customers within social networks.

Our contributions can be summarized as follows:

(i) We introduced the concept of community cores and used pattern recognition techniques, represented in Growing Hierarchical Self-Organizing Maps (GHSOM) to detect them.

(ii) We coupled the genetic algorithms with Growing Hierarchical Self-OrganizingMaps (GHSOM) to extract the final communities. It is not a simple succession of steps. The genetic algorithm is tuned when working with the results of the Growing Hierarchical Self-Organizing Maps. This bias makes the genetic algorithm faster and more efficient.

The remainder of this paper is organized as follows. Section 2 reviewed the works related to community detection, while Section 3 introduced our approach and contributions. Section 4 was devoted to revealing the results achieved by our approach. The major conclusions were drawn in the final section before suggesting some perspectives for our future research work.

2. Related Works

Review papers [3, 4] classify these works into two perspectives: divisive and agglomerative approaches. Divisive approaches are top-down ones. They start with the entire graph, and they split it into partitions by removing edges. However, agglomerative approaches are bottom-up approaches; they start from vertices that will be gradually merged to build communities. The best partition is the one that maximizes a given metric. Most of the proposed works are agglomerative ones.

2.1. Divisive Approaches. In the divisive category, Girvan and Newman [5] proposed an approach based on the concept of edge betweenness defined as the number of shortest paths between pairs of nodes that run along that edge. Li et al. proposed [6] a divisive method also, namely local edge centrality (LEC) for community detection. In the first phase, a weight is computed for each edge. The authors relied on the node dissimilarity degree and edge betweenness. Nonimportant edges are deleted to obtain an initial partition of the network. After that, modularity optimization is used to get the final partition of the network.

2.2. Agglomerative Approaches

2.2.1. Modularity-Based Approaches. In 2004, Newman proposed the concept of modularity [7] used in many agglomerative works. It is a metric that has been widely used to characterize the partition quality and was used in Clauset et al. [8]. In their approach known as CNM, the authors start from lonely nodes, and the edges of the network are added progressively to increase the modularity. Blondel et al. [9] created the well-known Louvain method and also built their approach on modularity optimization. Neighbor nodes are grouped together through a repetitive step, and at each step, the modularity is computed to evaluate the achieved gain. Džamić et al. [10] proposed a community detection system that maximizes the modularity function to find the best partition. Hoffman et al. [11] use Cohen's similarity measure for categorical data. After that, the clustering is performed using k-means. The number of k-means clusters ranges from 2 to N (number of nodes). The best partition is the one that maximizes the modularity function.

2.2.2. Evolutionary-Based Approaches. There are approaches that used the genetic algorithms to optimize an objective function and to find the best community partition [12–18]. The most used objective function is the modularity of Newman [7]. Some of these approaches used more than one evaluation function; they are multiobjective [16]. Said et al. [19] proposed an approach that uses a genetic algorithm for detecting communities. The novelty suggested by the authors is a new way for generating the initial population and a new method for the mutation operation. The initial population is made up of nodes that have neighbors because putting isolated nodes in the solution space may increase the convergence time of the genetic algorithm. The mutation operation proposed by the authors is based on carrying out the operation on the selected solution and its neighbors.

Recently, Li et al. [20] proposed a community detection approach that uses attributes such as age, education background, hobby, and profession in addition to the topological structure. The community detection problem is transformed into a multicriteria optimization problem. To find the best community partition they used multiobjective genetic algorithms.

In [21], authors proposed an approach similar to genetic algorithms. They proposed an evolutionary method based on a fitness function and evaluating the quality of the partition using a fitness function. The authors also proposed new operators named vertex substitute operator and community substitute operator.

Qin et al. [22] worked in the same way as in [20]. They combined topology and content. An adaptive parameter is used to combine topology and content to effectively control the impact of content on community discovery. In the same way and using another evolutionary technique, Rostami et al. [23] proposed a particle swarm optimization-based multiobjective approach to detect central nodes in medical datasets. Ben Romdhane et al. [24] proposed the concept of purity and density of communities to define an objective function. They use the ant colony technique to realize the random walk and to optimize this objective function. In the same way, Majbouri et al. [25] used the ant colony to predict information diffusion paths. They study and model the propagation routes. They cluster nodes, and the final information diffusion paths are predicted using the ant colony.

In [26], Cai et al. proposed an approach based on multiagent systems. Single nodes are associated with agents. The agents affiliated with a similar cluster should gradually assemble in their common state space. The authors used the concept of consensus or quasi-consensus of the motions of dynamical systems to make the final clustering.

2.2.3. Label Propagation Algorithm-Based Approaches. Label propagation algorithm (LPA) [27–30] is also another interesting technique, considered as the fastest because it is a near-linear time algorithm.

The LPA uses only the network structure to guide the exploration process. They are well adapted for large-scale networks; they do not use a defined objective function nor ask for any preliminary information on the existing communities. The label assigned to each node depends on the labels assigned to the neighbor nodes. The main drawback of this technique is the fact that it does not provide a unique solution but an aggregate of many solutions.

In [31], authors used also the LPA to detect communities. They worked on complex attributed networks. From these networks, they develop a weighted graph. The weight of each node is computed using Laplacian centrality. The propagation of labels is proportional to the influence among the adjacent nodes. Nodes with higher influence in terms of structure and attributes update many tags. Community overlap propagation algorithm (COPRA) [32] is an overlapping community detection method derived from the label propagation algorithm. In the propagation process, the node label is determined based on labels of adjacent nodes, and hence, a node may belong to many communities.

2.2.4. Spectral Graph Partitioning Approaches. The spectral graph partitioning approaches are based on the eigenvectors of the Laplacian matrix. The eigenvector components with similar values represent the nodes that belong to the same community. In [33], Newman proposed the modularity matrix that is made up of the eigenvectors computed for the network. This enhancement leads to a spectral approach that returns better results than the classic modularity. Narantsatsral and Kang [34] proposed an agglomerative approach for social community detection. In this approach, the densely connected clusters are identified while agglomerating. Nodes are projected into an eigenvector space to be able to significantly distinguish between them.

2.2.5. Statistics-Based Approaches. Li et al. [35] proposed a Markov cluster approach known as MCL. It is based on simulations of using the concept of Markov chains to build a fast and scalable unsupervised Markov clustering algorithm. The order statistics local optimization method (OSLOM) [36] is an approach based on the local optimization of an evaluation function. The entire graph network is transformed into a network of subgraphs representing the communities. In addition, the OSLOM can detect overlapping communities.

2.2.6. Metric-Based Approaches. Rosvall and Bergstorm [37] introduced a random walk-based approach for detecting communities known as Infomap. They consider social networks as a set of regularities (patterns). Through a random walk, they try to detect these regularities by finding the best path that maximizes the compactness and minimizes information loss.

In [38], the authors proposed an approach based on thread-level parallelism for the calculation of adding qualified neighbor nodes to the community. This approach is performing overweighted networks in irregular topologies. Zardi et al. [39] proposed a hierarchical clustering. They define some metrics characterizing a good quality community partition. All these metrics are used to build an objective function that should be maximized. The nodes represent the initial communities, and they are merged progressively to detect the final community partition. C-Finder [40] is a local approach presented by Palla et al. whose main principle is to detect k-cliques inside the network. k-cliques mean small groups of k nodes that are totally linked. Two cliques may form a community if they are adjacent. Adjacent means they have at least $(k-1)$ common nodes. Communities are made by merging the adjacent k-cliques. In the same way, Zhang et al. [41] addressed the problem overlapping communities by detecting weak cliques and merging them. They proposed the Salton index to characterize node similarities, and the weak cliques detected were merged into larger communities, whenever possible. In [42], the authors proposed an original detection algorithm based on the fire propagation behavior. The approach works in two phases. The algorithm starts with a random node, and they simulate the effect of fire spread to aggregate nodes and constitute communities.

2.2.7. Influential Nodes Detection Based Approaches. The identification of the most influential node in social media networks has received a lot of attention in the data mining community. It has become a crucial step in the community detection approaches [43]. For instance, Chaabani and Akaichi [44] proposed an approach that operates in two steps. The first step aims at defining the communities and detecting the most important nodes in them. In the second step, the partition is defined, and the main communities are

detected. The authors also introduced a function that measures the strength of the links to define the communities.

3. Proposed Approach

3.1. Problem Formulation. The graphs were used to represent social networks. The graph's nodes represent the social actors, and its edges are the connections between the nodes. In our case, the social network is modeled as a graph $G = (V, E)$, where V is a set of nodes or vertices and E is a set of links or edges connecting two elements of V. To represent a graph, we use the adjacency matrix A. If the network is made by N nodes, the graph will be represented with the $N \times N$ adjacency matrix A, where the entry at position (i, j) is 1 if there is an edge from node i to node j, 0 otherwise. The row i of the adjacency matrix represents the features of the node.

3.2. Approach Overview. Different from the proposed works in the literature, we added a real step of initialization in our work. We did not tackle the community clustering directly. The first step consists in detecting the intrinsic agglomerations containing core nodes. After that, these agglomerations can be merged to generate the final communities (Figure 1).

3.3. Detection of Agglomerations. The first step involves identifying the skeleton of the communities called core nodes. Core nodes are generally nodes that are in the center of a community, and they are linked to most of the community nodes. We can distinguish them even without achieving community detection. They are generally located in a small remarkable agglomeration. The first step is to detect them. It gives a starting point better than single nodes to start detecting communities.

The main advantage of self-organizing maps is that they give an efficient way to explore unbalanced and complex structures. SOM provides a bidimensional visualization of multidimensional data.

Moreover, we can use the neighborhood property in self-organizing maps to have a better understanding of the relationships between agglomerations of nodes. We have used this tool in our previous works, in different contexts and the results were very encouraging [45, 46].

In the literature, there are works that use only SOM [18] to detect communities. The results found were not satisfactory. SOM cannot give the real borders of communities. They can only project input data on a bidimensional map. Moreover, in [47], a classical variant of SOM has been used to detect communities. This variant does not give a good scalability level and especially when dealing with big size social networks.

3.4. SOM and GHSOM. A self-organizing map is a set of connected neurons on which we map input elements represented by n-dimensional vectors $X = [x_1, x_2, \ldots, x_n]$ [48] (see Figure 2). The input elements are linked to the neurons through weights W_{ij} (see Figure 2). The neuron to which an input element is attached is called the winning neuron.

Self-organizing maps work as follows:

Step 1: The connection weights are randomly initialized.

Step 2: The winning neuron is calculated using the following formula:

$$j^* = \operatorname{argmin}_j \sum^{n} \left(X_i(t) - W_{ij}(t)\right)^2. \tag{1}$$

Step 3: The weights of the winning neuron and their neighbors are updated at every iteration as follows:

$$W_{ij}(t) = W_{ij}(t-1) + a(t)h_j(j^*, t)\left[X_i(t) - W_{ij}(t-1)\right], \tag{2}$$

where t represents the time, $a(t)$ is a variable decreasing with time, and $h(t)$ represents a neighborhood function. The principle is to reduce the influence when the neighborhood radius increases.

The main limitation of the classic SOM is its static architecture. The size of the map should be defined initially. For small problems, it can be used with no significant effects. However, when we deal with big and complex data, specifying the size of the map becomes very important, and its exploration becomes very difficult. For all these reasons, we used another variant of the SOM called the Growing Hierarchical Self-Organizing Maps GH-SOM [49]. These maps proved their effectiveness with big data problems [50].

The GHSOM represents more faithfully the input space by arranging it according to the shape of the data and its structure. It grows both in hierarchical and horizontal ways. Instead of representing all the input space by one SOM, the data are represented by multiple layers with a hierarchical structure, where each layer includes an independent SOM (see Figure 3). The training process starts with one layer (layer 0). It consists of one neuron only. The weights vector representing this neuron is the average value of the input vectors.

This vector is called $m_{01} = [w_{11}, w_{12}, \ldots, w_{1n}]$, where n is the dimension of the input space.

In layer 1, a map of 2×2 is created and randomly initialized. It is trained by the standard SOM learning algorithm (see formula (2)). The GHSOM growth strategy is based on the mean quantization error metric computed for each map by averaging the quantization errors of the neurons of the map as follows:

$$MQE_m = \frac{1}{u} \sum_i mqe_i, \tag{3}$$

where u refers to the number of units i contained in the SOM m.

The quantization error of the neuron i of the map is computed as follows:

$$mqe_i = \frac{1}{d} \sum_{j=1}^{d} m_i - X_j, \tag{4}$$

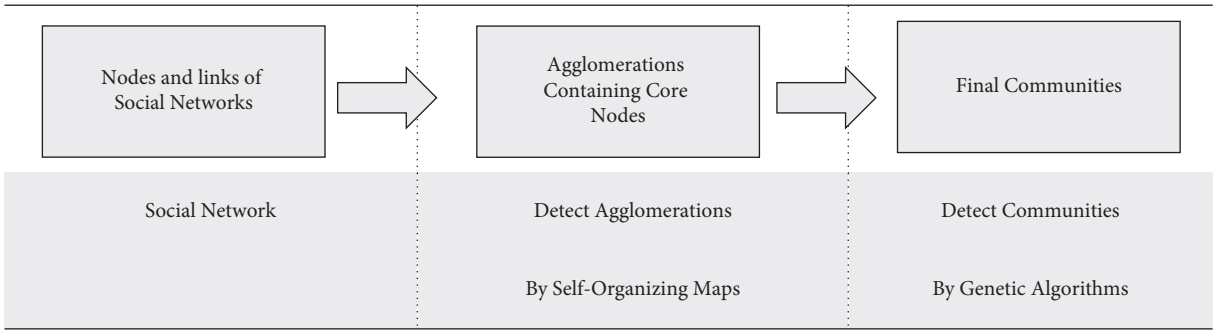


Figure 1: Community detection steps.

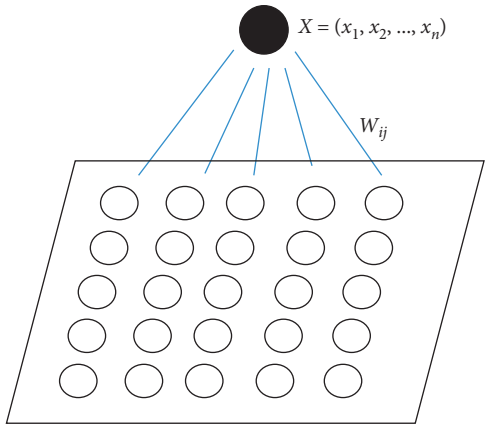


Figure 2: A classic self-organizing map layer; each neuron has four neighbors at most.

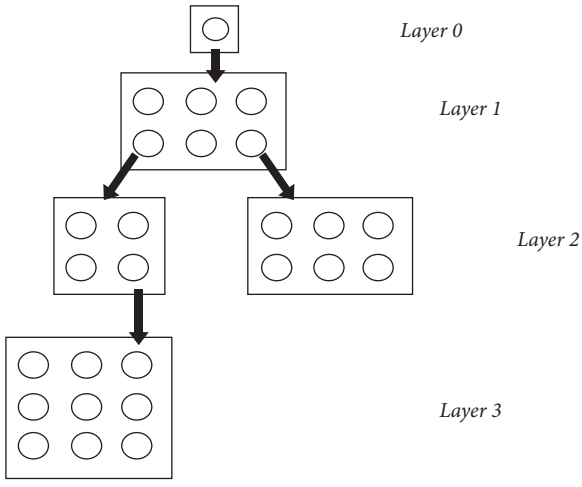


Figure 3: Growing hierarchical self-organizing map.

where m_i is the vector representing the neuron i, d represents the number of inputs having the neuron m_i as winning neuron. The main idea of the growing process in GHSOM is that each layer represents a deviation of the input data. In other words, the GHSOM will grow horizontally and vertically to reduce the deviation of the neuron of the previous layer to a given rate.

The criterion for the horizontal expansion is as follows:

$$MQE_m > \tau_1 \cdot MQE_0. \quad (5)$$

Hence, if this criterion is met on a specific neuron called "e" on a given map, a new map will be added to this neuron. The neurons' initial weights of this new map will be computed based on the weights of the neighbors of the neuron "e." The learning of the GHSOM and its expansion will continue until the two criteria are no longer satisfied.

Figure 4 shows a GHSOM used to detect teams' agglomerations inside the American football college data set. This data set is made up of 115 teams organized in 12 conferences. The edges correspond to matches played during the 2000 season. The objective is to retrieve the 12 conferences. As it can be noticed from Figure 4, the GHSOM output is interesting because it detects almost all the conferences. However, this is a specific case in which the data set is not big, and hence, the GHSOM provides good results on the first attempt. In our framework, the GHSOMs are mainly used to provide only the starting point for detecting the communities.

3.5. Community Detection. The detected agglomerations represent the skeleton of the future communities. They can be in communities themselves, or by merging them with other agglomerations, they form new communities. In the literature, there are many criteria that can be used to evaluate a community partition. The most known one is modularity. The modularity Q proposed by Girvan and Newman [7] is defined as follows:

$$Q = \sum_{c=1}^{n} \left[\frac{l_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right], \quad (6)$$

where n is the number of detected communities, l_c is the number of edges linking the nodes of the community, and l_c/m represents the percentage of links that join the same community. d_c is the total degree of nodes of C. The value of Q ranges between "−1" and "1." The value "1" means that we have a good network partition.

Maximizing the modularity means maximizing the two terms: $\sum_{c=1}^{n} l_c/m$ and $\sum_{c=1}^{n} (d_c/2m)^2$.

Maximizing the first term means having densely intraconnected communities, while maximizing the second term means having sparsely interconnected communities.

Genetic algorithms [51] are well-known for their global search capability. We used them in many previous works

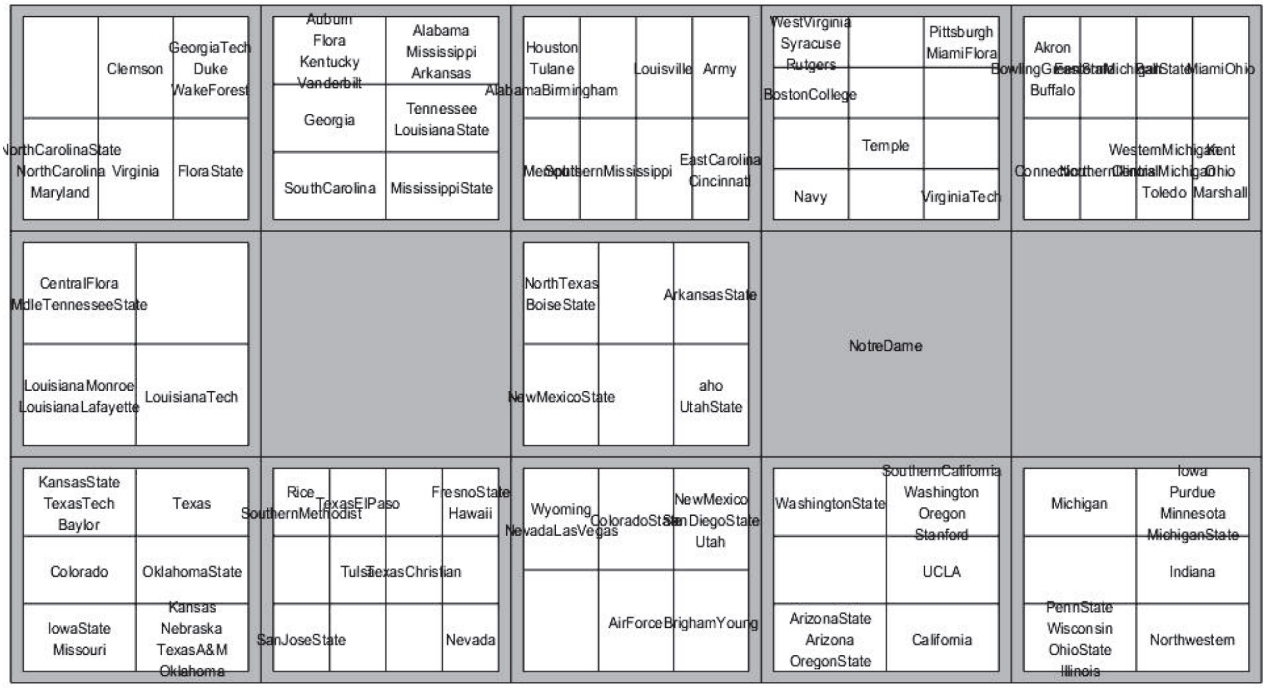


Figure 4: Detecting agglomerations of teams in American football college data set.

[52–54], and they proved their efficiency. In our approach, the exploration of solutions is guided by the GHSOM. In fact, it is absurd to put two agglomerations in the same community if they are not adjacent in the GHSOM. The proximity in the GHSOM means that the agglomerations share some common features and have strong relationships.

Genetic algorithms have also been widely used [12–17] in community detection. However, our proposal is different. Our contribution lies in the fact that the initial population and the genetic operations are made by considering the layout of agglomerations in the GHSOM.

The solution (the chromosome) is represented in the format of an integer array. Every gene of the array represents an agglomeration of the GHSOM. So, if the GHSOM detects N agglomerations, the chromosome will contain N genes that can take values ranging from "1" to N. If the j^{th} and the i^{th} gene have the same value, this means that i and j are in the same community. This representation is further explained by Figure 5. The network is made up of 7 nodes.

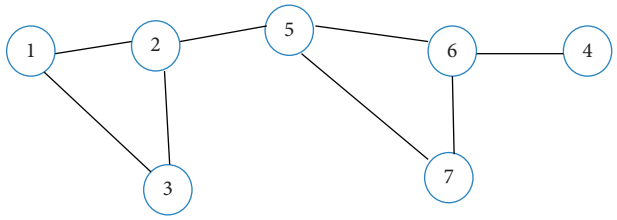
The network is made of 7 nodes that can be partitioned into two communities. The community partitioning may be represented by the chromosome $C_1 = \{1, 1, 1, 2, 2, 2, 2\}$ or the chromosome $C_2 = \{5, 5, 5, 6, 6, 6, 6\}$. The values of the genes, the community identifiers, do not have a real meaning they are simple labels.

3.5.1. Initialization. Creating an initial population consists in generating a set of chromosomes randomly initialized. Every gene of the chromosome is assigned to a random community identifier. However, as we have already evoked, the optimization process will be guided by GHSOM. So, when we initialize the chromosomes, only adjacent agglomerations could have the same community ID. This bias in the initial population makes the genetic algorithm converge faster and reduces the number of iterations.

3.5.2. Crossover. The goal is to make two new chromosomes called children. These children represent two new solutions that are added to the solution space hoping to increase the fitness function. However, this classic technique of crossover is not efficient for our encoding. In fact, the same community identifier in the two-parent chromosome may represent different communities. The crossover that we used is called one-way crossover and was introduced in [15]. Two chromosomes are selected: one is called the source, and the other is called the destination. From the source chromosome, we select one gene, and we look for the genes that have the same community ID. The community ID will be transferred from the source chromosome to the destination one by replacing the corresponding genes in the destination with the same community ID. Following this procedure, we are sure that the communities are faithfully transferred between chromosomes. Figure 6 shows an example of a crossover operation. The target community ID is 1.

3.5.3. Mutation. In our genetic algorithm, the mutation is performed by selecting one node and changing its community ID to another, respecting the GHSOM neighborhood principle.

3.6. Complexity of Our Approach. The complexity of the approach is crucial in social network community detection due to their large size. To evaluate the time of our approach,



Pos	1	2	3	4	5	7
C1	1	1	1	2	2	2
C2	5	5	5	6	6	6

Figure 5: Encoding of the genetic algorithm.

Source	1	1	3	4	4	6	6	5	8	1
Destination	2	3	4	4	1	5	7	8	8	2

Offspring	1	1	3	4	4	6	6	5	8	1
	1	1	4	4	1	5	7	8	8	1

Figure 6: Crossover operation.

we start by calculating the complexity of each step. The computational cost for GHSOM exhibits linear complexity. The processing time of a GHSOM is proportional to the social network size n. Therefore, the complexity is on the order of $O(n)$. A genetic algorithm's complexity is $O(g(nm + nm + n))$ with g the number of generations, n the population size, and m the size of the individuals. Therefore, the complexity is on the order of $O(gnm))$. Therefore, the complexity of our approach is $O(n)$

4. Experiments

We detailed, in this section, a set of experiments to show the efficiency of our approach. The used data sets are real and artificial social networks.

4.1. Real Networks. We tested our system on four real networks widely used in the literature. The data set includes Zachary's network of karate club [55], Lusseau's network of bottlenose dolphins [56], the American college football network [5], and the political books network [7] as displayed in Table 1.

4.1.1. Zachary's Karate Club. This network was made by Zachary. He studied the behavior of 34 members of a karate club for 2 years. He constructed a network of friendships between the members of the club, using a variety of measures to estimate the strength of ties between individuals [55]. He identified 2 communities of friendship in his network as plotted in Figure 7.

4.1.2. Bottlenose Dolphins Network. The bottlenose dolphins network is achieved on a study on 62 bottlenose dolphins, living in New Zealand. The study was made by Lusseau [56]. The nodes are dolphins, and the edges are relationships observed among the dolphins. The relationships are established by observation of statistically significant frequent associations. The number of edges in this network is 159. Two communities are clearly identified as displayed in Figure 8.

4.1.3. American College Football Network. The American college football network comes from the United States college football [5]. The data set is made up of 115 teams organized in 12 conferences (see Figure 9). The 616 team edges correspond to matches played by the teams against each other during the regular season of the fall of 2000.

4.1.4. Political Books Network. In this network, nodes represent political books published in 2004 and that are purchased online through the site Amazon.com [7]. Two books are connected by an edge if they were frequently purchased together. The network is made up of four communities (see Figure 10). The number of nodes (books) is 105, and the number of edges is 441.

4.1.5. Performance and Comparison Results. We compared our work with the Agrawal approach [12], SOMSN [47], MeanCD [44], Infomap [37], and CNM [8]. These approaches were selected for the following reasons. First, we compared our approach with Agrawal [12] because it used the genetic algorithms, and we used its genetic operators. The goal was to see the GHSOM contribution when used in conjunction with genetic algorithms. Second, we compared our approach with SOMSN [47] because it is the only approach that uses a self-organizing map to detect communities. Third, we compared our approach with MeanCD, Infomap, and CNM [8, 37, 44] because they are well known for their good performance for their performance (results and time execution). MeanCD is a recent approach and is based on influential node detection like our approach. Finally, for all these approaches, we have either the results that they achieved on the real networks mentioned above or their source code.

Table 1: Results on real networks.

Network	Nodes/edges	SOMG	Infomap	CNM	GA	SOMSN	MeanCD
Zachary karate club	34/77	0.40/	0.38	0.38	0.38	0.28	0.4
American college football	115/615	0.61	0.55	0.59	0.50	0.45	0.56
Bottlenose dolphins	62/159	0.46	0.44	0.46	0.42	0.36	0.42
Political books	105/441	0.5	0.46	0.5	-	0.44	0.48

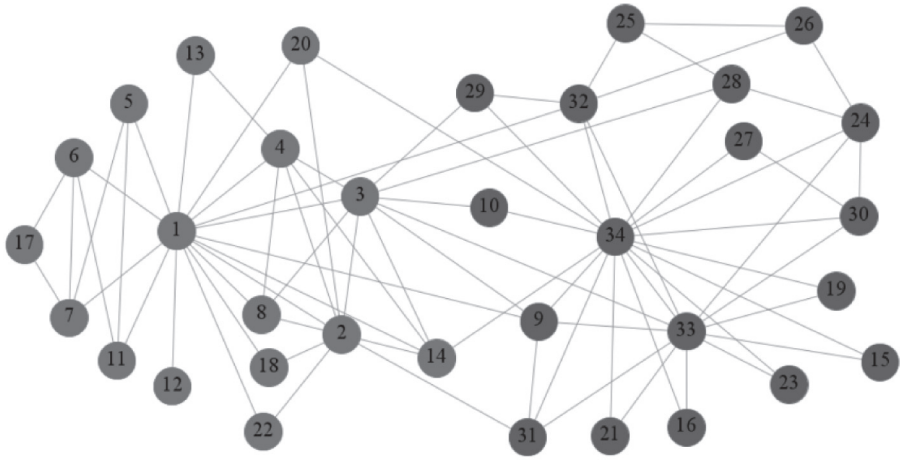


Figure 7: Zachary karate club network.

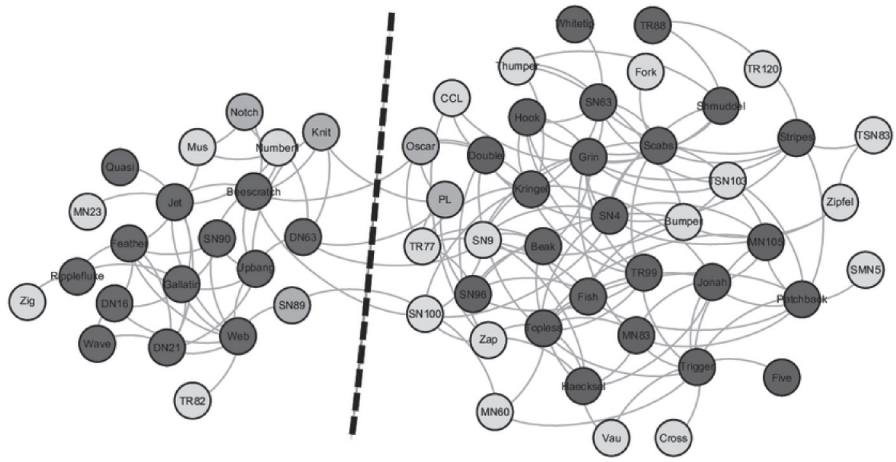


Figure 8: Bottlenose dolphins network.

The modularity value was used to evaluate the performance of each system. The results of this comparison are displayed in Table 1. Our system is denoted by SOMG.

When examining the obtained results, our work, MeanCD [44], and CNM [8] performed better than the others. Moreover, when we focus on the results obtained by the SOMSN system, we can conclude that using only self-organizing maps cannot generate good community partitions. Indeed, self-organizing maps can give the morphology of the communities, the skeleton of the communities, but not the whole communities' structures.

The results of genetic algorithms as implemented by Agrawal [12] are interesting. However, they did not perform as good as SOMG. When using the classical genetic algorithms implementation, the initialization and the genetic operations are achieved without any considerations of the structure of the social network. The process is completely random. The quality of the obtained solutions will be impacted by the initialization.

CNM and Infomap [8, 37] detect communities starting from lonely nodes through achieving progressive node clustering. The clustering should increase the modularity. Although Infomap and CNM have an oriented clustering process, our approach performed better than them in all the social networks. In fact, the use of GHSOM in the first step to making initialization of communities made the community

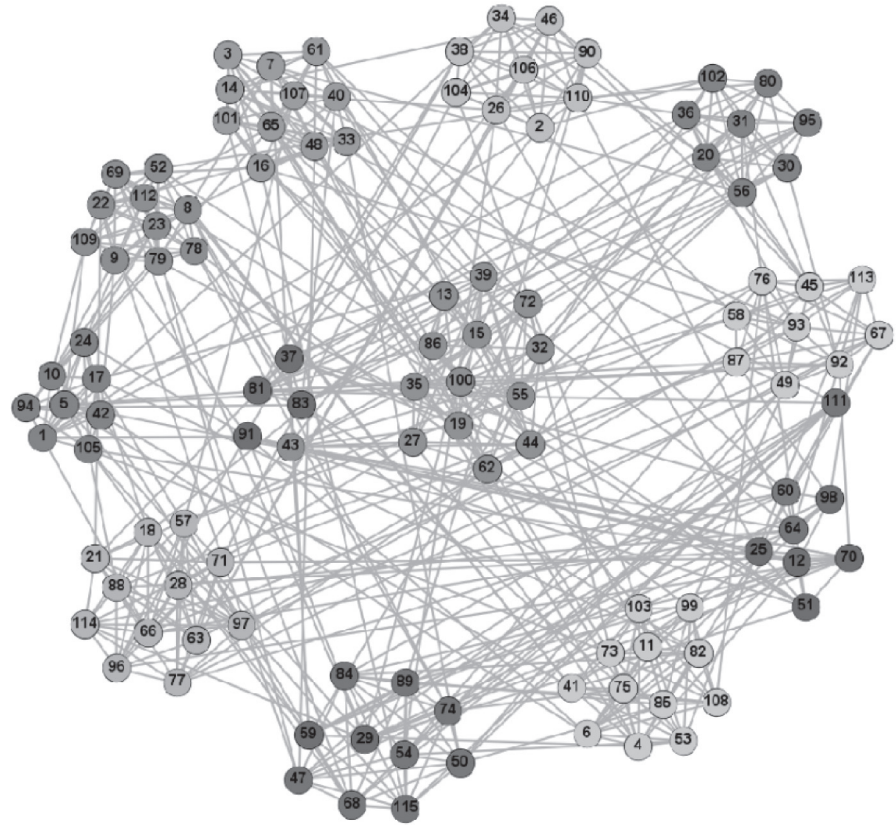


Figure 9: The American college football network.

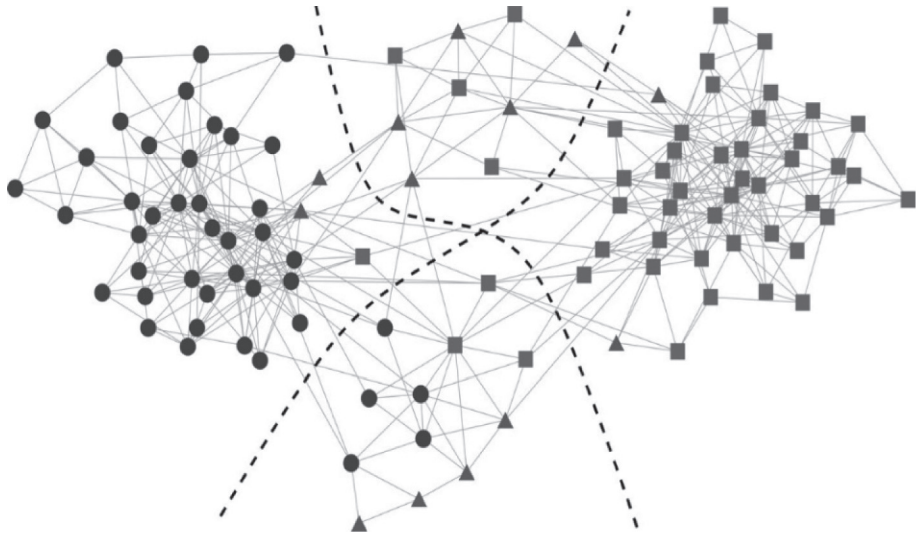


Figure 10: Political books network.

partitioning more efficient. The advantage of GHSOMs is the fact that they provide reliable initial partitioning. In fact, nodes located on the same neurons on the map certainly belong to the same community. This is an intrinsic property of GHSOM. GHSOM preserves the topology of social networks. The mapping of social networks preserves the relative distance between nodes. Nodes that are close to each other in the social network are mapped to adjacent neurons in GHSOM.

The MeanCD also operates in two steps like our approach SOMG; yet SOMG performs better. In SOMG, the detection of agglomerations is based on pattern recognition techniques. However, in MeanCD, the detection of agglomerations is based on measures computed on node pairs.

Detecting agglomerations must be relatively achieved to all the nodes and not through finding direct similarities between them two by two. This may lead to oversegmentation.

According to Table 2, SOMG, Infomap, and CNM performed clearly better in terms of execution time than the other approaches. Infomap is the best because it is based on a random walk. This makes the complexity of the Infomap approach nearly linear and makes it quicker.

4.2. Artificial Networks. Our work, MeanCD, and the CNM, which made the best performance in real networks, have been tested on the LFR benchmark (Table 3). This benchmark was developed by Lancichinetti et al. [57].

In this study, the authors created a software that generates a graph with a customized structure. The purpose was to compare the three works on big size networks having different structure features. Among these features, we mention the number of nodes N; the average degree of incoming edges k; the maximum degree of the incoming edges max k; the fraction between incoming and outgoing edges inside a community; the minimal community size min c and the maximal community size max c; and the mix parameter μ which controls the fraction of edges between communities. An important value of μ corresponds to a network with a blurred community structure. This experiment allowed us to test the scalability of our approach.

To measure the performance of the two systems, we used the NMI measure instated of modularity. In fact, contrary to real networks, artificial ones have ground-truth partitions. For this reason, we used the normalized mutual information (NMI) proposed by Danon et al. [58]. The NMI value helps to compare an obtained partition A and the ground-truth partition B. When reading the formula, we can notice that when partitions A and B are totally independent, the NMI value will be 0. However, if they are matching, the NMI value will be 1.

$$\mathrm{NMI}(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} N_{ij}\log\left(N_{ij}N/N_{i.}N_{.j}\right)}{\sum_{i=1}^{C_A} N_{i.}\log\left(N_i/N\right) + \sum_{i=1}^{C_B} N_{.j}\log\left(N_j/N\right)}. \tag{7}$$

4.2.1. Small Networks and Small Communities. In the first test, we targeted the case of small networks (important number of nodes) and small communities (small number of nodes per community). We fixed the parameters as follows: the number of nodes N is set to 1,000, the community size $C \epsilon$ [50–100], and the average degree of nodes K is set to 25. The mix parameter μ ranges from 0.1 to 1. Results are displayed in Figure 11.

4.2.2. Small Networks and Large Communities. In the second test, we targeted the case of small networks (total number of nodes) and large communities (number of nodes per community). We fixed the parameters as follows: the number of vertices N is set to 1,000, the community size $C \epsilon$ [100–250], and the average degree of nodes K is set to 25. The mix parameter μ ranges from 0.1 to 1. Results are displayed in Figure 12.

4.2.3. Large Network and Small Communities. In the third test, we targeted the case of large networks (important number of nodes) and small communities (a small number of nodes per community). We fixed the parameters as follows: the number of vertices N is set to 10,000, the community size $C \epsilon$ [50–100], and the average degree of nodes K is set to 25. The mix parameter μ ranges from 0.1 to 1. Results are displayed in Figure 13.

4.2.4. Large Network and Large Communities. In the fourth test, we targeted the case of large networks (important number of nodes) and large communities (important number of nodes per community). We fixed the parameters as follows: the number of vertices N is set to 10,000, the community size $C \epsilon$ [100–250], and the average degree of nodes K is set to 25. The mix parameter μ ranges from 0.1 to 1. Results are displayed in Figure 14.

4.2.5. Performance and Comparison Results. As shown in Figures 11–14, the NMI value is decreasing with the increase of the mixing parameter μ for all works. This is not surprising because it is easier for each system to detect communities in a social network with a clear community structure. In fact, when the mixing parameter becomes bigger, the structure of the network becomes blurred, and the communities become hardly distinguishable. However, it can also be noticed that for all the tests, the CNM and MeanCD performance decreases faster than that of our system. This proves that our system is less sensitive to blurring. Our system can detect the communities' borders better than both the CNM and MeanCD. However, we consider that the blurred networks remain a real limitation in our approach and need more investigation on the fitness function. The actual fitness function is suitable for distinguishable communities rather than overlapping communities.

In the case of small networks and large communities, all three works achieved comparable results. In this case and when the communities are distinguishable, we mean the mixing parameter is less than 0.6, retrieving them is not a complex task. However, when the communities become smaller and precision becomes crucial, our approach performs clearly better especially for the values of mixing parameter ranging from 0.3 to 0.6.

The step of detecting agglomerations made by the GHSOM provided a considerable contribution to discovering at least the cores of communities. On the contrary, the CNM failed in detecting the essential part of each community and provided under segmented communities. The MeanCD performed better, and this is due to the initialization phase.

The results obtained for large networks are coherent with those obtained with the small ones. When the size of the

Table 2: Execution time in seconds on real networks.

Network	Nodes/edges	SOMG	Infomap	CNM	GA	SOMSN	MeanCD
Zachary karate club	34/77	10.2	2.5	5.6	25.3	15	13.85
American college football	115/615	15.36	3.52	7.58	38.69	27.58	28.59
Bottlenose dolphins	62/159	11.23	2.63	8.64	20.48	17.7	19.8
Political books	105/441	10.41	3.57	8.2	45.25	32.14	29.15

Table 3: Synthetic networks.

Network	Total number of nodes	Number of nodes per community
Small networks and small communities	1,000	[50–100]
Small networks and large communities	1,000	[100–250]
Large network and small communities	10,000	[50–100]
Large network and large communities	10,000	[100–250]

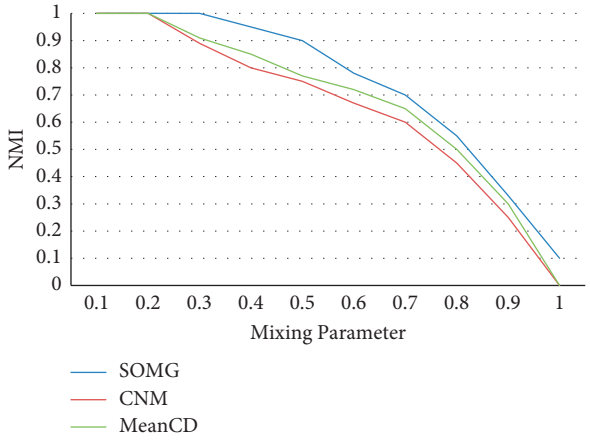


Figure 11: Comparison of the NMI values with different values of the mixing parameter of our system and CNM for the case of small networks and small communities.

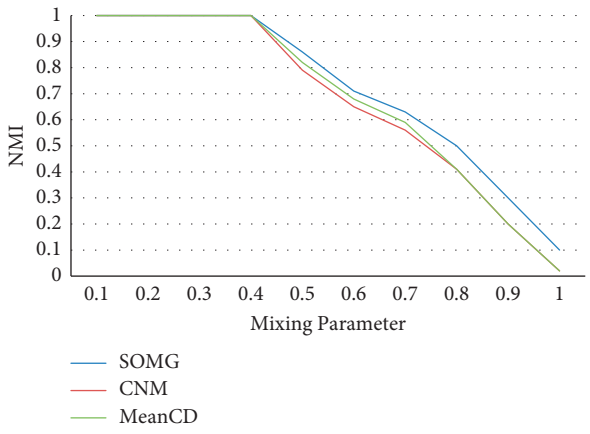


Figure 12: Comparison of the NMI values with different values of the mixing parameter of our system and CNM for the case of small networks and large communities.

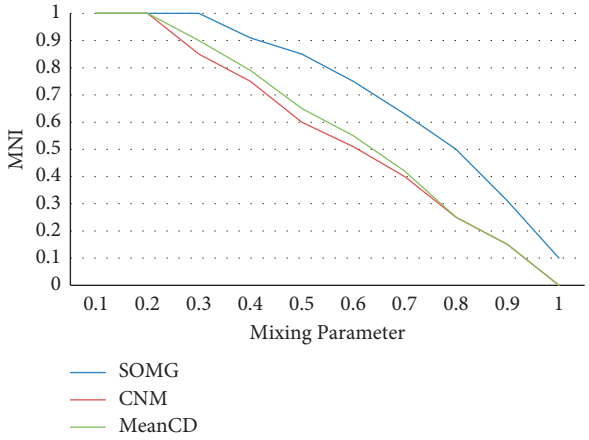


Figure 13: Comparison of the NMI values with different values of the mixing parameter of our system and CNM for the case of large networks and small communities.

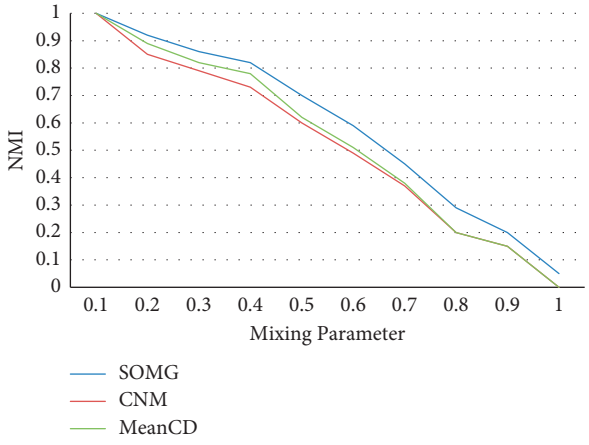


Figure 14: Comparison of the NMI values with different values of the mixing parameter of our system and CNM for the case of large networks and large communities.

community decreases, our system performs better. The gap in performance becomes even more important. For the values of the mixing parameter ranging from 0.3 to 0.8, we performed clearly better. However, it is worth noticing that for all works, the performance decreases faster when the size of the network becomes important.

Table 4: Average execution time in seconds.

Network	SOMG	CNM	MeanCD
Small networks and small communities	80	69	120
Small networks and large communities	115	90	190
Large network and small communities	300	270	450
Large network and large communities	420	350	560

CNM performed slightly better than our approach in terms of execution time (Table 4). This is due to the simplicity of the CNM approach that starts from lonely nodes, and the edges of the network are added progressively to increase the modularity. We have to improve our performance in this criterion. To reduce the time execution we can parallelize the execution of a genetic algorithm. A parallel genetic algorithm is an algorithm that uses multiple genetic algorithms to solve a single task. All these algorithms try to solve the same task, and after they have completed their job, the best individual of every algorithm is selected.

5. Conclusion

A two-stage system to detect communities inside social networks was proposed in this paper. The main idea of our approach was to start by detecting cores of communities and after that refining them to detect the final communities. The experimental results on real and artificial networks showed that starting by detecting community cores has an important contribution. In fact, the two stages proposed in our system are complementary. The first stage, which consists in detecting cores of communities through the GHSOM, was aimed at providing good initial conditions for the whole process. However, the second was a refining stage in which the genetic algorithms detected the final communities through an oriented process. The obtained results are encouraging and should stimulate future research. The overlapping communities and blurred networks constitute our first focus. The second focus will be parallelizing the execution of genetic algorithms to reduce the execution time.

Data Availability

The test data used in this study have been taken from the website (https://www-personal.umich.edu/~mejn/netdata/). The implementation of self-organizing map that we used can be downloaded from https://ifs.tuwien.ac.at/~andi/ghsom/. The implementation of genetic algorithms that we used can be downloaded from https://www.mathworks.com/matlabcentral/fileexchange/39021-basic-genetic-algorithm.

Conflicts of Interest

The author declares that he has no conflicts of interest.

Acknowledgments

The author would like to acknowledge the financial support for this work by grants from the General Direction of Scientific Research and Technological Renovation (DGRSRT), Tunisia, under the PEJC program.

References

[1] P. Bedi and C. Sharma, "Community detection in social networks," *Data Mining and Knowledge Discovery*, vol. 6, pp. 15–135, 2016.

[2] S. Harenberg, G. Bello, L. Gjeltema et al., "Community detection in large-scale networks: a survey and empirical evaluation," *WIREs Computational Statistics*, vol. 6, no. 6, pp. 426–439, 2014.

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

[4] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: a survey," *Physics Reports*, vol. 533, no. 4, pp. 95–142, 2013.

[5] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[6] X. Li, S. Zhou, J. Liu, G. Lian, G. Chen, and C.-W. Lin, "Communities detection in social network based on local edge centrality," *Physica A: Statistical Mechanics and its Applications*, vol. 531, Article ID 121552, 2019.

[7] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, Article ID 066133, 2004.

[8] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 70, Article ID 066111, 2004.

[9] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 10, p. 1000, 2008.

[10] D. Džamić, D. Aloise, and N. Mladenović, "Ascent–descent variable neighborhood decomposition search for community detection by modularity maximization," *Annals of Operations Research*, vol. 272, pp. 273–287, 2019.

[11] M. Hoffman, D. Steinley, K. Gates, M. Prinstein, and M. Brusco, "Detecting clusters/communities in social networks," *Multivariate Behavioral Research*, vol. 53, pp. 1–17, 2017.

[12] R. Agrawal, "Bi-objective community detection (BOCD) in networks using genetic algorithm," *Communications in Computer and Information Science*, vol. 168, pp. 5–15, 2011.

[13] C. Pizzuti, "GA-net A Genetic algorithm for community detection in social networks," in *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature*, pp. 1081–1090, Dortmund, Germany, September 2008.

[14] G. Chen, Y. Wang, and J. Wei, "A new multi-objective evolutionary algorithm for community detection in dynamic complex networks," *Mathematical Problems in Engineering*, vol. 2013, Article ID 161670, 2013.

[15] M. Tasgin, A. Herdagdelen, and H. Bingol, "Community detection in complex networks using genetic algorithms," 2007, https://arxiv.org/abs/0711.0491.

[16] P. Wu and L. Pan, "Multi-objective community detection based on memetic algorithm," *PLoS One*, vol. 10, Article ID e0126845, 2015.

[17] A. I. Hafez, N. I. Ghali, A. E. Hassanien, and A. A. Fahmy, "Genetic algorithms for community detection in social networks," in *Proceedings of the International Conference on, Intelligent Systems Design and Applications*, pp. 460–465, Kochi, India, November 2012.

[18] M. Rostami, K. Berahmand, and S. Forouzandeh, "A novel community detection based genetic algorithm for feature selection," *Journal of Big Data*, vol. 8, 2021.

[19] A. Said, R. A. Abbasi, O. Maqbool, A. Daud, and N. R. Aljohani, "CC-GA: a clustering coefficient based genetic algorithm for detecting communities in social networks," *Applied Soft Computing*, vol. 63, pp. 59–70, 2018.

[20] Z. Li, J. Liu, and K. Wu, "A multiobjective evolutionary algorithm based on structural and attribute similarities for community detection in attributed networks," *IEEE Transactions on Cybernetics*, vol. 48, no. 7, pp. 1963–1976, 2018.

[21] X. Zhou, K. Yang, Y. Xie, C. Yang, and T. Huang, "A novel modularity-based discrete state transition algorithm for community detection in networks," *Neurocomputing*, vol. 334, pp. 89–99, 2019.

[22] M. Qin, D. Jin, K. Lei, B. Gabrys, and K. Musial-Gabrys, "Adaptive community detection incorporating topology and content in social networks," *Knowledge-Based Systems*, vol. 161, pp. 342–356, 2018.

[23] M. Rostami, S. Forouzandeh, K. Berahmand, and M. Soltani, "Integration of multi-objective PSO based feature selection and node centrality for medical datasets," *Genomics*, vol. 112, no. 6, pp. 4370–4384, 2020.

[24] L. Ben Romdhane, Y. Chaabani, and H. Zardi, "A robust ant colony optimization-based algorithm for community mining in large scale oriented social graphs," *Expert Systems with Applications*, vol. 40, no. 14, pp. 5709–5718, 2013.

[25] K. Majbouri Yazdi, A. Majbouri Yazdi, S. Khodayi et al., "Prediction optimization of diffusion paths in social networks using integration of ant colony and densest subgraph algorithms," *Journal of High Speed Networks*, vol. 26, no. 2, pp. 141–153, 2020.

[26] N. Cai, C. Diao, and M. J. Khan, "A novel clustering method based on quasi-consensus motions of dynamical multiagent systems," *Complexity*, vol. 8, 2017.

[27] J. Xie and B. K. Szymanski, "LabelRank: a stabilized label propagation algorithm for community detection in networks," in *Proceedings of the IEEE Network Science Workshop*, pp. 138–143, West Point, NY, USA, May 2013.

[28] E. Ferrara, "Community structure discovery in Facebook," *International Journal of Social Network Mining*, vol. 1, no. 1, pp. 67–90, 2012.

[29] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 76, Article ID 036106, 2007.

[30] M. Asadi and F. Ghaderi, "Incremental community detection in social networks using label propagation method," in *Proceedings of the FRUCT Association*, pp. 39–47, Jyvaskyla, Finland, May 2018.

[31] K. Berahmand, S. Haghani, M. Rostami, and Y. Li, "A new attributed graph clustering by using label propagation in complex networks," *Journal of King Saud University of Computer Information Science*, 2020.

[32] S. Gregory, "Finding overlapping communities in networks by label propagation," *New Journal of Physics*, vol. 12, 2010.

[33] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 74, Article ID 036104, 2006.

[34] U.-U. Narantsatsral and S. Kang, "Social network community detection using agglomerative spectral clustering," *Complexity*, vol. 2017, Article ID 3719428, 10 pages, 2017.

[35] H. J. Li, Y. Wang, L. Y. Wu, J. Zhang, and X. S. Zhang, "Potts model based on a Markov process computation solves the community structure problem effectively," *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 86, Article ID 016109, 2012.

[36] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, "Finding statistically significant communities in networks," *PLoS One*, vol. 6, Article ID e18961, 2011.

[37] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[38] S. Souravlas, A. Sifaleras, and S. Katsavounis, "A parallel algorithm for community detection in social networks, based on path analysis and threaded binary trees," *IEEE Access*, vol. 7, pp. 20499–20519, 2019.

[39] H. Zardi and L. B. Romdhane, "An o(n2) algorithm for detecting communities of unbalanced sizes in large scale social networks," *Knowledge-Based Systems*, vol. 37, pp. 19–36, 2013.

[40] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[41] X. Zhang, C. Wang, Y. Su, L. Pan, and H.-F. Zhang, "A fast overlapping community detection algorithm based on weak cliques for large-scale networks," *IEEE Transactions on Computational Social Systems*, vol. 4, no. 4, pp. 218–230, 2017.

[42] H. S. Pattanayak, A. L. Sangal, and H. K. Verma, "Community detection in social networks based on fire propagation," *Swarm and Evolutionary Computation*, vol. 44, pp. 31–48, 2019.

[43] M. Azaouzi and L. B. Romdhane, "An efficient two-phase model for computing influential nodes in social networks using social actions," *Journal of Computer Science and Technology*, vol. 33, no. 2, pp. 286–304, 2018.

[44] Y. Chaabani and J. Akaichi, "Meaningful communities detection in medias network," *Social Network Analysis and Mining*, vol. 7, pp. 1–11, 2017.

[45] T. Ayadi, M. Ellouze, T. M. Hamdani, and A. M. Alimi, "Movie scenes detection with MIGSOM based on shots semi-supervised clustering," *Neural Computing and Applications*, vol. 22, no. 7-8, pp. 1387–1396, 2013.

[46] M. Ellouze, N. Boujemaa, and A. M. Alimi, "Scene pathfinder: unsupervised clustering techniques for movie scenes extraction," *Multimedia Tools and Applications*, vol. 47, no. 2, pp. 325–346, 2010.

[47] F. Ghaemmaghami and R. Manouchehri Sarhadi, "SOMSN: an effective self organizing map for clustering of social networks," *International Journal of Computer Applications*, vol. 84, no. 5, pp. 7–12, 2013.

[48] T. Kohonen, "The self-organizing map," in *Proceedings of the IEEE*, pp. 464–1480, 1990.

[49] A. Rauber, D. Merkl, and M. Dittenbach, "The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data," *IEEE Transactions on Neural Networks*, vol. 13, no. 6, pp. 1331–1341, 2002.

[50] A. Rauber, M. Dittenbach, and D. Merkl, "Automatically detecting and organizing documents into topic hierarchies: a neural network based approach to bookshelf creation and arrangement," in *Proceedings of the European Conference on Research and Advanced Technologies for Digital Libraries*, pp. 348–351, Lisbon, Portugal, September 2000.

[51] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Longman, Boston, MA, USA, 1st edition, 1989.

[52] M. Ellouze, N. Boujemaa, and A. M. Alimi, "IM(S)2: interactive movie summarization system," *Journal of Visual Communication and Image Representation*, vol. 21, no. 4, pp. 283–294, 2010.

[53] M. Ellouze, H. Karray, and A. M. Alimi, "Genetic algorithm for summarizing news stories," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, pp. 303–308, Spain, Barcelona, March 2006.

[54] M. Ellouze, H. Karray, and A. M. Alimi, "REGIM, research group on intelligent machines, Tunisia, at TRECVID 2008, BBC rushes summarization," in *Proceedings of the International Conference ACM Multimedia, TRECVID BBC Rushes Summarization Workshop*, New York, NY, USA, 2008.

[55] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.

[56] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.

[57] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 78, Article ID 046110, 2008.

[58] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, no. 9, Article ID 09008, 2005.