



# A new algorithm for communities detection in social networks with node attributes

Haifa Gmati<sup>1</sup> · Amira Mouakher<sup>1</sup> · Antonio Gonzalez-Pardo<sup>2</sup>  · David Camacho<sup>2</sup>

Received: 16 June 2018 / Accepted: 21 October 2018 / Published online: 31 October 2018  
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

## Abstract

Revealing the community structure in social networks witnessed a determined effort. In this respect, a different category of social network can be handled, such as, dynamic social networks, social networks with node attributes, etc. In this article, we introduce a new method to solve this thriving issue in the social network with node attributes. This latter can be represented by a bipartite graph, which consists of a two sets of nodes and edges connecting these nodes. The tendency of people with similar node attributes leads to the hidden information of clusters or communities. A wealthy number of community-detection algorithms have been proposed for bipartite graphs and applied to several domains in the literature. To palliate some of the highlighted shortcomings, we introduce a new approach, called Fast-Bi Community Detection (FBCD), that aims to an efficient community detection in social networks. The main idea of this approach is to explore the set of maximum matching in the bipartite graph in order to reduce the complexity of our algorithm. The carried out experiments show the high quality of the obtained communities versus those by the pioneering ones of the literature.

**Keywords** Social networks · Maximum matching · Node attributes · Bipartite graphs · Communities detection · Quality criteria

## 1 Introduction

Community structures are possessed by many real-world networks, e.g., neural networks in biology, social networks in the humanities, or interbank networks in economics, to cite but a few. The notion of community often appears of paramount importance, since it allows a unveil the hidden structure of the network. They are usually considered as groups of nodes that are strongly linked to each other.

Over the past few years, community detection has emerged as a cornerstone task in the area of network analysis, and provides insight into the underlying structure and potential functions of the networks (Girvan and Newman 2002; Newman 2003). The goal of the community detection is to organize the different nodes of a graph into several groups or communities. This process is carried out in such

way that nodes belonging to the same community are very similar, while being different from the nodes belonging to other communities.

Many prominent researchers focused on extracting disjoint communities that partition the set of nodes within a network (Blondel et al. 2008; Pons and Latapy 2006; Rosvall and Bergstrom 2008). Recently, researchers have observed the increase in intra-community overlap and have proposed algorithms for finding overlapping communities (Yong-Yeol et al. 2010; Coscia et al. 2012; Lancichinetti et al. 2010; Yang and Leskovec 2012; Jelassi et al. 2014). For instance, in a social network, individuals may belong to multiple strong social communities, corresponding to groups, such as families, colleagues and friends. Any social network with attribute node can be represented by a bipartite graph and the extracted bi-communities can be explored later such as the case of author communities in bipartite bibliographic network which can be used for citation recommendation (Dai et al. 2018). The bipartite graphs have a particular coverage property, called the maximum matching. The latter consists in extracting the maximum number of links covering all the graph. Our approach is mainly based on this property. Indeed, we

✉ Antonio Gonzalez-Pardo  
antonio.gonzalez@uam.es

<sup>1</sup> Faculty of Sciences of Tunis, LIPAH-LR 11ES14, University of Tunis El Manar, 2092 Tunis, Tunisia

<sup>2</sup> Escuela Politecnica Superior, Universidad Autonoma de Madrid, Madrid, Spain

introduce a new approach for communities detection in social network using bipartite graph and maximum matching property. This contribution explores the properties of the bipartite graphs for finding the pertinent communities in terms of quality. In fact, we aggregate different criteria to determine these communities, such as, stability (Mouakher and Yahia 2016; Mouakher and Ben Yahia 2019; Mouakher et al. 2019), modularity (Newman and Girvan 2004), bond (Omiecinski 2003) and we rely on the overlapping measure to reduce the inter-communities overlap.

The main trust of this contribution stands in the use of the maximum matching. The latter provides a powerful mean to reduce the search space of communities in large networks. Furthermore, it allows a straightforward distribution of the algorithm in order to fulfill the scalability requirements. The contribution of the maximum matching is clearly shown by the results of the experimental section. We mainly study the performance of our approach versus those of the literature in terms of time execution and quality metrics.

The remainder of this paper is organized as follows: In the following section, we sketch the basic concepts of bipartite graph analysis and communities. In Sect. 3, various methods for community detection are categorized and scrutinized. Section 4 is devoted for a thorough description of the new contribution for communities detection in bipartite graph using the maximum matching. The benefits of the maximum matching is clearly shown by the results put in the penultimate section. The latter describes the complete experimental study and the obtained results. The last section recalls our contribution and sketches issues of future work.

## 2 Key notions

In this section, we briefly sketch the key notions used in the remainder of this paper. These notions covers the following concepts: bipartite graph, biclique, maximal biclique (Ben Yahia and Mephu Nguifo 2004), Galois connection (Hamrouni et al. 2008), community, pseudo-community (Mouakher and Yahia 2016) and maximum matching.

**Definition 1 (Bipartite Graph)** A simple graph is called bipartite and denoted by  $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ , if its vertex set can be partitioned into two disjoint subsets  $\mathcal{U}$  and  $\mathcal{V}$ , where  $\mathcal{E}$  is the set of edges. Note that every edge has the form  $e = (u, v)$  where  $u \in \mathcal{U}$  and  $v \in \mathcal{V}$  (Asratian et al. 1998), such that no vertex both in  $\mathcal{U}$ , or both in  $\mathcal{V}$ , are connected.

**Example 1** Figure 1 illustrates an example of a bipartite graph composed by 7 nodes and 8 edges.

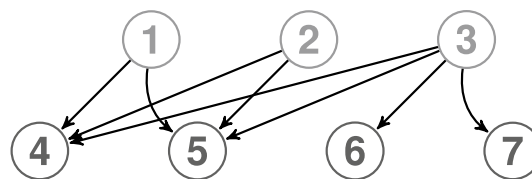


Fig. 1 An example of a bipartite graph with  $\mathcal{U} = \{1, 2, 3\}$  and  $\mathcal{V} = \{4, 5, 6, 7\}$

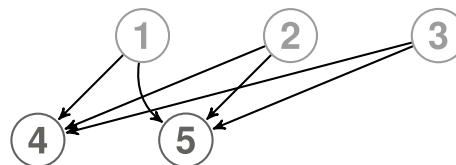


Fig. 2 An example of maximal biclique  $\langle 123, 45 \rangle$

**Definition 2 (Biclique)** Let  $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$  denote a bipartite graph. A biclique  $C = (\mathcal{U}', \mathcal{V}')$  is a subgraph of  $\mathcal{G}$  induced by a pair of two disjoint subsets  $\mathcal{U}' \subseteq \mathcal{U}$ ,  $\mathcal{V}' \subseteq \mathcal{V}$ , such that  $\forall u \in \mathcal{U}', v \in \mathcal{V}', (u, v) \in \mathcal{E}$ .

**Definition 3 (Maximal Biclique)** A maximal biclique is a largest biclique in a graph. Given a bipartite graph  $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ , a biclique  $(S_x, S_y)$  is a maximal biclique of  $\mathcal{G}$  if no proper superset of  $(S_x, S_y)$  is a biclique, i.e., there exists no biclique  $(S'_x, S'_y) \neq (S_x, S_y)$  such that  $S_x \subseteq S'_x$  and  $S_y \subseteq S'_y$ .

**Example 2** An example of a maximal biclique is illustrated by Fig. 2.

**Definition 4 (Galois connection)** Let  $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$  be a bipartite graph. The application  $\psi$  is defined from the first set of nodes (i.e.,  $\mathcal{P}(\mathcal{U})$ ) to the second set (i.e.,  $\mathcal{P}(\mathcal{V})$ ). It associates to  $U$  the set of nodes  $v \in \mathcal{V}$  that are common to all nodes  $u \in U$ :

$$\psi : \mathcal{P}(\mathcal{U}) \rightarrow \mathcal{P}(\mathcal{V})$$

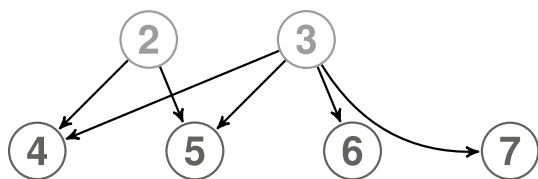
$$U \mapsto \psi(U) = \{v \in \mathcal{V} | \forall u \in U, (u, v) \in \mathcal{E}\}$$

In a dual way, the application  $\phi$  is defined from the set of nodes (i.e.,  $\mathcal{P}(\mathcal{V})$ ) to the set (i.e.,  $\mathcal{P}(\mathcal{U})$ ). It associates to  $V$  the set of nodes  $u \in \mathcal{U}$  that contains all nodes  $v \in V$ :

$$\phi : \mathcal{P}(\mathcal{V}) \rightarrow \mathcal{P}(\mathcal{U})$$

$$V \mapsto \phi(V) = \{u \in \mathcal{U} | \forall v \in V, (u, v) \in \mathcal{E}\}$$

The coupled applications  $(\psi, \phi)$  form a Galois connection between the set of nodes in  $\mathcal{U}$  and that of  $\mathcal{V}$  Barbut and Monjardet (1970).



**Fig. 3** An example of the community  $\langle 23, 4567 \rangle$  from the graph illustrated by Fig. 1

**Definition 5 (Community)** Informally, a community  $C$  is a subset of nodes of  $\mathcal{U}$  and  $\mathcal{V}$  which are connected to each other more than to other nodes of the network.

Let  $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$  be a bipartite graph. We define a community  $C = \langle A, B \rangle$  with  $A$  and  $B$  two subsets of nodes belonging respectively to  $\mathcal{U}$  and  $\mathcal{V}$ . We define by  $|E^{in}|$  the cardinality of edges inside  $C$  and  $|E^{out}|$  those are outside  $C$ ,  $A$  and  $B$  are highly connected, if the ratio between  $|E^{in}|$  and  $|E^{out}|$  is very important.

**Example 3** In Fig. 3, we can observe that node  $\{2,3\}$  and  $\{4,5,6,7\}$  are highly connected in the graph illustrated by Fig. 1.  $|E^{in}|$  is equal to 6 and  $|E^{out}|$  is equal to 2. Thus, the ratio is equal to 3. Consequently, these nodes can form a community denoted by  $\langle 23, 4567 \rangle$ . However, this is not the case for  $\langle 12, 67 \rangle$  with a ratio equal to 0. In the same respect, the ratio of  $\langle 2, 4567 \rangle$  is equal to  $\frac{2}{6} = 0.33$ . Indeed, the latter can not build a community, because the nodes  $\{2\}$  and  $\{4,5,6,7\}$  appear in a larger ratio community which is  $\langle 23, 4567 \rangle$ .

**Definition 6 (Pseudo-Community)** The pseudo-community associated to the couple  $(u, v)$ , denoted  $PC_{(u,v)}$ , is a sub-bipartite graph computed by getting the cartesian product of the maximal set of nodes fulfilling  $u$  in  $V$  and the maximal set of nodes satisfactory  $v$  in  $U$  Mouakher and Yahia (2016).

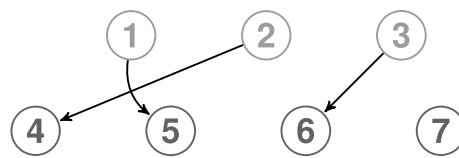
Formally,

$$PC_{(u,v)} = \{(u', v') \mid (u', v') \in \phi(v) \times \psi(u) \subseteq \mathcal{E} \mid u' \in \phi(v) \wedge v' \in \psi(u)\}.$$

The strong point of this new notion of pseudo-community associated to the couple  $(u, v)$  that all the communities which contains  $(u, v)$  and which make it possible to maximize the intra-community relation can be determined from this sub-graph. We also define the density of a given pseudo-community  $PC_{(u,v)}$  as follows:

$$Density(PC_{(u,v)}) = \frac{|PC_{(u,v)}|}{|\psi(u)| \times |\phi(v)|}$$

$|PC_{(u,v)}|$  represents the cardinality of  $PC_{(u,v)}$ . The latter cardinality is equal to the number of existing edges, whereas



**Fig. 4** An example of the maximum matching of the bipartite graph illustrated by Fig. 1

$|\psi(u)|$  is equal to the number of outgoing links from  $u$  and  $|\phi(v)|$  is equal to the number of outgoing links from  $v$ .

**Example 4**  $PC_{(2,5)} = \langle \{1, 2, 3\}, \{4, 5\} \rangle$  is a pseudo-community associated to the couple  $(2, 5)$  in the bipartite graph shown by Fig. 1 and its corresponding density is equal to  $\frac{6}{2 \times 3} = 1$ .

**Definition 7 (Maximum Matching)** A matching in a Bipartite Graph is a set of the edges chosen in such a way that no two edges share an endpoint. A maximum matching (Gibbons 1985) is defined as a matching of maximum size (maximum number of edges). In a maximum matching, if any edge is added to it, then it is no longer a matching. There can be more than one maximum matching for a given Bipartite Graph (Mucha and Sankowski 2004).

**Example 5** An example of a maximum matching is shown by Fig. 4. In this example, the set of edges  $\{(1,5), (2,4), (3,6)\}$  represent the maximum matching of the bipartite graph illustrated by Fig. 1.

After introducing the key concepts, we need to analyze how the community detection issue has been tackled by the research community.

### 3 Scrutiny of the related work

Community detection (González-Pardo et al. 2017) has been extensively studied within the context of unipartite graphs. Most of these algorithms rely on the modularity measure, as defined by Newman and Girvan (2004).

$$Q = \frac{1}{2m} \sum_{i,j} \left[ A_{ij} - \frac{K_i * k_j}{2m} \right] \delta(c_i, c_j) \tag{1}$$

where  $A_{ij}$  is the value of the adjacency matrix between the vertices  $i$  and  $j$ ,  $K_i$  is the sum of the weights of the edges adjacent to  $i$ ,  $m$  is the number of edges of the graph,  $c_i$  indicates the class assigned to the node  $i$  and  $\delta(c_i, c_j)$  is the Kronecker delta which is 1 if  $c_1$  is equal to  $c_2$ , and 0 otherwise.

$$\delta(c_i, c_j) = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are in the same community,} \\ 0 & \text{otherwise} \end{cases}$$

In the remainder of this section, we discuss community detection algorithms that are intended for bipartite networks. At a glance, the dedicated literature witnessed three main streams for addressing such a task: (1) modularity-based algorithms; (2) minimum description length algorithms; and (3) link partitioning algorithms. In the following, we sketch these approaches.

### 1. Modularity-based algorithms

Most approaches follow the modularity method, proposed by Newman and Girvan (2004), to identify communities in bipartite networks. Due to the particular structure of these kind of networks, modularity optimization required some modifications. In this respect, GUIMERA et al. proposed a bipartite modularity as the cumulative deviation from the random expectation of the number of edges between vertex members of the same bipartite community (Guimerà et al. 2007). The main weakness of this definition is that it focuses on connectivity from the perspective of only one vertex type. In the same trend, BARBER extended the definition of Newman's modularity in a unipartite network to be appropriate for bipartite networks and introduced a bipartite modularity. The latter relies on the assumption that there is a one-to-one correspondence between communities of different node types (Barber 2007). However, this definition has a limitation of assuming a one-to-one correspondence between the communities from both vertex types – i.e., the number of communities should be equal on both sides. It is worth mentioning that the main weakness of Barber's bipartite modularity stands on that the number of communities has to be determined in advance. Consequently, it is not practical in many real-life applications. Later, Murata's definition overcame the above limitations by not enforcing a one-to-one mapping between the communities of both sides (Murata 2009). Unlike previous proposals, his proposal handles two types of nodes in a uniform framework.

In this respect, Raghavan et al. proposed an algorithm for detecting communities using the techniques of Label Propagation Algorithms (LPA), which assigns unique labels to nodes and repeatedly updates the label of each vertex by assigning the most frequent labels of its neighbors until it fulfills the terminal condition (Raghavan et al. 2007). Later, Barber and Clark reformulated LPA as an optimization problem, denoted LPAb, addressed its drawbacks with additional constraints, and produced several variants of The LPA algorithm. LPAb is one of these variants that can be used to find modules in bipartite networks. The algorithm proceeds in two main stages the 'bottom up' and the 'top down'. In the first, it tries to maximize the modularity node-by-node using the propagation of labels. Next,

it tries to join modules together as far as it increases the network modularity. Subsequently, Liu and Murata introduced an improved version of LPAb, called LPAb+. The latter has been shown to have the most reliable algorithm having the highest bipartite modularity (Murata 2009).

### 2. Minimum description length algorithms

A minimum description length greedy algorithm (MDL-greedy) have been proposed by XU et al. for choosing a good modular structure in bipartite networks (Xu et al. 2010). MDL-greedy is an heuristic algorithm based on combination theory. It seeks to combine the communities obtained during the previous phase in order to find the optimal communities structure at the current phase. The latter searches automatically the number of partitions, and requires no user intervention.

### 3. Link partitioning algorithms

The idea of partitioning links instead of nodes to discover community structure has also been explored. A node in the original graph is called *overlapping*, whenever links connected to it are put in more than one cluster. Ahn et al. proposed an overlapping community detection algorithm called *Link Community*, LC, that uses the similarity of the edges to identify hierarchical communities of edges rather than communities of nodes (Yong-Yeol et al. 2010). Given a pair of links  $e_{ik}$  and  $e_{jk}$  incident on a node  $k$ , a similarity can be computed through the Jaccard index as follows:

$$S(e_{ik}, e_{jk}) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}$$

where  $N_i$  is the neighborhood of node  $i$  including  $j$ .

In the state of the art, the structure of the modules differs from one approach to another. In fact, the output communities cannot be fully connected. Roughly speaking, all the nodes can not be necessarily strongly linked to each other. In addition, the network can be or not all covered by these communities.

Table 1 summarizes the different outputs of the above scrutinized approaches. The first column represents the different required input for each surveyed algorithm, which could be different from one algorithm to another. The second column describes the communities returned, which are not necessarily maximum bi-cliques and they do not guarantee to cover all the graph. The penultimate column shows the objective function used by each method. Finally, the fourth column describes the dependency between the different algorithm tasks, to wit dependent or independent. This characteristic allows to indicate whether it is possible to optimize the work and to treat these tasks in distributed manner. All of these algorithms are dependent tasks, i.e., the graph can not be split and the processing can not be distributed.

**Table 1** A comparison between the studied bipartite community detection approaches

Communities		Input	Not fully connected communities	Covering graph	Objective function	Tasks
<i>LPAb+</i>	Binary matrix	✓			Barber’s modularity	Dependent
<i>MDL-greedy</i>	Set of edges	✓	✓	✓	Bit length	Dependent
<i>LinkComm</i>	Set of edges Jaccard function	✓	✓	✓	Jaccard function	Dependent

The major moan that can be addressed to these algorithms stands in the absence of scalability, i.e., large graphs can not be processed. To palliate such a drawback, we introduce, through this paper, a new method that splits community detection processing into independent tasks.

### 4 The fast-bi community detection (FBCD) approach

This section contains the definition of the new approach designed to identify the pertinent community structure in bipartite networks using the maximum matching, called FBCD algorithm. This latter ensures the cover of all edges and vertex in the network. Therefore, we can determine each community existence through these critical edges, called maximum matching edges. It is worth mentioning that, all the links between these edges are disjoint, i.e., we can not find two links that share the same node. This point makes easier the distribution of the algorithm by treating each element independently. Concurrently, the search space will be reduced since the algorithm does not need to treat all the existing edges.

The proposed algorithm relies on a heuristic based on quality score optimization. This score is determined by the aggregation of four different criteria according to user needs. The latter criteria are as follows:

**Stability** (Roth et al. 2007): the stability metric for a given community  $\langle A, B \rangle$ , denoted by  $\sigma(\langle A, B \rangle)$ , describes the proportion of subsets of nodes in  $A$  whose closure is exactly equal to  $B$ . This metric reflects the dependency of  $B$  on particular nodes of  $A$ .

$$\sigma(\langle A, B \rangle) = \frac{|\{X \subseteq A \mid \psi(X) = B\}|}{2^{|A|}} \tag{2}$$

The higher the stability is, the higher the quality of the community.

**Example 6** Given the bipartite graph depicted in Fig. 4. The stability of community  $\langle 23, 4567 \rangle$  is computed as follows :  $A = \{2, 3\}, B = \{4, 5, 6, 7\}$

$$\begin{aligned} \psi(2) &= \{4, 5\} \\ \psi(3) &= \{4, 5, 6, 7\} = B \\ \psi(2, 3) &= \{4, 5, 6, 7\} = B \\ \sigma(\langle 23, 4567 \rangle) &= \frac{2}{(2^2)} = 0.5 \end{aligned}$$

**Modularity** (Newman and Girvan 2004): the modularity metric is defined as the ratio of difference between the actual number of edges within the community and expected number of edges in a randomized graph with the same number of nodes and the same degree sequence. A better community quality is assessed through a higher modularity.

$$\left( \frac{|E_c^{in}|}{|E|} \right) - \left( \frac{(2|E_c^{in}| + |E_c^{out}|)}{2 \times |E|} \right)^2 \tag{3}$$

**Example 7** Given the bipartite graph depicted in Fig. 4, the modularity of  $\langle 23, 4567 \rangle$  is computed as follows:  $E^{in} = 6, E^{out} = 2$  and  $|E| = 8$   $Mod = \left( \left( \frac{6}{8} \right) - \left( \frac{(2 \times 6 + 2)}{2 \times 8} \right)^2 \right) = 0.75 - 0.766 = -0.016$ .

**Bond** (Omiecinski 2003): the bond metric computes the ratio between the conjunctive<sup>1</sup> and the disjunctive<sup>2</sup> support. Thus, the bond measure of a community  $C = \langle A, B \rangle$  is defined as follows:

$$Bond(\langle A, B \rangle) = \frac{Supp(\wedge A)}{Supp(\vee A)} \tag{4}$$

**Example 8** Given the bipartite graph depicted in Fig. 4, the bond of  $\langle 23, 4567 \rangle$  is computed as follows :

$$\begin{aligned} Supp(\wedge A) &= |\psi(2) \cap \psi(3)| = |\{4, 5\}| = 2. \\ Supp(\vee A) &= |\psi(2) \cup \psi(3)| = |\{4, 5, 6, 7\}| = 4. \\ Bond(\langle 23, 4567 \rangle) &= \frac{2}{4} = 0.5. \end{aligned}$$

**Overlapping**: the overlapping metric is defined as the redundancy of each link in the extracted communities. A

<sup>1</sup>  $Supp(\wedge I) = \cap \phi(i), i \in I.$

<sup>2</sup>  $Supp(\vee I) = \cup \phi(i), i \in I.$

smaller value of overlapping induces a better quality of the community.

**Example 9** Given the bipartite graph depicted in Fig. 4. Let us suppose that we just returned the community  $\langle 12, 45 \rangle$  and that we are interested in computing the overlapping for the community  $\langle 23, 4567 \rangle$ . The edges that already exist in the returned communities are  $\{(2, 4); (2, 5)\}$ . So, the overlapping of  $\langle 23, 4567 \rangle$  is equal to  $|\{(2, 4); (2, 5)\}| = 2$ .

After computing the different measures for each community per iteration, the algorithm applies a method to compute an aggregated score for the different communities. In order to do that, the algorithm relies in the TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) method which is a multi-criteria decision analysis method, developed by Hwang et al. (1993). In this method, two artificial alternatives are hypothesized:

- *Ideal alternative* One which has the best attributes values.
- *Negative ideal alternative* One which has the worst attributes values.

Let  $x_{ij}$  be the evaluation of the community  $i$  according to the measure  $j$ , and  $w_j$ , the weight of measure  $j$ . The TOPSIS method operates in six steps described as follows:

**Step 1:** Standardize the decision-matrix

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}, i = 1 \dots m; j = 1 \dots n$$

**Step 2:** Construct the weighted standardize decision-matrix by multiplying attributes weight to each rating.

$$v_{ij} = w_j r_{ij}, i = 1 \dots m; j = 1 \dots n$$

where  $w_j$  is the weight of criteria  $j$ .

**Step 3:** Determine the ideal solution  $A^*$  and the negative ideal solution  $A^-$ .

$$A^* = \{v_1^*, v_2^* \dots v_j^* \dots v_n^*\} \\ = \{(\max_i v_{ij} | j \in J_1), (\min_i v_{ij} | j \in J_2) | i = 1 \dots m\}.$$

$$A^- = \{v_1^-, v_2^- \dots v_j^- \dots v_n^-\} \\ = \{(\min_i v_{ij} | j \in J_1), (\max_i v_{ij} | j \in J_2) | i = 1 \dots m\}.$$

where  $J_1$  is the set of criteria to be maximized and  $J_2$  is the set of criteria to be minimized.

**Step 4:** Determine the separation from the ideal solution:

$$S_i^* = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^*)^2}$$

**Step 5:** Determine the separation from the negative ideal solution:  $S_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2}$

**Step 6:** Compute the relative closeness to the ideal solution:  $C_i^* = \frac{S_i^-}{(S_i^* + S_i^-)}$

In terms of performance, the TOPSIS has been compared versus a number of other multi-attribute methods and was found to perform almost as well as multiplicative additive weights and better than analytic hierarchy process in matching a base prediction model (Zanakis et al. 1998).

### 4.1 Description of the proposed algorithm

Figure 5 sketches a diagram that describes the different steps of the FBCD algorithm. The pseudo-code of the FBCD algorithm is given by Algorithm 1.

The first step illustrated by part (1) of the Fig. 5 considers the set of maximum matching  $\mathcal{MM}$  and the initial network  $\mathcal{G}$  as input of the algorithm. Then, the Pseudo-Community  $PC$  of each couple in  $\mathcal{MM}$  are extracted (line 4 in Algorithm 1). This stride is needed to distribute the process in the following step described by part (2) of the Fig. 5. In this step, a distributed call to PSEUDO-COMMUNITY-DETECTION is made to extract the communities for each element in  $\mathcal{MM}$ . During the final step which is explained by the part (3) of the Fig. 5, the algorithm reduces the set of the already returned communities in a list of pertinent communities (line 7 Algorithm 1). This list is considered as the output of our algorithm (line 8 Algorithm 1).

---

#### Algorithm 1: Fast-Bi Community Detection (FBCD) algorithm

---

**Data:** The bipartite graph  $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ , the maximum matching set  $\mathcal{MM}$   
**Result:** The list of pertinent communities  $\mathcal{F}_{\mathcal{G}}$

```

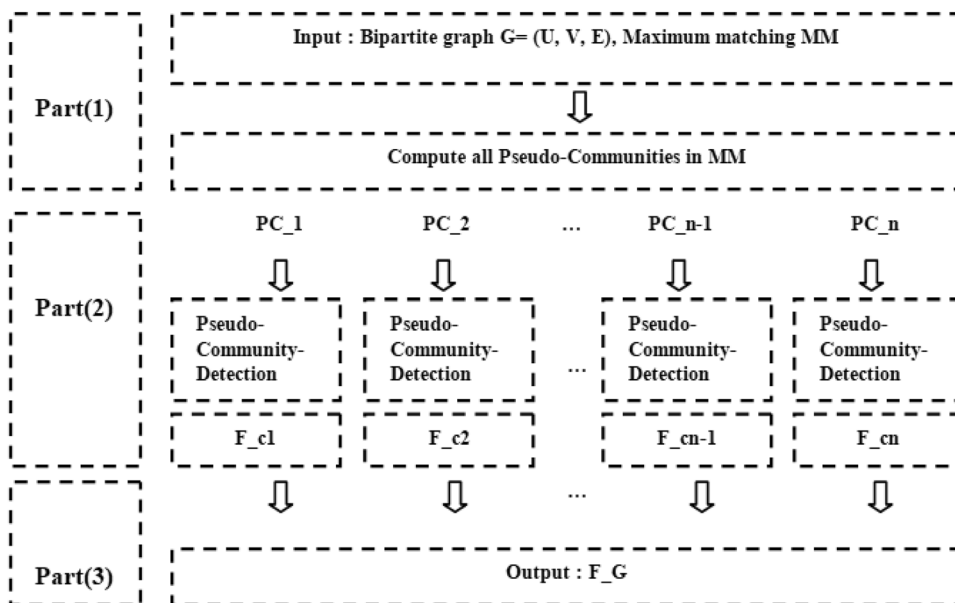
begin
1   $\varphi \leftarrow \{(a, b) \in \mathcal{MM}\};$ 
2   $\mathcal{F}_{\mathcal{G}} \leftarrow \emptyset;$ 
3  foreach couple  $(a, b) \in \varphi$  do
4     $PC_{(a,b)} = \text{GET\_PSEUDO\_COMMUNITY}(\mathcal{G}, a, b);$ 
5  foreach couple  $(a, b) \in \varphi$  do
6     $\text{PSEUDO-COMMUNITY-DETECTION}(PC_{(a,b)});$ 
7  Reduce  $(\mathcal{F}_{\mathcal{G}});$ 
8  return  $\mathcal{F}_{\mathcal{G}};$ 

```

---

The Algorithm 2 describes the Pseudo-Community-Detection step, which takes as input a bipartite graph  $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$  associated to current pseudo-community, as well as the quality metrics i.e., stability, modularity, overlapping and bond. It outputs a community partition of the current pseudo-community  $PC$ . The pseudo-code of this part is given by algorithm 2. In a first step,  $\varphi$  is set to all couples  $(u, v)$  of  $\mathcal{E}$  (line 1), and the community partition  $\mathcal{F}_{\mathcal{G}}$  is initialized to the empty set (line 2). Then, our algorithm invokes the GET\_PSEUDOCOMMUNITY function (line 4) in

**Fig. 5** Diagram describing the different steps of the FBCD algorithm



order to compute the pseudo-community for each couple belonging to  $\varphi$ .

In the next step, the GET\_DENSITY function (line 5) assesses the density  $\omega$  for each pseudo-community  $PC_{(u,v)}$ . Afterwards, a decreasing sort is carried out on the couples in  $\varphi$  through the SORT\_ELEMENTS function (line 6). Then, for each element belonging to  $\varphi$ , the algorithm extracts a local pertinent maximal bi-clique from its pseudo-community  $PC_{(u,v)}$ .

The pertinent maximal bi-clique is obtained through the BUILD\_COMMUNITY function (line 9). In the case where  $\omega$  ( $PC_{(u,v)}$ ) is equal to 1, then the latter is reduced to a maximal bi-clique. Otherwise, our algorithm proceeds to the extraction of all the local maximal bi-cliques  $S_c$  enclosed into  $PC_{(u,v)}$  by calling the function named GET\_ALL\_MAX\_BICLIQUES (line 10). It is worth citing that the extraction of the maximal bi-cliques is carried by a slightly modified version of the very efficient LCM algorithm (Uno et al. 2004). The choice of this algorithm is argued by the fact that it has a linear complexity in the number of closed attributes. Moreover, it has been shown to be one of the best algorithms dedicated to such a task. After that, the GET\_METRICS procedure (line 12) is invoked in order to compute, for each maximal bi-clique in  $S_c$ , the associated metrics values. The GET\_AGGREGATION function (line 13) is invoked in order to compute for each maximal bi-clique its corresponding score. To do so, we use the multi-criteria aggregation method TOPSIS (Hwang and Yoon 1981). Depending on the value of the score measure, the algorithm elects the pertinent maximal bi-clique, through the GET PERTINENT\_COMMUNITY function (line 14). During the current iteration, the chosen pertinent community is added to the list of pertinent communities  $\mathcal{F}_G$  (line 15). Then, the couples included in the chosen community are removed

from  $\varphi$  before going through the REMOVED\_LINKS function (line 16). The algorithm comes to an end whenever  $\varphi$  list is exhausted and returns the final set of communities  $\mathcal{F}_G$  (line 17).

**Algorithm 2: PSEUDO-COMMUNITY-DETECTION**

```

Data: The Pseudo-Community  $\mathcal{G} = (U, V, \mathcal{E})$ 
Result: The list of pertinent communities  $\mathcal{F}_G$ 
begin
1   $\varphi \leftarrow \{(u, v) \in \mathcal{E}\};$ 
2   $\mathcal{F}_G \leftarrow \emptyset;$ 
3  foreach couple  $(u, v) \in \varphi$  do
4     $PC_{(u,v)} = \text{GET\_PSEUDO\_COMMUNITY}(\mathcal{G}, u, v);$ 
5     $\omega = \text{GET\_DENSITY}(PC_{(u,v)});$ 
6  SORT_ELEMENTS( $\varphi, \omega$ );
7  while  $\varphi \neq \emptyset$  do
8    if  $(\omega(PC_{(u,v)}) == 1)$  then
9       $\langle A, B \rangle = \text{BUILD\_COMMUNITY}(PC_{(u,v)});$ 
10     else
11        $S_c = \text{GET\_ALL\_MAX\_BICLIQUES}(PC_{(u,v)});$ 
12       foreach  $c \in S_c$  do
13         GET_METRICS( $c, \text{Stability, Modularity, Overlapping, Bond}$ );
14          $\text{Score} = \text{GET\_AGGREGATION}(c);$ 
15          $\langle A, B \rangle = \text{GET\_PERTINENT\_COMMUNITY}(S_c, \text{Score});$ 
16        $\mathcal{F}_G = \mathcal{F}_G \cup \langle A, B \rangle;$ 
17       REMOVE_LINKS( $\varphi, \langle A, B \rangle$ );
18 return  $\mathcal{F}_G;$ 

```

**4.2 Complexity analysis of FBCD**

The complexity of the FBCD algorithm is depends on that of PSEUDO-COMMUNITY-DETECTION part. This last is assessed as follows:

Let  $n = |V_1|$  and  $m = |V_2|$  be, respectively, the number of vertex for each set of the bipartite graph.

1. First part of PSEUDO-COMMUNITY-DETECTION: the complexity of this part depends on the complexity of the

two functions GET\_PSEUDOCOMMUNITY and GET\_DENSITY (lines 3-5).

2. Second part of PSEUDO-COMMUNITY-DETECTION: we have chosen the QUICKSORT algorithm to sort elements  $(u, v)$  of  $\omega$  which has a complexity of  $O(n \log(n))$  (line 6).
3. Third part of PSEUDO-COMMUNITY-DETECTION: the complexity of this part depends on the number of iterations of the “while” loop (line 7). In fact, the maximal number of iterations is estimated at  $t_1 = \text{Max}(n, m)$  (lines 7–16). The process done by this part, can also be split into three subparts:
  - (a) lines 8–9: the complexity of this part is equal to that of the function BUILD\_COMMUNITY( $PC_{(u,v)}$ ). The latter is about  $O(n)$  in terms of number of iterations ( $|\psi(u)| + |\phi(v)| - 1$ ).
  - (b) lines 10–14: The GET\_ALL\_MAX\_BICLIQUES function invokes the LCM algorithm. The time complexity of the latter is theoretically bounded by a linear function in the number of frequent closed attributes (Uno et al. 2005). Indeed, it enumerates all frequent closed pattern to derive its closure. So, its complexity is equal to  $O(n^2)$ . Then, assessing the metrics used for selecting the pertinent community has a complexity  $O(t_2)$ , where  $t_2 = \text{Max}(|\psi(u)|, |\phi(v)|)$ . After that, the TOPSIS method used to aggregate the metrics, has a complexity of  $O(t_2)$  per iteration. Thus, the complexity of the second subpart is then about  $O(t_1^2)$ .
  - (c) lines 15–16: The final subpart of our algorithm is about  $O(|A| * |B|)$ .

To sum up, we can conclude that the complexity of the FBCD algorithm is determined by summing the corresponding complexity of its three parts of the PSEUDO-COMMUNITY-DETECTION, which are treated in a distributed processing. Finally, we can say that the theoretical complexity of our algorithm is about  $n \log(n) + n * [2n + n^2] + n^2 = n^3 + 3n^2 + n \log(n) = O(n^3)$ .

## 5 Experimental results

This section presents a detailed study of the performance of the proposed algorithm. thus, we briefly introduce the datasets and the real-world networks used during this study. Then, we discuss the outputs of the obtained results.

### 5.1 Real-world networks and datasets

In our approach, we used 20 different datasets extracted from 3 well-known repositories: KONECT newtworks, Network Repository and SNAP dataset. A brief summary of all the

datasets can be observed in Table 2. This table contains for each dataset, the following informations: the repository where the dataset can be found (Source column), the application domain of the dataset, it’s name, and the characteristics defining the dataset. Note that for the ‘Domain’ column, we used the following acronyms: Social Networks (SN); Web Graphs (WG); Network Dataset (ND); Brain Network (BN); Collaboration Network (CN); Recommendation Network (RN) and Citations Network (CIT).

1. KONECT networks<sup>3</sup>: For testing our first contribution, We use four datasets from Konect databases: Southern Women (Davis et al. 2009), American Revolution, Corporate Leadership, South Africa Companies. The “Southern women” network collected by Davis et al. Davis et al. (2009)<sup>4</sup> shows the participation of 18 white women (who form the primary set  $U$ ) in 14 social events (the secondary set  $V$ ) over a nine-month period. The data was collected in the Southern United States of America in the 1930s. There is an edge for every woman who participates in an event. The first column contains the women, the second column contains the events. The “American Revolution”<sup>5</sup> contains membership information of 136 people (forming the primary set  $U$ ) in 5 organizations dating back (the secondary set  $V$ ) to the time before the American Revolution. The list includes well-known people such as the American activist Paul Revere. Left nodes represent persons and right nodes represent organizations. An edge between a person and an organization shows that the person was a member of the organization. The “Corporate Leadership”<sup>6</sup> contains person-company leadership information between 20 companies (forming the primary set  $U$ ) and 24 corporate directors (the secondary set  $V$ ). The data was collected in 1962. Left nodes represent persons and right nodes represent companies. An edge between a person and a company shows that the person had a leadership position in that company. The “South African Companies”<sup>7</sup> contains person-company shared leadership relations of “the five most representative companies” that are claimed to represent “the small inner ring of South African Finance”. Left nodes represent persons (the primary set  $U$ ) and right nodes represent companies (the secondary set  $V$ ). An edge between a person and a company shows that the person had a leadership position in that company.

<sup>3</sup> <http://konect.uni-koblenz.de/>.

<sup>4</sup> Southern women network dataset-KONECT (2016).

<sup>5</sup> American Revolution network dataset-KONECT (2016).

<sup>6</sup> Corporate Leadership network dataset-KONECT (2016).

<sup>7</sup> South African Companies network dataset-KONECT (2016).



**Table 2** The datasets and the real-world networks used for experiments

Source	Domain	Dataset	$\ln I_1$	$\ln I_2$	$ E $
KONECT networks	SN	South Africa	6	5	13
		Souther women	18	14	89
		Corporate leadership	20	24	99
		American revolution	136	5	160
Network repository	SN	soc-karate	34	34	78
		soc-Dolphins	62	62	159
	WG	Web-polblogs	643	643	2280
		Web-edu	3031	3031	6474
		Web-EPA	4773	4773	8965
	ND	DD-g522	736	736	3482
	BN	bn-macaque-rhesus-brain-1	241	341	4090
	CN	ca-Erdos992	6100	6100	7515
		ca-GrQc	4158	4158	13,422
	SN	socfb-nips-ego	2981	2888	2888
		socfb-Caltech36	769	769	16,656
		socfb-Cal65	11,247	11,247	351,358
		socfb-Bingham82	10,004	10,004	351,358
	RN	rec-amazon	91,813	91,813	125,704
SNAP	CIT	Cit-HepTH	27,770	27,770	352,807
		Cit-HepPh	34,546	34,546	421,578

2. SNAP networks<sup>8</sup>: We choose two networks with ground truth communities collected by SNAP[38]: Cit-HepPh and Cit-HepTh. These latter indicate the relation between citations and the ground truth communities are paper defined groups. If a paper  $i$  cites paper  $j$ , then the graph contains a directed edge from  $i$  to  $j$ . If a paper cites, or is cited by, a paper outside the dataset, the graph does not contain any information about this.
3. Network Repository<sup>9</sup>: A network repository is a logical and physical grouping of data from related but separate network. In these cases, a repository is necessary to bring together the discrete data items and operate on them as one. We used different categories of this type of network, such as: Social Networks, Web Graphs, Network Datasets, Brain Networks, Collaboration Networks, Facebook Networks and Recommendation Networks.

## 5.2 Experiments on real-world networks

In the following, we start by presenting the quality metrics of use to assess the performance of the introduced algorithm.

### 5.2.1 Quality metrics

In the following, we put the focus on the evaluation of the performances of the FBCD algorithm using various metrics,

such as the modularity (Murata 2009), conductance (Kannan et al. 2000) and density (Viard and Latapy 2014). These metrics describe how community-like is the connectivity structure of a given set of nodes. Indeed, they rely on the fact that communities are sets of nodes with many internal edges and few external ones. Thus, given a network  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  and a community or a set of nodes  $C$ . The number of nodes in the community is set to  $|C|$  and  $|E_C^{in}|$  presents the total number of edges in  $C$  for unweighted networks or the total weight of the edges for weighted networks. In addition, we denote by  $|E_C^{out}|$  the total number of edges from the nodes in community to the nodes outside  $C$  for unweighted networks or the total weight of such edges for weighted networks. In the following, we review the following metrics.

1. *Modularity* This metric was designed to assess the strength of division of a network into communities. Indeed, networks with high modularity have dense connections between the nodes within communities but sparse connections between nodes in different communities. According to Table 3, we can detect the different results between the FBCD algorithm and those of the surveyed algorithms of the literature. The following results are obtained on four medium-sized graphs. As shown by Table 3, the worst results on average were obtained by the MDL-GREEDY algorithm. For the Corporate Leadership graph, the latter gives the same result as the FBCD algorithm which is the highest obtained value for this graph. Even though the LPAb+ algorithm does

<sup>8</sup> <http://snap.stanford.edu>.

<sup>9</sup> <http://networkrepository.com>.

**Table 3** The quality of returned communities in terms of the modularity metric

	LPAb+	MDL-GREEDY	LC	FBCD
Southern women	0.05	0.02	0.01	0.10
American revolution	0.17	0.17	0.33	0.17
Corporate leadership	0.04	0.06	0.02	0.06
South Africa	0.14	0.12	0.14	0.20
Average	0.10	0.09	0.12	0.14

**Table 4** The quality of returned communities in terms of the conductance metric

	LPAb+	MDL	LinkComm	FBCD
Southern women	0.56	0.73	0.66	0.48
American revolution	0.19	0.19	0.21	0.19
Corporate leadership	0.55	0.14	0.67	0.50
South Africa	0.35	0.59	0.36	0.32
Average	0.39	0.41	0.47	0.37

not yield any highest results for all the graphs, its corresponding results are better than those obtained by the MDL-GREEDY algorithm. On average, the highest value equal to 0.14, is yielded by the FBCD algorithm which outperforms all its competitors.

2. **Conductance** The detected communities can also be assessed through the conductance metric. The latter is based on the density of communities and the number of links emerging from them. A structure community is supposed to flag out a high number of links within it and a weak number of outbound links. The conductance metric is based on the ratio of the number outbound links,  $E_C^{out}$  and the total number of links (inside the  $E_C^{in}$ ) for a community C. If we consider a community C of a graph  $G = (V_1, V_2, E)$ , with  $C = (V_C, E_C)$  ( $V_C$  the set of vertexes of C and  $E_C$  the set of edges of C), the conductance of this community is defined by  $\Psi(C, G) = \frac{|E_C^{out}|}{2|E_C^{in}| + |E_C^{out}|}$ . Considering a partition  $P = C_1, \dots, C_k$  into  $k$  parts of disjoint nodes, the conductance of  $G$  is defined as follows:  $\Psi_G = \frac{1}{k} \sum_{C=1}^k (\Psi(C, G))$ ,  $\Psi_G = \frac{1}{k} \sum_{C=1}^k \left( \frac{|E_C^{out}|}{2|E_C^{in}| + |E_C^{out}|} \right)$ . The conductance values stand within the unit interval. The closer this value to 0, the higher community density is. In this respect, Table 4 illustrates a comparison between our algorithm versus its competitors. We note that the

that our algorithm outperforms its competitors in terms of conductance.

3. **Density** We start by discussing the intra-community density. The latter is defined as the number of existing edges over the number of edges that could exist in community. Plainly speaking, it is the probability that two nodes chosen at random in from the two sets  $V_{C1}, V_{C2}$  the same community are linked together. Considering a community  $c$  of a graph  $G = (V_1, V_2, E)$ , with  $C = \{V_{C1}, V_{C2}, E_C\}$  ( $V_{C1}$  (resp.  $V_{C2}$ ) the set of vertexes of C in  $V_1$  (resp.  $V_2$ ) and  $E_C$  the set of edges of C), the density of this community is equal to  $MQ^+(C) = \frac{|E_C^{in}|}{|V_{C1}| * |V_{C2}|}$ . Considering a partition  $P = C_1, \dots, C_k$  into  $k$  parts of disjoint nodes, the density of  $G$  is defined as follows:  $MQ_G^+ = \frac{1}{k} \sum_{C=1}^k (MQ^+(C))$ ,  $MQ_G^+ = \frac{1}{k} \sum_{C=1}^k \left( \frac{|E_C^{in}|}{|V_{C1}| * |V_{C2}|} \right)$ . The intra-community density can have a value between 0 and 1. A large value is better than a small one in terms of the community quality assessment. The inter-community density is the probability that two different nodes, chosen at random, in two different communities are linked together. So, given a graph  $G = (V_1, V_2, E)$  and a partition  $P = C_1, \dots, C_k$  into  $k$  communities, the inter-community density is defined by the ratio between the number of edges connecting vertexes of communities  $C_i$  and  $C_j$  and the maximum possible number of such edges:

$$MQ_{ci,cj}^- = \frac{|(v_{i1}, v_{j2}); v_{i1} \in V_{i1}, v_{j2} \in V_{j2}, (v_{i1}, v_{j2}) \in E| + |(v_{j1}, v_{i2}); v_{j1} \in V_{j1}, v_{i2} \in V_{i2}, (v_{j1}, v_{i2}) \in E|}{|V_{i1}| * |V_{j2}| + |V_{j1}| * |V_{i2}|}$$

$$MQ_G^- = \frac{1}{k(k-1)/2} \sum_{i=1}^{k-1} \sum_{j=1}^k (MQ_{ci,cj}^-)$$

linkComm algorithm gives bad values for all of the considered datasets, even though it has shown very good performances in terms of modularity. If we glance on the values for the American Revolution graph, the optimal is given by all the other algorithms; that is to say that the three approaches give the same detections. In this case, we can deduce the optimal detection for the American Revolution graph. Finally, according to Table 4, we find

The inter-community density values stand within the unit interval. A weak value is preferable to a large one in terms of the community quality assessment.

The main trust of the density metric is to assess the average density of communities to the density of edges between communities:  $MQ_G = MQ_G^+ - MQ_G^-$ . The quality of the output communities depends on a higher intra-community density  $MQ_G^+$  and a lower inter-community density  $MQ_G^-$ . The density range is between  $-1$  and  $1$ .

**Table 5** The quality of returned communities in terms of the density metric

		LPA <sub>b</sub> +	MDL	linkComm	FBCD
Southern women	Inter	0.79	0.95	0.85	1.00
	Inter	0.31	0.27	0.45	0.32
	Density	0.48	0.67	0.41	0.68
American revolution	Inter	1.00	0.80	0.81	1.00
	Inter	0.14	0.06	0.57	0.14
	Density	0.86	0.74	0.24	0.86
Corporate leadership	Inter	0.93	0.21	0.94	1.00
	Inter	0.11	0.20	0.31	0.22
	Density	0.80	0.01	0.64	0.78
South Africa	Inter	1.00	0.67	0.92	1.00
	Inter	0.14	0.26	0.47	0.21
	Density	0.86	0.40	0.46	0.79
Average	Inter	0.93	0.66	0.88	1.00
	Inter	0.17	0.20	0.45	0.22
	Density	0.75	0.46	0.43	0.78

Table 5 shows the density of obtained communities. In fact, our algorithm maximizes the intra-community density for all the considered graphs. Even though the inter-community is not minimized by an acceptable value is obtained in average, to wit 0.78. This result shows the coherence of the obtained communities versus those obtained by its competitors.

### 5.2.2 Processing time

Finally, we analyze the performance of the proposed algorithm using the maximum matching. To do so, we present a comparative study using a set of sample networks and some of the algorithms studied in Section 3. According to Table 6, the community detection is almost impossible with large datasets. Furthermore, Table 6 shows that our algorithm is more efficient than those of the literature in terms of execution time. Using the networks given by KONECT datasets, the FBCD algorithm provides the highest value on average which is equal to 0.01. Clearly, our algorithm outperforms MDL, LPA + and linkComm which respectively provide 1.197, 5.831 and 0.049.

BCD is supposed to handle all the existing edges in the network and we compare this hypothesis with our new algorithm. The results of maximum matching presented by Table 7. The column FBCD contains the obtained results by the algorithm using the maximum matching. Whereas the penultimate column (BCD) corresponds to the execution of the algorithm without considering the maximum matching. According to Table 7, we find that the search space of the new algorithm FBCD is very limited

**Table 6** The quality of returned communities in terms of processing time in second

	LPA <sub>b</sub> +	MDL	LinkComm	FBCD
Southern women	0.782	5.281	0.053	0.011
American revolution	1.611	7.860	0.081	0.015
Corporate leadership	1.304	7.871	0.030	0.009
South Africa	1.090	2.310	0.030	0.005
Average	1.197	5.831	0.049	0.010

**Table 7** Analysis of the influence of the maximum matching for both of the BCD and the FBCD algorithms

DataBase	IEI	IMMI	BCD	FBCD
Network repository				
Soc-Karate	78	14	0.242	<b>0.151</b>
Soc-Dolphins	159	43	0.545	<b>0.151</b>
Web-polblogs	2280	279	44.734	<b>1.152</b>
Socfb-nips-ego	2888	17	11.390	<b>0.110</b>
bn-macoquerehesus-brain-1	4090	218	2812.320	<b>85.402</b>
DD-g522	3482	737	429.554	<b>1.385</b>
Web-edu	6474	741	23.128	<b>1.559</b>
ca-Erdos992	7515	246	32.651	<b>2.528</b>
Web-EPA	8965	543	484.056	<b>2.341</b>
ca-GrQc	13422	530	1886.380	<b>10.341</b>
socfb-Caltech36	16656	659	1115.750	<b>224.994</b>
rec-amazon	125704	26671	987651	<b>1108.7</b>
socdb-Cal65	351358	<b>131</b>	1,034,140	3566
soc-fb-Bigham82	362894	111	1,175,800	<b>326.299</b>
SNAP networks				
Cit-HepTH	352807	970	1,978,332	<b>9776.6</b>
Cit-HepPH	421578	1760	2,165,660	<b>15180.9</b>

Bold values indicate the best solutions in terms of maximum matching

compared to that of BCD. Indeed, the latter explores all the edges. The same process is done by all the surveyed approaches. This characteristic is valid for large networks as well as small ones. This fact is clearly stressed by both of the large datasets *socfb-Cal65* and *socfb-Bingham82*, which have respectively 131 and 111 edges of maximum matching.

## 6 Conclusion

In this article, we propose a new paradigm of community structure for social network analysis and community detection in bipartite graph. We presented a formal definition

of the concept of 'community structure', and proposed a systematic algorithm called FBCD to discover these communities.

We conduct a comprehensive benchmarking study on approaches to community detection in social networks. Through these extensive experiments, we demonstrate that community structure exists in real-world networks of various domains. Our proposed method significantly outperforms those of the literature including modularity-based algorithms, minimum description length methods and link partitioning approaches. Avenues of future work are as follows:

1. *Consideration of other types bipartite networks* We are currently about exploring the extraction of communities from other types of networks, e.g. directed, weighted or dynamic networks. In fact, we provide to apply our community detection algorithm as the basis in order to design a new method to find the community structure in this new category of networks.
2. *Scalability* The considered datasets are not the most representative ones of the era of Big data. It is also a compelling task to provide an implementation under the distributed framework Spark.

**Acknowledgements** This work has been funded by the Justice Program of the European Union (2014–2020) 723180—RiskTrack—JUST-2015-JCOO-AG/JUST-2015-JCOO-AG-1. The contents of this publication are the sole responsibility of their authors and can in no way be taken to reflect the views of the European Commission.

## References

- Asratian AS, Denley TMJ, Häggkvist R (1998) Bipartite graphs and their applications. Cambridge University Press, New York
- Barber MJ (2007) Modularity and community detection in bipartite networks. *Phys Rev E* 76(6):066102
- Barbut M, Monjardet B (1970) Ordre et classification. *Algèbre et Combinatoire*, Hachette, Tome II
- Ben Yahia S, Mephu Nguifo E (2004) Approches d'extraction de règles d'association basées sur la correspondance de galois. *Ingénierie des Systèmes d'Information* 9(3–4):23–55
- Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. *J Stat Mech: Theory Exp* 10:P10008
- Coscia M, Rossetti G, Giannotti F, Pedreschi D (2012) DEMON: a local-first discovery method for overlapping communities. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, pp 615–623
- Dai T, Zhu L, Cai X, Pan S, Yuan S (2018) Explore semantic topics and author communities for citation recommendation in bipartite bibliographic network. *J Ambient Intell Human Comput (JAIHC)* 9(4):957–975
- Davis A, Gardner BB, Gardner MR (2009) Deep south: a social anthropological study of caste and class. University of South Carolina Press, Southern classics series
- Gibbons A (1985) Algorithmic graph theory. Cambridge University Press, Cambridge
- Girvan M, Newman ME (2002) Community structure in social and biological networks. *Proc Natl Acad Sci* 99(12):7821–7826
- González-Pardo A, Jung JJ, Camacho D (2017) Aco-based clustering for ego network analysis. *Future Gener Comput Syst* 66:160–170
- Guimerà R, Sales-Pardo M, Amaral LAN (2007) Module identification in bipartite and directed networks. *Phys Rev E* 76(3):036,102
- Hamrouni T, Ben Yahia S, Mephu Nguifo E (2008) Succinct minimal generators: theoretical foundations and applications. *Int J Found Comput Sci* 19(02):271–296
- Hwang CL, Yoon K (1981) Multiple attribute decision making: methods and applications, vol 186. Springer, Berlin
- Hwang CL, Lai YJ, Liu TY (1993) A new approach for multiple objective decision making. *Comput Operat Res* 20:889–899
- Jelassi MN, Largeron C, Ben Yahia S (2014) Efficient unveiling of multi-members in a social network. *J Syst Softw* 94:30–38
- Kannan R, Vempala S, Vetta A (2000) On clusterings-good, bad and spectral. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science, IEEE Computer Society
- Lancichinetti A, Radicchi F, Ramasco JJ, Fortunato S (2010) Finding statistically significant communities in networks. *CoRR abs/1012.2363*
- Mouakher A, Ben Yahia S (2019) On the efficient stability computation for the selection of interesting formal concepts. *Inf Sci* 472:15–34
- Mouakher A, Yahia SB (2016) Qualitycover: efficient binary relation coverage guided by induced knowledge quality. *Inf Sci* 355–356:58–73
- Mouakher A, Ktayfi O, Ben Yahia S (2019) Scalable computation of the extensional and intensional stability of formal concepts. *International Journal of General Systems*
- Mucha M, Sankowski P (2004) Maximum matchings via gaussian elimination. In: Proceedings of the 45th IEEE Symp. Foundations of Computer Science FOCS, IEEE Computer Society, pp 248–255
- Murata T (2009) Community division of heterogeneous networks. In: Proceedings of 1st international conference complex sciences: theory and applications. Springer, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 4, pp 1011–1022
- Newman ME (2003) The structure and function of complex networks. *SIAM Rev* 45(2):167–256
- Newman ME, Girvan M (2004) Finding and evaluating community structure in networks. *Phys Rev E* 69(026113):026113
- Omiecinski ER (2003) Alternative interest measures for mining associations in databases. *IEEE Trans Knowl Data Eng* 15(1):57–69
- Pons P, Latapy M (2006) Computing communities in large networks using random walks. *J Graph Algorithms Appl* 10(2)
- Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. *Phys Rev E* 76(3):036106
- Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. *Proc Natl Acad Sci* 105(4):1118–1123
- Roth C, Obiedkov S, Kourie DG (2007) Towards concise representation for taxonomies of epistemic communities. In: Proceedings of the 4th international conference on concept lattices and their applications (CLA), Springer, Lecture Notes in Computer Science, vol 4923, pp 240–255
- Uno T, Asai T, Uchida Y, Arimura H (2004) An efficient algorithm for enumerating closed patterns in transaction databases. In: Proceedings of the 7th international conference discovery science, DS 2004, Padova, Italy, pp 16–31
- Uno T, Kiyomi M, Arimura H (2005) Lcm ver.3: Collaboration of array, bitmap and prefix tree for frequent itemset mining. In: Proceedings of the 1st international workshop on open source data

- mining: frequent pattern mining implementations, ACM, New York, NY, USA, OSDM '05, pp 77–86
- Viard J, Latapy M (2014) Identifying roles in an IP network with temporal and structural density. In: Proceedings of the IEEE INFOCOM workshops, Toronto, ON, Canada, pp 801–806
- Xu K, Tang C, Li C, Jiang Y, Tang R (2010) An MDL approach to efficiently discover communities in bipartite network. In: Proceedings of the 15th international conference database systems for advanced applications, Springer, Lecture Notes in Computer Science, vol 5981, pp 595–611
- Yang J, Leskovec J (2012) Community-affiliation graph model for overlapping network community detection. In: Proceedings of the 12th IEEE international conference on data mining, ICDM 2012, Brussels, Belgium, pp 1170–1175
- Yong-Yeol A, Bagrow JP, Lehmann S (2010) Link communities reveal multiscale complexity in networks. *Nature* 466:761
- Zanakis SH, Solomon A, Wishart N, Dubliss S (1998) Multi-attribute decision making: a simulation comparison of select methods. *Eur J Oper Res* 107(3):507–529

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.