

Community Detection in Dynamic Social Networks: A Multi-Agent System based on Electric Field

E.A Abdulkreem¹, H. Karamti³

CCIS Department
Princess Nourah bint Abdulrahman University
Riyadh, Saudi Arabia

H. Zardi²

CS Department
ISIMA, Monastir University
Monastir, Tunisia

Abstract—In recent years, several approaches have been proposed in order to detect communities in social networks. Most of them suffer from the recurrent problems: no detection of overlapping communities, exponential running time, no detection of all possible communities transformations, don't consider the properties of social members, inability to deal with large scale networks, etc. Multi-agent systems are very suitable for modeling the phenomena in which various autonomous entities in interactions able to evolve in a dynamic environment. Considering the advantages of multi-agent simulations for social networks, in the present study, an incremental multi-agent system based on electric field is proposed. In this approach, a group of autonomous agents work together to discover the dynamic communities. Indeed, an agent is associated to each detected community. To update its community according to the dynamic of its members, each agent creates an electric field around it. It applies an attractive force to add very connected and similar members and neighboring communities. In the same time, it applies a repulsive force to reject some members and to get away from other communities. These forces are based on the structural and attributes similarity. To study the performance of this approach, set of different experiments is performed. The obtained results show the efficiency of the proposed model that was able to overcome all mentioned problems.

Keywords—Community detection; dynamic social networks; network evolution; multi-agent system; electric field; attractive force; repulsive force; attributes similarity; overlapping communities

I. INTRODUCTION

Since their introduction, social networks sites such as Facebook, Instagram and Google+ have attracted millions of users, many of whom have integrated these sites into their daily practices. These networks have recently become an important research topic that attracts more and more scientists.

A social network is a dynamic set of members and their relationships. New members appear, existing ones disappear, new relations appear every day and old relations weaken gradually, disappear or on the contrary, they reinforce.

Moreover, social networks are characterized by actors that divide up naturally into groups called “communities”. Conventionally, a community is defined as a group of users who interact with each other more frequently than with those outside the group, and that share similar topics of interest [1], [2], [3]. Since social networks are usually modeled by a graph such as nodes represent social actors and edges represent the relationships, a community is defined as a set of nodes that are densely connected among themselves and sparsely connected to the rest of nodes. These communities evolve over time

according to the evolution of the actors and their interactions. Fig. 1 illustrates the basic communities transformations that have been identified in a number of studies (see for example [4], [5]): growth, contraction, splitting into many communities, merging of many communities to one, birth and death of communities.

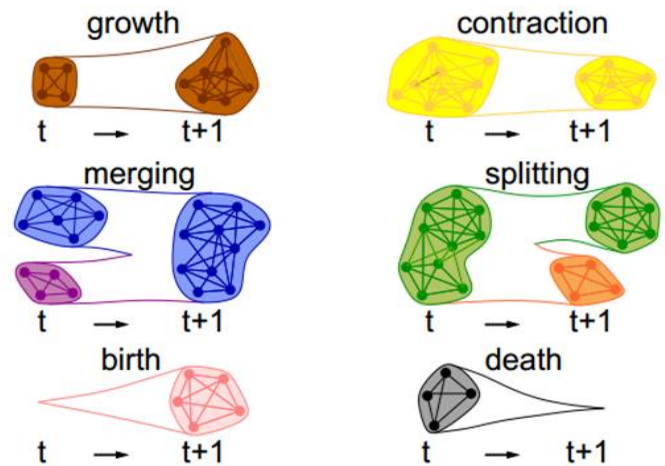


Fig. 1. Basic transformations of community structure [4]

The automatic detection of these dynamic communities provides a basis for studying the emerging phenomena within the network: it allows to determine the number and sizes of the communities and to track their evolution over time. It is also used to recommend establishing new relations that is a service frequently offered in social networks like Facebook. In addition, community detection algorithms can be used to develop other more complex processes such as network visualization.

The first problem confronted here is the complexity of this problem which is NP-hard [6]. The second obstacle is intrinsically linked to the absence of an exact definition of the dynamic community notion: should it be defined as a succession of static communities or as a set of communities that evolve over time?

Several research studies are deal with the community detection problem. In the second section, the main approaches proposed in the literature for the detection of dynamic communities are presented. In the third section, the proposed approach is detailed. In the fourth section, the experimental results are

presented before concluding this paper.

II. RELATED WORK

Community detection, also known as graph clustering, has been extensively studied in the literature [7]. The first approaches consider only a static view of the network; they study a snapshot G_t at a particular time t [8], [9], [2], [10], [11], [12], [13], [14], [15]. In this way, the evolutionary information of the network and its communities is lost because real-world networks are always evolving, either by adding or removing nodes or edges over time.

Recently, some approaches for the detection of dynamic communities have been proposed [16], [17], [18], [19], [20], [5], [21], [22], [23], [24]. One way to analyze communities in an evolving network is to consider the dynamic graph as a succession of independent captures of the graph, all of which are static graphs. These approaches consist in applying a static community detection algorithm on each snapshot of the network and then finding a correspondence between the detected communities in the consecutive captures [4], [25], [26]. For example, in [16], authors detect dynamic communities by optimizing a quality function which considers both the quality and the stability of communities. While in [20], authors fit the evolving network to a dynamic version of the stochastic block model, and determine the community assignment by estimating the parameters of the model.

One of the problems of these approaches is the instability of the static community detection methods [27]: these methods can give different results for similar networks. Also, track the evolution of communities, over a set of independent time snapshots, leads to the loss of information related to the evolution of network. The main disadvantage of these algorithms is that they are commonly time-consuming when the network evolves rapidly and the time slices are extremely small, i.e., the network has a lot of snapshots to be computed.

Another set of proposed approaches update adaptively the current community structure based on the previous ones according to the network modifications. So the evolution of the network is no longer considered as a succession of snapshots, but as a succession of modifications on the network. The idea is to start with an initial partition and to update it according to the latest evolution of the network instead of finding a new partition [28], [29], [23], [24]. The detection of communities is therefore not done on the whole network, but only by minor and successive local modifications. For example, Nguyen NP et al. [30] proposed a modularity-based algorithm which greedily changes memberships of nodes by optimizing a local modularity function whenever a small modification occurs in the network. In a similar way, LabelRankT [21] and ALPA [23] adjust its detecting communities according to the network modifications through a stabilized label propagation process by taking advantage of what is already obtained in the previous snapshots.

Although these approaches appear to be much more relevant than computing communities on each snapshot separately, they are not able to deal with large networks. This failure is mainly due to their centralized nature.

Some multi-agents approaches are also proposed for the discover of communities in dynamic social networks [31], [5],

[32], [33], [34]. These approaches seem to be most appropriate for observing the network evolution and updating communities locally and they show interesting results thanks to their decentralized aspect (see Section III). However, these approaches suffer from several shortcomings. The main problem is the disability to detect all possible transformations of communities, specially complex evolutions such as splitting and merging. Another problem of these approaches is the use only of network structural information for the identification of the dynamic communities and ignore the characteristics of social actors (age, education, city of residence, etc.). However, these properties are often very important data for the detection of communities. Really, in the ideal partition, communities must have members that are not only highly connected but also have similar properties (i.e., attributes). In this case, the generated communities will have, on the one hand, a cohesive intra-community structure and, on the other hand, homogeneous nodes.

III. DESCRIPTION OF THE MULTI-AGENT PROPOSED APPROACH

A Multi-Agent System (MAS) is defined as a system in which several autonomous and intelligent entities, called agents, interact together to achieve a set of goals or tasks. Multi-agent systems are very suitable for modeling the phenomena in which the interactions between various entities are complex. The power of expression of multi-agent models allows to represent autonomous entities in interactions and able to evolve in a dynamic environment [35], [36]. Considering the advantages of multi-agent simulations for social networks, a multi-agent framework allowing to detect the dynamic social network is proposed. In this section, we start by the problem definition. Subsequently, a proposed model description is given.

A. Problem Definition

The social network is modeled by an attributed graph $G = (V, E, A)$ such that $V = \{v_1, v_2 \dots v_n\}$ is the set of nodes representing social actors, $E = \{e_1, e_2 \dots e_m\}$ is the set of edges, representing different relationships and $A = \{a_1, a_2 \dots a_k\}$ is the set of attributes associated with the nodes, that represent properties associated with the social actors. The node v_i has a vector $[a_{i1}, a_{i2} \dots a_{ik}]$ where its value on attribute a_j is a_{ij} . The objective of this work is to find communities in an attributed graph, i.e., to partition the graph into communities $G_i = (V_i, E_i, A)$, where $V_i \cap V_j \neq \emptyset$. Nodes in the same communities are expected to be highly connected and have similar attributes.

B. Proposed Model

In this paper, an incremental multi-agent approach based on electric field that is called *MASEF* i.e is presented. Multi-Agent System for Community Detection. In this incremental proposal, the dynamic social network is defined as a single graph with a set of events (succession of modifications) on nodes and edges. We start by a random partition and according to the evolution of the network, the previous detected partition is adapted in real time.

To do so, a group of autonomous agents is used and they work together to update the communities. In fact, an agent

is associated to each detected community. The environment in which the agents live, evolve and die is described by the graph. To update its community according to the dynamic of its members, we were inspired by the laws of electric fields. Indeed, each agent is seen as a particle that creates an electric field around it. It applies an attractive force to add new members and some neighboring communities. In the same time, it applies a repulsive force to reject some members and to get away from other communities. These forces are based on the structural and the attributes similarity as detailed in the rest of this section.

According to Coulomb's Law, the electric force applied by an agent is defined as:

$$\vec{F} = q \cdot \vec{E} \quad (1)$$

such that \vec{E} is the electric field vector. Communities are assumed to be **positively** charged particles. So nodes and communities with positive charge will be attracted. However, nodes and communities with negative charge will be rejected (see Fig. 2):

- if $q < 0$: the direction of E is opposite to that of the electric force F (attractive force).
- if $q > 0$: the direction of E is that of the electric force F (repulsive force).

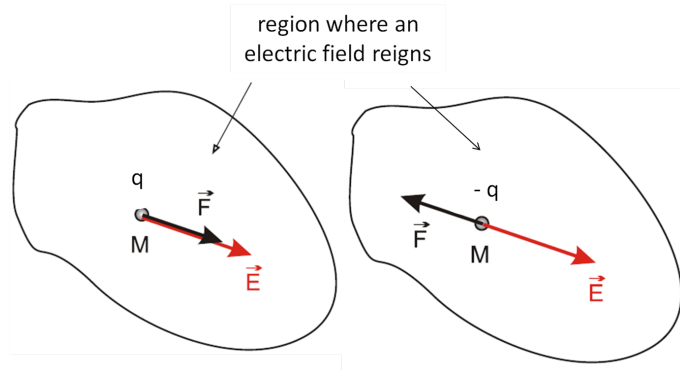


Fig. 2. An electric field with a positive charge.

Community-node Force

In the proposed model, q , that is the charge of a node n submitted to the electric field created by the community C , is defined as:

$$q(C, n) = Rep(C, n) - Att(C, n) \quad (2)$$

such that

$$Att(C, n) = \alpha \frac{NB(C, n)}{|C|} + (1 - \alpha) sim(C, n) \quad (3)$$

and

$$Rep(C, n) = \alpha \frac{NB(C, n)}{|C|} + (1 - \alpha)(1 - sim(C, n)) \quad (4)$$

where $NB(C, n)$ is the number of neighbors of n in C , $NB(C, n)$ is the number of nodes in C that are not related to n ,

$|C|$ is the number of nodes in C , $sim(C, n)$ is the similarity of n to the members of C and $\alpha \in [0, 1]$ is a weighting parameter to balance the trade-off between the structural similarity and the attributes similarity of n to C . More n is attached and similar to members of C , more $Att(C, n)$ will be important. On the other hand, $Rep(C, n)$ is important when the node n has few neighbors in C and is dissimilar to the members of C . We note that:

- If $Att(C, n) > Rep(C, n)$ so $q < 0$ (attractive force).
- If $Att(C, n) < Rep(C, n)$ so $q > 0$ (repulsive force).

For the definition of the similarity, the similarity proposed by Gonzalo in [37] is chosen and it defined as:

$$S(C) = \frac{1}{D} \left[\sum_{v, w \in C} \sum_{j=1}^D \frac{sim(x_{jv}, x_{jw})}{|C|^2} \right],$$

$D = |X|$ is the number of attributes in G , x_{jv} is the value of the j -th attribute for node v , and $sim(x_{jv}, x_{jw})$ is a function of the similarity between x_{jv} and x_{jw} .

For binary attributes, $sim(x_{jv}, x_{jw})$ is given by the *simple matching coefficient* between x_{jv} and x_{jw} :

$$sim(x_{jv}, x_{jw}) = \frac{\sum_{k=1}^d (x_{jv_k} \wedge x_{jw_k}) \vee (\neg x_{jv_k} \wedge \neg x_{jw_k})}{d}$$

For categorical attributes, $sim(x_{jv}, x_{jw})$ is given by the *Jaccard similarity index* between the "1-of-N" binary encodings of x_{jv} and x_{jw} :

$$sim(x_{jv}, x_{jw}) = \frac{\sum_{k=1}^d x_{jv_k} \wedge x_{jw_k}}{\sum_{k=1}^d x_{jv_k} \vee x_{jw_k}}$$

For numeric attributes, $sim(x_{jv}, x_{jw})$ is given by the inverse of one plus the *Euclidean distance* between x_{jv} and x_{jw} :

$$sim(x_{jv}, x_{jw}) = \frac{1}{1 + \sqrt{\sum_{k=1}^d (x_{jv_k} \wedge x_{jw_k})^2}}$$

where d is the number of dimensions of the j -th attribute, x_{jv_k} is the value of the k -th coordinate of the j -th attribute for node v , and \neg , \wedge and \vee are the logical NOT, AND, and OR operators, respectively.

This attribute similarity function allows the combination of the attributes of different types, which is essential given the heterogeneous nature of many real-world networks.

Community-Community Force

q , the charge of a community C' submitted to the electric field created by the community C , is defined as:

$$q(C, C') = Rep(C, C') - Att(C, C') \quad (5)$$

such that

$$Att(C, C') = \frac{Overlap(C, C')}{|C'|} \quad (6)$$

and

$$Rep(C, C') = \frac{Overlap(C, C')}{|C'|} \quad (7)$$

where $Overlap(C, C')$ is the number of overlapping nodes, i.e. nodes that belong to C and to C' . If the number of overlapping nodes is greater than the number of non overlapping nodes in C' , this community will be attracted by the C :

- If $Att(C, C') > Rep(C, C')$ so $q < 0$ (attractive force).
- If $Att(C, C') < Rep(C, C')$ so $q > 0$ (repulsive force).

1) *When are the agents reacting?*: The events observed in the network can be classified into two types: negligible events and important events. Negligible events are those that do not change the community structure (such as creating a relationship between two members of the same community or breaking a relationship between two members belonging to different communities). These events do not require the communities update.

On the other hand, important events are those that can alter the existing community structure such as the creation of new relationships between members belonging to different communities (this may lead to the fusion of these two communities), or the breaking of relations between members belonging to a same community (which may lead to the division of this community). In this model, the agents react only when important events occur which makes it possible to neglect minor events and to treat very dynamic networks.

2) *Agents*: The proposed model start by a random partition. An agent is assigned to each detected community and it is defined by his *id* and the following properties:

- List of its members, which are the components of its community.
- List of nodes having relations with its members.
- List of neighboring agents (agents associated with communities having relationships with its community).
- List of the most connected nodes (i.e. having the biggest degree) in its community. This list contain only the 20% of the community members.

An agent is defined by the following possible behaviors:

- Decide to integrate or not a node to itself, which leads to the *growth* of the community.
- Decide to remove or not a node from itself, which leads to the *contraction* of the community.
- Decide to integrate or not another community, which leads to the merge of the community.
- Decide to divide itself or not, which leads to the division of the community.
- Decide to create a community, which leads to the birth of a new community.
- Decide to die, which leads to the death of the community.

Growth and contraction of a community

When a new node n appears in the network and creates relationships with members of existing communities, n will be subject to the electric fields created by these communities. Similarly, if a node n belonging to community C_n creates new relationships with some members of another community C , C applies an electric field on n .

The node n will therefore be subject to several force created by the different neighboring communities (see Figure 3). These forces can be attractive or repulsive. Finally, n will be attracted by the community that applies the most attractive force. In the case of equality (several communities apply the same strength of attractive force), n will be integrated by several communities at the same time and in this case, it will be an overlapping node. Once integrated into the community C , the agent a_C associated with C informs the agent a_{C_n} associated with C_n (to which n belonged). a_{C_n} will remove n from the list of its members. We notice the contraction of the community C_n , since it has lost a member, and the growth of C .

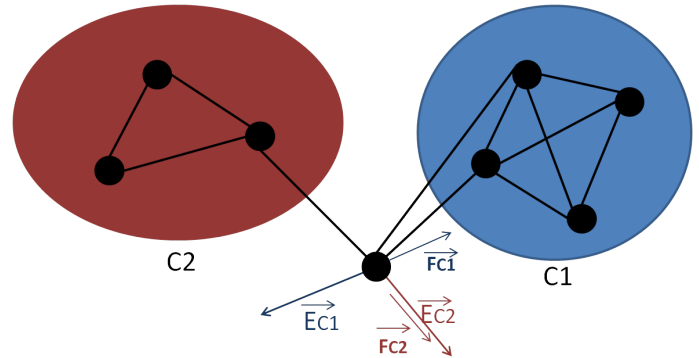


Fig. 3. An example of a node subject to two electric fields emitted by two communities.

Birth of a community

If a member n appears in the network and it is attached to a set of new members newly appeared in the networks, it will be subject to the electric field applied by the virtual community containing these friends. If n is attached to some existing communities, it will be also subject to forces created by these communities (see Fig. 4). n will finally be attracted by the community that exerts the most important attractive force.

Division of a community

When an agent deletes nodes belonging to the list of the most connected nodes, this agent uses the model of *Gonzalo* [37] (see Section IV-A) to detect the possible sub-communities within its community (see Fig. 5). Subsequently, it assigns a new agent to each sub-community and thereafter it leaves the system.

Merge of communities

As it is noted, each community applies an electric field to neighboring communities. When the number of overlapping nodes between two communities C and C' will be greater than the number of the non overlapping nodes in C' (see Fig. 6), the force applied by C on C' will be an attractive force. C' will then merged with C .

When several communities apply an attractive force on a

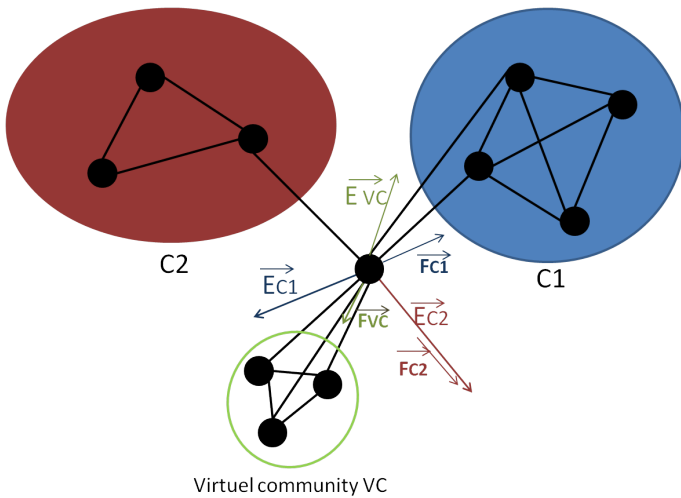


Fig. 4. An example of a node subject to an electric field emitted by existing communities and a virtual community.

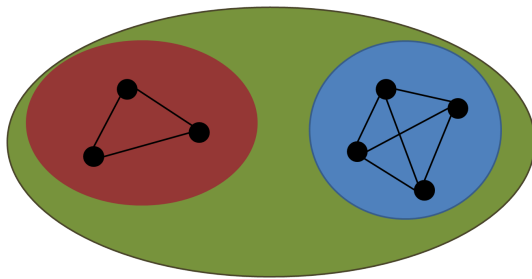


Fig. 5. An example of a community division.

community C' , this one will be integrated to the community that applies the greater force.

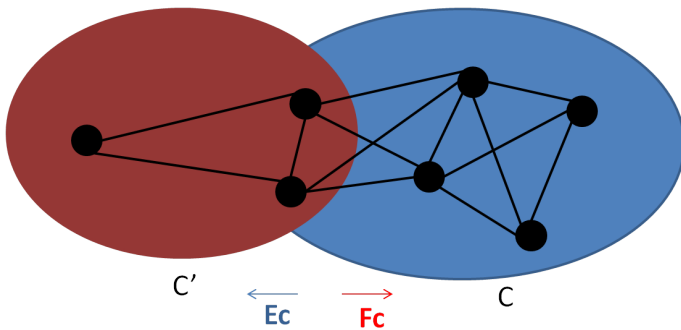


Fig. 6. An example of a community subject to an electric field created by another community.

Death of a community

As we have shown, the community can be attracted by an existing community. In this case, we are talking about the death of this community. This result can also be considered in a particular case in which the community loses all its members.

Following these different reactions of the agents, the user can observe the communities that emerge over time and evolve according to the dynamic of social members and their interactions.

IV. EXPERIMENTATION AND EVALUATION

In order to study the performance of the proposed approach, it is applied to real data from well-known online social networks and artificial data. In this section, the main results of his experiments are reported.

A. Choice of Algorithms to Compare

In these experiments, the results of *MASEF* are compared with the those of two models among the best performing. These models are:

- *GM* model (Gonzalo' model) [37] : the goal of this algorithm of community detection in attributed graphs is to maximize both the modularity and the attribute similarity (presented in section III-B of the partition of the graph. To track the evolution of communities over time, Gonzalo's algorithm identifies communities at each time step and then matches the communities identified at consecutive time steps.
- The *iLCD* model (intrinsic, Longitudinal Community Detection) [5] which is an incremental and multi-agent model. In this work, the author associates with each detected community an autonomous agent called agent-community. The detection of a community is done by replaying the creation of the network edge by edge. Once a clique (usually 3 or 4 members) is present, a new community with a new community agent is created. This agent successively adds neighbors who improve the quality of the community. Through the actions of creating new communities and integrating members into existing communities, the approach succeeds in finding the initial partition. Subsequently, community agents update their communities according to the evolution of the network.

B. Evaluation Measures

For the evaluation of *MASEF*, two measures well known in the literature are chosen. The first measure is the weighted modularity [6] which evaluates the quality of the partition obtained based in the internal and external links of its communities. This measure is the most used to qualify a partition of a graph. For a weighted graph, modularity is defined as the fraction of the weight of the links ending in the same community subtracted from the same value if the links were placed at random. The more important the modularity, the better is the partition. Weighted modularity is defined by:

$$MQ_w = \frac{1}{2W_s} \sum_{ij} [W_{ij} - \frac{W_i W_j}{2W_s}] \delta_{(c_i, c_j)} \quad (8)$$

such that W_{ij} is the weight of the links connecting the nodes i and j , $W_i = \sum_j W_{ij}$ is the sum of the link weights of the node i , the sum of weight is $2W_s = \sum_{ij} W_{ij}$, c_i means that the node i belongs to the community c_i . The $\delta_{(c_i, c_j)}$ function takes the value 1 if i and j belong to the same community, and takes the value 0 otherwise.

The second measure used to evaluate the quality of detected partitions is the weighted entropy that evaluates the partition

based on the attributes similarity of the members in a same community. Let A be a graph with $|V|$ nodes. To each node, a set of attributes $A = \{a_1, \dots, a_m\}$ is associated. To each attribute a_i is associated a weight w_i . n_i denote the number of values that the attribute a_i can take. The entropy for a partition $P = \{C_1, \dots, C_k\}$ of this graph is defined by [38]:

$$entropic(P) = \sum_{i=1}^m \frac{w_i}{\sum_{p=1}^m w_p} \sum_{j=1}^k \frac{|C_j|}{|V|} entropic(a_i, C_j) \quad (9)$$

such that

$$entropic(a_i, C_j) = - \sum_{n=1}^{n_i} p_{ijn} \log_2(p_{ijn}) \quad (10)$$

and p_{ijn} is the percentage of nodes in the community C_j having a value a_{in} for the attribute a_i . The P entropy measures the weighted entropy for all attributes in the k communities.

The entropy belongs to the interval $[0, \infty [$. A low entropy value is equivalent to a great homogeneity of the attributes of the nodes in the same community.

C. Application to Real Networks

Datasets

The performance of the proposed algorithm is evaluated on large-scale dynamics attributed networks constructed from four real-world data sets: DBLP, Yelp, TripAdvisor and Google+. Table I summarizes the characteristics of these networks such that $\#n_i$ and $\#m_i$ represent the initial number of nodes and edges, respectively, $\#n_f$ and $\#m_f$ represent the final number of nodes and edges, respectively, and $\#timestep$ denotes the number of time steps.

TABLE I. CHARACTERISTICS OF REAL-WORLD NETWORKS.

	DBLP	Yelp	TripAdvisor	Google+
$\#n_i$	13,782	7	15	300 000
$\#n_f$	110,065	97,039	297,301	500 000
$\#m_i$	33,528	14	26	625 124
$\#m_f$	426,548	10,372,332	28,288,858	1 685 124
$\#time\ step$	7	7	7	4

TABLE II. RUNNING TIME IN SECONDS FOR REAL WORLD NETWORKS.

	DBLP	Yelp	TripAdvisor
Proposed model	10.1	101.2	166.1
iLCD	9.5	88.3	102.1
GM	25.29	299.81	385.27

- The DBLP ¹ data set provides publication records from 1991 to 2000. In the corresponding network, an edge between two nodes is present if the authors represented by those two nodes collaborate in a publication. The authors have 19 categorical attributes representing each author's areas of publication (e.g., artificial intelligence, bioinformatics, security).
- The Yelp ² data set provides user reviews of a select set of businesses from 2004 to 2012. In the corresponding network, an edge between two nodes is

present if the users represented by those two nodes reviewed the same business. Nodes have 38 categorical attributes representing the type of businesses reviewed by each user (e.g., restaurants, shops, services), as well as a numeric attribute corresponding to the average rating assigned by each user.

- The TripAdvisor ³ data set provides hotel reviews from 2002 to 2012. In the corresponding network, an edge between two nodes is present if the users represented by those two nodes reviewed the same hotel. Nodes have a numeric attribute corresponding to the average rating assigned by each user.
- The Google+ data set ⁴ : UC Berkeley has published four snapshots of a part of Google+ network. These Data contain also some attributes of social members : job, school ,address.

Experimental Setup for real networks

Communities in these real-world networks using *MASEF*, as well as *iLCD* and *GM* are identified. Note that *GM* is an algorithm for detecting communities in dynamic attributed graphs, as for *iLCD*, they do not consider the nodes attributes. All experiments on real-world networks were performed on an Intel machine running RHEL Server 6.7 consisting of two hex-core E5645 processors and 64GB DDR2 RAM. The proposed algorithm was implemented in JAVA. It is to be noted that in all these experiments, the value of the weighting parameter α of the presented equations is set to 0.5.

The algorithms were compared in terms of the quality of the identified communities and the efficiency of the implementation. To evaluate their structural properties, the modularity of the graph partition is measured, and to evaluate the homogeneity of their attribute information, the average entropy is measured. The obtained results are shown in Fig. 7,8,9,10,11,12,13 and 14.

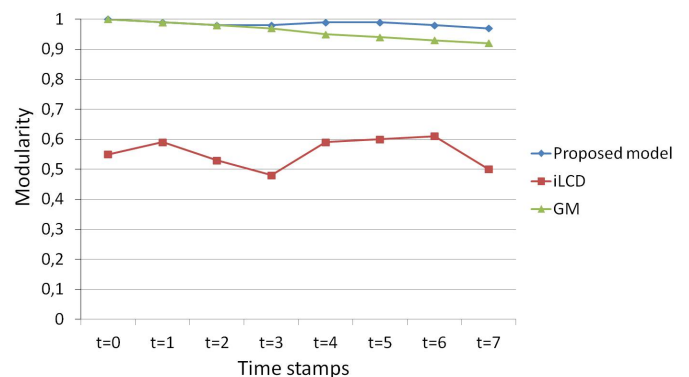


Fig. 7. Variation of modularity of the tested models for DBLP network.

Finally, the Table II summarize the running times in seconds.

Discussion of Results for real networks

¹dblp.uni-trier.de/xml

²www.yelp.com/dataset_challenge

³times.cs.uiuc.edu/~{ }wang296/Data

⁴http://projects.csail.mit.edu/dnd/

TABLE III. DESCRIPTION OF THE DYNAMIC NETWORK GENERATOR PARAMETERS.

Parameter	Signification	Value
K	Number of communities	50
N	Number of nodes	5000
p	Number of numerical attributes	10
$A = \gamma_1, \dots, \gamma_p$	Standard deviations of the attributes generated using centered normal distributions	0
E_{with}^{max}	Maximum number of edges connecting a new node to nodes in its community	10
E_{btw}^{max}	Maximum number of edges connecting a new node to nodes in a different community	10
$NbRep$	Maximum number of representatives of each community	5
MTE	Minimum number of total edges	10 000
$P_{randomCommunity}$	A threshold to decide if a new node joins a randomly selected community or not	0.5
$ProbaMicro$	A threshold to select if the micro dynamic updates are performed or not	0.5
$Addnode$	Ratio defining the number of nodes inserted	0.3
$Removenode$	Ratio defining the number of nodes removed	0.3
$UpdateAttr.$	Ratio defining the number of attributes updated	0
$AddBtw.Edges$	Ratio defining the number of between edges inserted	0.3
$RemoveBtw.Edges$	Ratio defining the number of between edges removed	0.9
$AddWth.Edges$	Ratio defining the number of within edges inserted	0.5
$RemoveWth.Edges$	Ratio defining the number of within edges removed	0.5
$Timestamps$	Number of graphs generated	10
$ProbaMigrate$	Probability to perform the migrate nodes operation	0.75

TABLE IV. DESCRIPTION OF “MACRO OPERATIONS” PARAMETERS FOR REFERENCES GRAPHS.

Parameter	Signification	Value
$P_{removeEdgeSplit}$	Proba. to remove an edge between two nodes in the same community when splitting a community	0.3
$P_{robaMerge}$	Probability to perform the merge operation	0.3
$P_{robaSplit}$	Probability to perform the split operation	0.3

TABLE V. RUNNING TIME (IN SECONDS) FOR REFERENCES GRAPHS

	$C\bar{P}U$
Proposed model	27
iLCD	18
GM	25.94

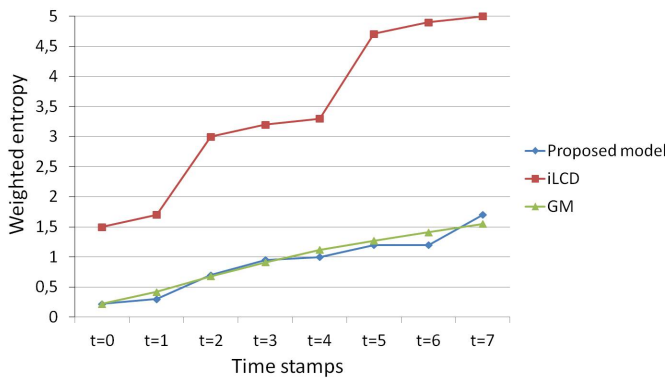


Fig. 8. Variation of weighted entropy of the tested models for DBLP network.

From these experiments on real graphs, we can conclude that partitions of *MASEF* as well as those of *GM* were particularly relevant given the high modularity and low weighted entropy of these partitions, which proves the good structural quality of the detected partitions as well as the homogeneity of the members within the same communities.

On the contrary, the quality partitions of *iLCD* still had inferior quality, due to the fact that *iLCD* does not integrate

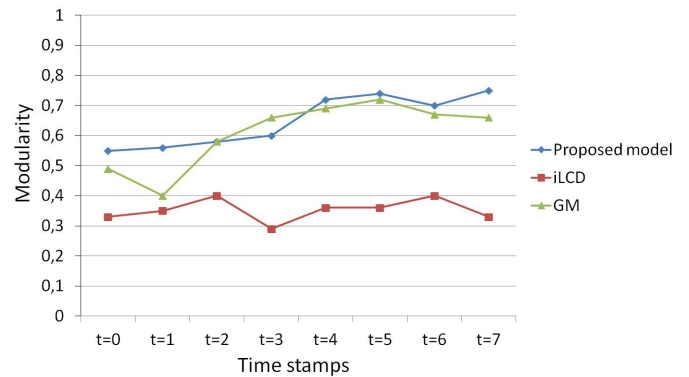


Fig. 9. Variation of modularity of the tested models for YELP network.

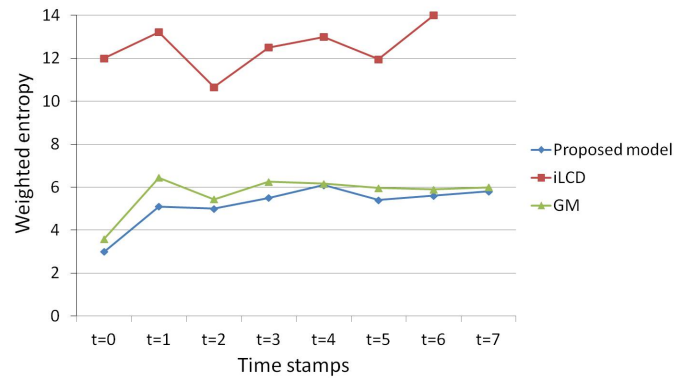


Fig. 10. Variation of weighted entropy of the tested models for YELP network.

the attribute similarity for the detection of communities. With respect to the efficiency of the algorithms, *MASEF* was proven to be slower than *iLCD*, because it takes time to find an initial partition. On the other hand, the proposed approach is much faster than the *GM*, which was expected as it is a centralized model. Thus, the quality of the communities found by *GM* model was at the expense of execution time which is considerably higher. This suggests that the proposed algorithm is able to achieve a better balance between the efficiency and

TABLE VI. DESCRIPTION OF “MACRO OPERATIONS” PARAMETERS FOR GRAPHS WITH SEVERAL MERGE OPERATIONS

Parameter	Signification	Value
$P_{removeEdgeSplit}$	Proba. to remove an edge between two nodes in the same community when splitting a community	0.3
$P_{robaMerge}$	Probability to perform the merge operation	0.75
$P_{robaSplit}$	Probability to perform the split operation	0.3

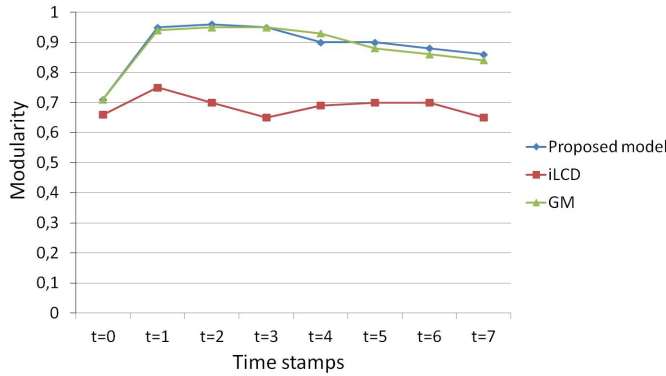


Fig. 11. Variation of modularity of the tested models for TripAdvisor network.

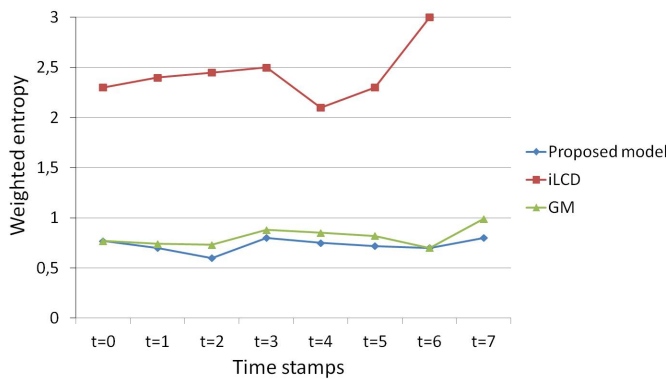


Fig. 12. Variation of weighted entropy of the tested models for TripAdvisor network.

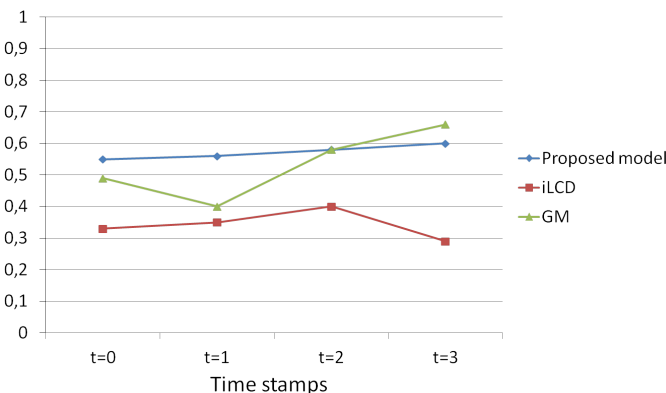


Fig. 13. Variation of modularity of the tested models for Google+ network.

TABLE VII. RUNNING TIME (IN SECONDS) FOR GRAPHS WITH SEVERAL MERGE OPERATIONS

	CPU
Proposed model	56
iLCD	45
GM	103

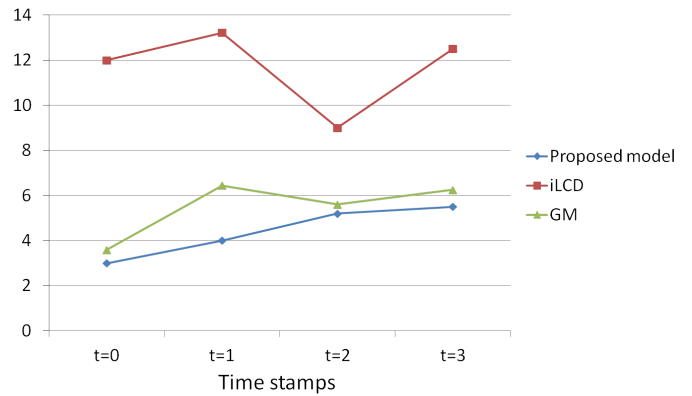


Fig. 14. Variation of weighted entropy of the tested models for Google+ network.

the quality of the identified communities.

D. Application to Artificial Networks

To build synthetic datasets, the generator *DANCer* presented in [39] is used. A network is defined by a sequence of undirected attributed graphs having a well defined partition. The ground truth partition is given by P_t^* with $t \in \{1, \dots, T\}$. The evolution of the network is obtained by removing or adding edges, by migrating nodes from a community to another one, by splitting a community into two new sub-communities or by merging two existing communities into a single community. Therefore, the real structure of each generated social network is used as a reference point. As evaluation criteria, the normalized mutual information (NMI) is used. *NMI* is the most standard commonly used measure to compare two partitions of the same graphs. It is defined as follows [40]:

$$NMI(A, B) = \frac{-2 \sum_{a \in A} \sum_{b \in B} |a \cap b| \log\left(\frac{|a \cap b|}{|a||b|}\right)}{\sum_{a \in A} |a| \log\left(\frac{|a|}{n}\right) + \sum_{b \in B} |b| \log\left(\frac{|b|}{n}\right)}, \quad (11)$$

with A and B being two distinct partitions of the same graph.

Using *DANCer*, 10 dynamic networks with the same set of parameters are generated. Table III presents a description of *DANCer* parameters and the common used values in all these simulations.

TABLE VIII. DESCRIPTION OF “MACRO OPERATIONS” PARAMETERS FOR GRAPHS WITH SEVERAL SPLIT OPERATIONS

Parameter	Signification	Value
$P_{removeEdgeSplit}$	Proba. to remove an edge between two nodes in the same community when splitting a community	0.75
$P_{robaMerge}$	Probability to perform the merge operation	0.3
$P_{robaSplit}$	Probability to perform the split operation	0.75

TABLE IX. RUNNING TIME (IN SECONDS) FOR GRAPHS WITH SEVERAL SPLIT OPERATIONS

	CPU
Proposed model	130
iLCD	76
GM	246

Experimental Setup for artificial networks

- 1) First case: references graphs
In the first case, reference graphs having “Macro operations” parameters are considered. These graphs are described in Table IV. The average NMI of *MASEF*, *iLCD* and *GM* is shown in Figure 15. The Table V summarize the running time (in seconds) of the three models for references graphs.

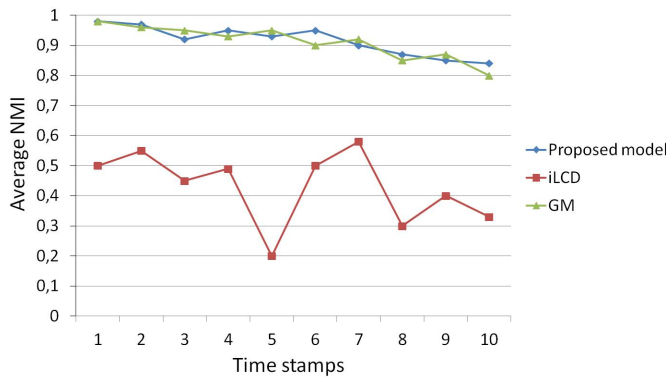


Fig. 15. Variation of NMI of the tested models for references graphs.

- 2) Merge operation
In this second set of runs, we were interested by graphs in which several communities merge as the merge is one of the most difficult movements to detect. The “Macro operations” parameters of this set of graphs are described in Table VI. The behavior of the different algorithms in this case is followed. The average NMI is showed in Fig. 16, and the Table VII summarize the running time (in seconds) of the algorithms for this set of graphs.
- 3) Split operation
In the third set of runs, graphs with an important number of split operation (see the Table VIII for the “Macro operations” parameters) are considered. The resulting NMI is presented in Fig. 17 and the resulting running time of the three approaches is summarized in Table IX.
- 4) Merge and Split operation
Finally, the most complex case, when several communities merge and other ones split, is considered.

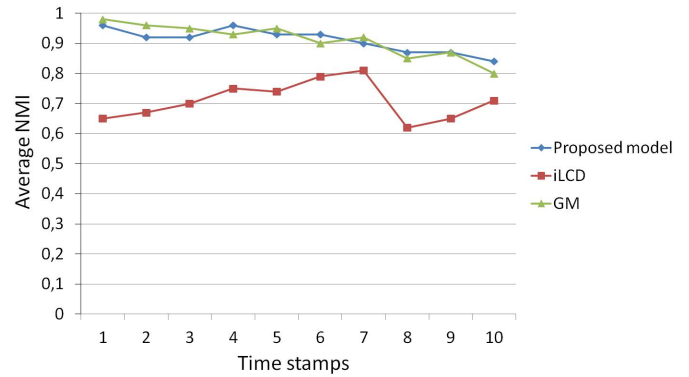


Fig. 16. Variation of NMI of the tested models for graphs with several merge operations.

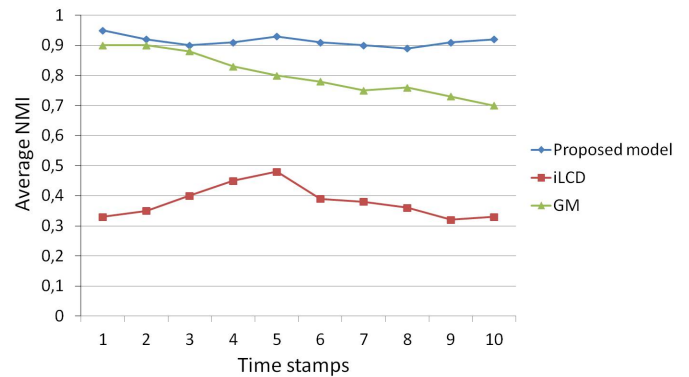


Fig. 17. Variation of NMI of the tested models for graphs with several split operations.

The “Macro operations” parameters of this set of graphs are described in Table X. The resulting NMI is presented in Fig. 18 and the resulting running time is summarized in Table XI

Discussion of Results for artificial networks

From Fig. 15,16 and 17, we can see that *MASEF* was able to find a very close partition to the exact partition for all graphs thanks to the accordance between the computed and the exact partitions ($NMI \simeq 1$). We notice that the partitions of *GM* were very close to *MASEF* partition, which in several times was closer to the correct partition than *GM*. Regarding the CPU-time, we notice in Tables V, VII, IX and XI that the proposed model was slower than *iLCD* and faster than *GM* in computing an optimal partition.

From these simulations on artificial networks, we can conclude that *MASEF* performed well for distinct types of graphs, and it was always able to compute the exact structure of each network regardless of its nature and complexity. On

TABLE X. DESCRIPTION OF “MACRO OPERATIONS” PARAMETERS FOR COMPLEX GRAPHS.

Parameter	Signification	Value
$P_{removeEdgeSplit}$	Proba. to remove an edge between two nodes in the same community when splitting a community	0.75
$P_{robaMerge}$	Probability to perform the merge operation	0.75
$P_{robaSplit}$	Probability to perform the split operation	0.75

TABLE XI. RUNNING TIME (IN SECONDS) FOR COMPLEX GRAPHS OPERATIONS.

	$C\bar{P}U$
Proposed model	254
iLCD	112
GM	4789

TABLE XII. CHARACTERISTICS OF THE WIDE-SCALE NETWORK.

Nb of time steps	T_0	T_{50k}	T_{100k}
Nb of nodes	500	1 M	2 M
Nb of links	2,5 k	10,2 M	22,9 M
Nb of events	0	11,2 M	25,6 M

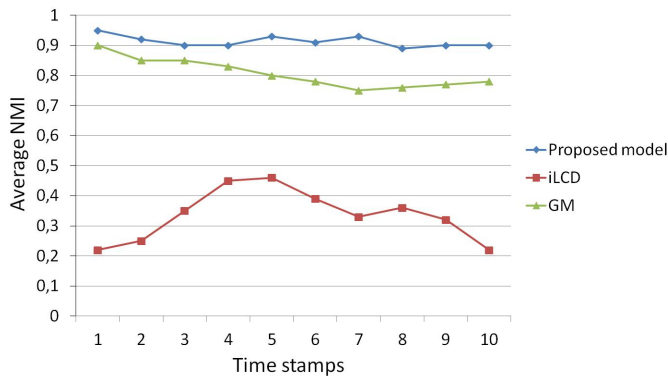


Fig. 18. Variation of NMI of the tested models for complex graphs.

the other hand, although *MASEF* was slower than *iLCD*, it detected a much better partition.

E. Application to Large Scale Artificial Networks

The objective of this experiment is to test the reliability and performance of *MASEF* to deal with graphs of large scales. For this purpose, we have generated a network with a size that grows progressively over time by adding a random number of links and nodes. After more than 100k steps of time, the size of the graph becomes large. Table XII summarizes the different characteristics of this network.

The simulation is launched on this network and it is stopped when we reach a size that exceeds two million nodes and 22 million links. This experience was limited by the size of available RAM (16 GB of RAM). The simulation was stopped after 15 hours and 51 minutes. During this same simulation time, *GM* managed only 170 250 nodes. *ILCD* was able to partition only 1,950,430 nodes. The advantage of *MASEF* lies in its decentralized nature, which consists of processing the only important events of the network in a local way. As a result, the proposed multi-agent system self-stabilizes rapidly and we speak of perturbations only at the level of the agents concerned by the event.

Fig. 19 shows the variation of the weighted modularity of the partitions detected in a few steps of time. The curves show the good qualities of *MASEF*'s partitions which do not degrade with the increasing size of the network. On the other hand, a significant degradation of the qualities of the partitions obtained by *iLCD* is noted. *GM* ensures a good partition quality but it cannot resist face to the increasing size of the network.

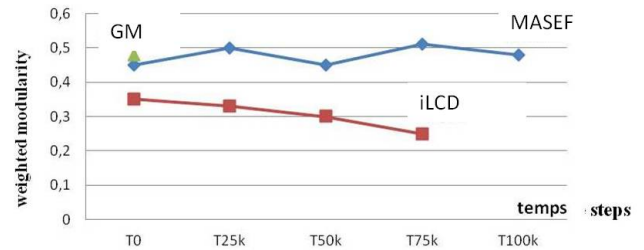


Fig. 19. Variation of the weighted modularity of the partitions detected for the large artificial network.

The results of these experiments show the effectiveness of *MASEF* to the detection of dynamic communities. Indeed, we have obtained particularly interesting results which the previous methods were not able to attain.

V. CONCLUSION AND PERSPECTIVES

In this work, an incremental multi-agent approach for community detection in dynamic social networks is presented. In the proposed method, a set of agents work together to update existing community. To do so, each one applies an electric field to attract similar and very connected members and reject the others.

Thanks to its decentralized and incremental nature, the *MASEF* approach can detect the dynamic of communities based on local computation allowing the adaptation of the existing partition. Therefore, this solution is able to treat large scale networks. The main contribution of the proposed approach compared to multi-agent methods presented in the literature is that it allows all operations on communities: birth, death, growth, contraction, and also the most complex events which are the merge and the division. In addition, the consideration of the characteristics of social actors presents a considerable contribution of *MASEF* compared to most existing models. Another originality of the proposed approach is the use of electric field as an auto-organization tool for the different agents. Experimental results on both synthetic and real-world networks demonstrate the effectiveness of the proposed approach.

In our future work, we aim to integrate the amount of the exchanged data between the social members for the purpose of community detection. In fact, in large social networks such as Facebook and Twitter, the communities can be recognized as the groups of users who are often interacting with each other. Therefore, the amount of the exchanged data could be applied as a parameter for an efficient community detection.

ACKNOWLEDGMENT

This research was funded by Deanship of Scientific Research at Princess Nourah bint Abdulrahman university (Grant No.39-YR-5).

REFERENCES

- [1] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs to test community detection algorithms," *Physical Review E*, vol. 78, no. 4, p. 046110, 2008.
- [2] H. Zardi and L. B. Romdhane, "An $o(n^2)$ for detecting communities of unbalanced sizes in large scale social networks," *Knowledge-Based Systems*, vol. 37, pp. 19–36, 2013.
- [3] H. Cai, V. W. Zheng, F. Zhu, K. C. Chang, and Z. Huang, "From community detection to community profiling," *CoRR*, vol. abs/1701.04528, 2017.
- [4] G. Palla, A. Barabasi, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, 2007.
- [5] R. Cazabet, "Detection of dynamic communities in temporal networks," Ph.D. dissertation, Université Paul Sabatier - Toulouse III, 2013.
- [6] S. Fortunato, "Community detection in graphs," *Physics Report*, vol. 486, no. 3-5, pp. 75–174, 2010.
- [7] J. Kim and J.-G. Lee, "Community detection in multi-layer graphs: A survey," *SIGMOD Rec.*, vol. 44, no. 3, pp. 37–48, Dec. 2015. [Online]. Available: <http://doi.acm.org/10.1145/2854006.2854013>
- [8] W. Chen, Z. Liu, X. Sun, and Y. Wang, "A game-theoretic framework to identify overlapping communities in social networks," *Data Mining and Knowledge Discovery*, vol. 21, no. 2, pp. 224–240, 2010.
- [9] 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.
- [10] H.Zardi and L. B. Romdhane, "Mwep: Efficiently mining community structures in weighted large scale social graphs," in *Proceedings of the first international conference on Reasoning and Optimization in Information Systems.*, 2013, pp. 30–38.
- [11] J. Ji, X. Song, C. Liu, and X. Zhang, "Ant colony clustering with fitness perception and pheromone diffusion for community detection in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 15, pp. 3260–3272, 2013.
- [12] T. Chakraborty and A. Chakraborty, "Overcite: Finding overlapping communities in citation network," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 25-28 August 2013, pp. 1124–1131.
- [13] L. Zhou, K. Lü, C. Cheng, and H. Chen, "A game theory based approach for community detection in social networks," in *Proceedings of the 29th British National Conference on Big Data*, Berlin, Heidelberg, 8-10 July 2013, pp. 268–281.
- [14] J. Qu, "Pso algorithm with repairing strategy for community detection," *Journal of Information and Computational Science*, vol. 10, no. 13, pp. 4167–4175, 2013.
- [15] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng, "Gbage: A general bayesian framework for attributed graph clustering," *ACM Transactions on Knowledge Discovery from Data*, vol. 9, no. 1, pp. 5:1–5:43, 2014.
- [16] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. Tseng, "Analyzing communities and their evolutions in dynamic social networks," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 2, pp. 8:1–8:31, 2009.
- [17] J. Liu, C. Gao, and N. Zhong, "Autonomy-oriented search in dynamic community networks: A case study in decentralized network immunization," *Fundam. Inf.*, vol. 99, no. 2, pp. 207–226, 2010.
- [18] Z. Chen, K. Wilson, Y. Jin, W. Hendrix, and N. Samatova, "Detecting and tracking community dynamics in evolutionary networks," in *IEEE International Conference on Data Mining Workshops*, 14-17 December 2010, pp. 318–327.
- [19] P. Mucha, T. Richardson, K. Macon, M. Porter, and J.-P. Onnella, "Community structure in time-dependent, multiscale, and multiplex networks," *Science*, vol. 328, no. 5980, pp. 876–878, 2010.
- [20] T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin, "Detecting communities and their evolutions in dynamic social networks—a bayesian approach," *Machine Learning*, vol. 82, no. 2, pp. 157–189, 2011.
- [21] J. Xie, M. Chen, and B. Szymanski, "Labelrank: Incremental community detection in dynamic networks via label propagation," in *Proceedings of the Workshop on Dynamic Networks Management and Mining*, New York, NY, USA, 22-27 June 2013, pp. 25–32.
- [22] H. Alvari, A. Hajibagheri, and G. Sukthankar, "Community detection in dynamic social networks: A game-theoretic approach," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 17-20 August 2014, pp. 101–107.
- [23] J. Han, W. Li, and L. Z. Z. S. Y. Z. W. Deng, "Community detection in dynamic networks via adaptive label propagation," *PLoS ONE*, vol. 12, no. 11, p. e0188655, 2017.
- [24] P. Agarwal, R. Verma, A. Agarwal, and T. Chakraborty, "Dyperm: Maximizing permanence for dynamic community detection," *CoRR*, vol. abs/1802.04593, 2018. [Online]. Available: <http://arxiv.org/abs/1802.04593>
- [25] Y. Wang, B. Wu, and N. Du, "Community evolution of social network: Feature, algorithm and model," *Science And Technology*, p. 60402011, 2008.
- [26] D. Greene, D. Doyle, and P. Cunningham, "Tracking the evolution of communities in dynamic social networks," in *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining*, Washington, DC, USA, 9-11 August 2010, pp. 176–183.
- [27] T. Aynaud, "Détection de communautés dans les réseaux dynamiques." Ph.D. dissertation, Université Pierre et Marie Curie, 2011.
- [28] H. Ning, W. Xu, Y. Chi, Y. Gong, and T. Huang, "Incremental spectral clustering with application to monitoring of evolving blog communities," in *6th SIAM International Conference on Data Mining*, 26-28 April 2007, pp. 261–272.
- [29] T. Falkowski, A. Barth, and M. Spiliopoulou, "Studying community dynamics with an incremental graph mining algorithm," in *Proceedings of the 14th Americas Conference on Information Systems*, 14-17 August 2008, pp. 29–40.
- [30] N. P. Nguyen, T. N. Dinh, Y. Shen, and M. T. Thai, "Dynamic social community detection and its applications," *PLOS ONE*, vol. 9, no. 4, pp. 1–18, 04 2014. [Online]. Available: <https://doi.org/10.1371/journal.pone.0091431>
- [31] B. Yang, J. Liu, and D. Liu, "An autonomy-oriented computing approach to community mining in distributed and dynamic networks," *Autonomous Agents and Multi-Agent Systems journal*, vol. 20, pp. 123–157, 2009.
- [32] J. Huang, B. Yang, D. Jin, and Y. Yang, "Decentralized mining social network communities with agents," *Mathematical and Computer Modelling*, vol. 57, no. 11-12, pp. 2998–3008, 2013.
- [33] R. Badie, A. Aleahmad, M. Asadpour, and M. Rahgozar, "An efficient agent-based algorithm for overlapping community detection using nodes' closeness," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 20, pp. 5231–5247, 2013.
- [34] Z. Bu, Z. Wu, J. Cao, and Y. Jiang, "Local community mining on distributed and dynamic networks from a multiagent perspective," *IEEE Transactions on Cybernetics*, vol. 46, no. 4, pp. 986–999, 2016.
- [35] J. Ferber, *Les Systèmes multi-agents: vers une intelligence collective*, ser. Informatique intelligence artificielle. InterEditions, 1995.
- [36] E. Le Strugeon, R. Mandiau, and G. Libert, "Towards a dynamic multi-agent organization," in *Proceedings of 8th International Symposium on Methodologies for Intelligent Systems*, 16-19 October 1994, pp. 203–212.

- [37] G. A. B. Lander, "Multi-objective graph mining algorithms for detecting and predicting communities in complex dynamic networks," Ph.D. dissertation, Faculty of North Carolina State University, 2017.
- [38] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," *The Proceedings of the Very Large Database Endowment*, vol. 2, no. 1, pp. 718–729, 2009.
- [39] O. Benyahia, C. Largeron, B. Jeudy, and O. R. Zaïane, "DANCer: Dynamic Attributed Network with Community Structure Generator," in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, vol. 9853, Riva del Garda, Italy, Sep. 2016, pp. 41 – 44.
- [40] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, 2007.