Triangle-Driven Community Detection in Large Graphs Using Propositional Satisfiability

Said Jabbour, Nizar Mhadhbi, Badran Raddaoui, Lakhdar Saïs

Triangle-Driven Community Detection in Large Graphs Using Propositional Satisfiability

Said Jabbour
CRIL - CNRS UMR 8188
University of Artois
Lens, France
jabbour@cril.fr

Nizar Mhadhbi
CRIL - CNRS UMR 8188
University of Artois
Lens, France
mhadhbi@cril.fr

Badran Radaoui
SAMOVAR, Telecom SudParis, CNRS
University of Paris-Saclay
Paris, France
badran.raddaoui@telecom-sudparis.eu

Lakhdar Sais
CRIL - CNRS UMR 8188
University of Artois
Lens, France
sais@cril.fr

***Abstract*—Discovering the latent community structure is crucial to understanding the features of networks. Several approaches have been proposed to solve this challenging problem using different measures or data structures. Among them, detecting overlapping communities in a network is an usual way towards network structure discovery. It presents nice algorithmic issues, and plays an important role in complex network analysis. In this paper, we propose a new approach to detect overlapping communities in large complex networks. First, we introduce a novel subgraph concept based on triangles to capture the cohesion in social interactions, and propose an efficient approach to discover clusters in networks. Next, we show how the problem of detecting overlapping communities can be expressed as a Partial Max-SAT optimization problem. Our comprehensive experimental evaluation on publicly available real-life networks with ground-truth communities demonstrates the effectiveness and efficiency of our proposed method.**

***Index Terms*—Social Networks, Community Detection, Propositional Satisfiability.**

I. Introduction

Many real world problems can be modeled as complex entity-relationship networks where nodes represent entities of interest and edges mimic the relationships among them. Such connections might represent different type of relations between individuals or entities. Fueled by technological advances and inspired by empirical analysis, the number of such problems and the diversity of domains from which they arise is growing steadily, including among others physics, sociology, biology, chemistry, metabolism and nutrition. The study of such networks can help us understand the structure and functionalities of such systems, potentially allowing one to predict interesting aspects of their behavior. Extracting *communities* (or clusters) is of particular interest in many of these applications. Community detection is the process of grouping a set of entities in such a way that entities in the same group (cluster) are more similar to one another than to the ones in other groups. As an example, in the graph of the World Wide Web (W3), community detection aims to find groups of pages dealing with the same or related topics. Also, in the context of social network, community detection identifies groups, which are closely related in such a way that inside each group entities share the same interests or the same topic. In recent years, there has been a surge of research interests on finding communities in networks.

From the point of view of the topological structure of the graphs, community detection forms groups of nodes with dense intra-groups connection and sparse intergroup connection. Generally, real networks involve large number of triangles, as communities involve highly connected vertices.

Our proposed overlapping communities detection framework heavily exploits such triangle-based subgraph structure and modeled in a declarative way as a partial maximum satisfiability optimisation problem.

Such formulation allows us to benefit from the recent advances in propositional satisfiability and its optimisation variants. Our proposed framework follows the recent data mining research trend exploiting two powerful declarative models, namely constraint programming and propositional satisfiability. Indeed, several data mining tasks including pattern mining [1] and clustering [2], [3] have been modeled and solved using these two well-known declarative and flexible models.

The paper is organized as follows. We first discuss some related work in Section II. Then, we introduce some preliminary definitions about propositional satisfiability and community detection with some metrics that we will use to evaluate the quality the detected communities (Section III). Our declarative triangle-based framework is described in Section IV. An extensive and comparative experimental evaluation on many real-world datasets is presented (Section V) before concluding.

II. Related Work

Related work can be classified in two categories: *non-overlapping community detection*, and *overlapping community detection*.

Early approaches such as random walks, spectral partitioning, hierarchical clustering, modularity maximization, differential equations, and statistical mechanics have all been used to identify disjoint communities. This type of detection assumes that the network can be partitioned into dense regions in which nodes have more connections to each other than to the rest of the network. For details on different non-overlapping community detection methods, readers are referred to [4] and [5].

Unlike previous clustering algorithms assuming that communities are mutually disjoint, many authors have made the observation that a node in real-world networks may participate in more than one cluster. Thus, there is growing interest in overlapping community detection algorithms that identify a set of clusters that are not necessarily disjoint.

There exist many different methods for identifying overlapping communities. These algorithms can be categorized into five classes which reflect how communities are identified.

- *Clique Percolation*: This Method is based on the assumption that a community consists of overlapping sets of fully connected subgraphs and detects communities by searching for adjacent cliques. Various algorithms have been introduced, including CFinder [6], Clique Percolation Method with weights [7], and Sequential Clique Percolation algorithm [8].
- *Line Graph and Link Partitioning*: In link clustering based algorithms, the edges between the nodes are partitioned and not the nodes themselves to discover community structure. A node is a part of more than one cluster if the edges connecting it are in different clusters. Several measures characterizing similarity between edges [9] and algorithms have been proposed [10], [11]
- *Local Expansion and Optimization*: These algorithms are based on growing a natural community or a partial community. Some of them rely on a local benefit function that characterizes the quality of a densely connected group of nodes. Among the approaches that belong to this class, we can cite [12]–[15] Some other algorithms in this category are based on the following principe: starting from a number of a seed nodes (sets) and expand them into communities by examining and analyzing only neighborhood of the seeds [16]. However, the choice of the seed nodes plays a very important role to obtain a high coverage of nodes and to produce clusters of nodes with high performance.
- *Agent-Based and Dynamical Algorithms*: A range of different community detection methods have been proposed. In particular, COPRA [17] and SLPA [18] extend LPA to detection of overlapping communities by allowing multiple labels to a node. A game-theoretic framework is proposed in [19], in which each node is modeled as a rational agent trying to optimize its own utility by joining or leaving communities.
- *Nonnegative Matrix Factorization*: Another line of work in addressing the overlapping community is based on extensions of Nonnegative Matrix Factorization. It is a feature extraction and dimensionality reduction technique in machine learning that has been adapted to community detection [20]–[22].

Our work is related to the second category of community detection. The proposed declarative method follows a different track by defining an alternative strategy for the detection of overlapping communities. This complementary approach follows from the need to satisfy the following features:

- *Declarative*: to support the high-level and natural modeling of community discovering tasks; that is, our approach should closely correspond to the definitions of community detection problems found in the literature;
- *Generic*: to be solver-independent, such that the best SAT solving method can be selected for the problem and data at hand. Supported methods should include both general purpose solvers and specialized efficient mining algorithms.

Our goal is to capitalize and extend the state-of-the-art community detection techniques, relying on cross-fertlization with artificial intelligence. In addition, our approach is characterised by being free of any parameters including the prior number of the expected communities and independent of any additional measures to decide the community structure.

III. Preliminaries

A. Propositional Logic and SAT Problem

Let $\mathcal{L}$ be a propositional language defined inductively from a finite set $\mathcal{PS}$ of propositional symbols, the boolean constants $\top$ (*true* or 1) and $\bot$ (*false* or 0) and the standard logical connectives $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ in the usual way. We use the letters x, y, z, etc. to range over the elements of $\mathcal{PS}$. Formulas of $\mathcal{L}$ are denoted by A, B, C, etc. A *literal* is a propositional variable (x) of $\mathcal{PS}$ or the negation of a variable ($\neg x$). The two literals x and $\neg x$ are called complementary. A *clause* is a (finite) disjunction of literals, i.e., $a_1 \vee \ldots \vee a_n$. For every propositional formula $\mathcal{A}$ from $\mathcal{L}$, $\mathcal{P}(\mathcal{A})$ denotes the symbols of $\mathcal{PS}$ occurring in $\mathcal{A}$. A *Boolean interpretation* $\mathcal{I}$ of a formula $\mathcal{A}$ is a truth assignement of $\mathcal{PS}$, that is, a total function from $\mathcal{P}(\mathcal{A})$ to $\{0, 1\}$. A *model* of a formula $\mathcal{A}$ is a Boolean interpretation $\mathcal{I}$ that satisfies $\mathcal{A}$, i.e. $\mathcal{I}(\mathcal{A}) = 1$. A formula $\mathcal{A}$ is satisfiable if there exists a model of $\mathcal{A}$. We denote by $\mathcal{M}(\mathcal{A})$ is the set of all models of $\mathcal{A}$.

As usual, every finite set of formulas is considered as the conjunctive formula whose conjuncts are the elements of the set. A formula in *conjunctive normal form* (CNF) is a (finite) conjunction of clauses. The propositional satisfiability (SAT) problem consists in deciding whether a given CNF formula admits a model or not. This well-known NP-Complete problem has seen spectacular progress these recent years.

SAT has seen many successful applications in various fields such as electronic design automation, debugging of hardware designs, artificial intelligence, and data mining. Several SAT extensions have been proposed to deal with optimisation problems. For example, the Max-SAT Problem seeks the maximum number of clauses that can be satisfied. In this paper, we consider one of these optimisation variants referred to as Partial Max-SAT problem. Partial Max-SAT sits between SAT and Max-SAT problems. While SAT requires all clauses to be satisfied, Partial Max-SAT relaxes this requirement by considering two kind of clauses, hard and soft. Partial MaxSAT is the problem of finding an optimal assignment to the variables that satisfies all the hard clauses, while satisfying the maximum number of soft clauses.

B. Overlapping Community Detection

In this subsection, we discuss the classic problem of detecting overlapping community structure in networks.

A network is a graph $\mathcal{N} = (V, E)$ where V is a set of nodes and $E \subseteq V \times V$ is a set of edges. We denote by n (respectively m) the number of nodes (respectively edges) in $\mathcal{N}$. For a node $u \in V$, we denote by N_u the set of neighbors of u, i.e., $N_u = \{v \in V : (u, v) \in E\}$. A triangle in $\mathcal{N}$ is a cycle of length 3. In this paper, we focus on undirected, unattributed graphs. In graph theory, a *community* (or cluster) is described as a set of nodes densely connected internally. More formally,

Definition 1: (**Community Partition**) Let $\mathcal{N} = (V, E)$ be an undirected graph. A community is a set of closely linked nodes in $\mathcal{N}$, and the community detection means to determine all the communities.

In real-world networks, nodes are organized into densely linked sets of nodes that are commonly referred to as *network communities*, clusters or modules. Notice that communities in networks often overlap as nodes can belong to multiple communities at once. Network *overlapping community detection* problem consists in dividing a network of interest into (overlapping) communities for intelligent analysis. It has recently attracted significant attention in diverse application domains. Identifying the community structure is crucial for understanding structural properties of the real-world networks. Various methods have been proposed to identify the community structure of complex networks (see [4], [23] for an overview).

Quality Metrics:

Several measures have been proposed for quantifying the quality of communities in networks (see [24] for a comparative study of quality measures). In this paper, we adopt three popular metrics to assess the performance of our method:

Modularity. The most widely used metric for measuring the quality of network's partition into communities (without a ground-truth) is Newman's *modularity* function [25]. Modularity quantifies the community strength by comparing the fraction of edges within the community with such fraction when random connections between the nodes are made. Networks with high modularity have dense connections between the nodes within communities but sparse connections between nodes in different communities. We use the following equation of modularity, an extension of Newman's modularity function designed to support overlapping communities proposed in [15]. For the given community partition of a network $\mathcal{N} = (V, E)$ with m edges, an extended modularity EQ is given by:

$$EQ = \frac{1}{2m} \sum_{C \in C_{\mathcal{N}}} \sum_{u,v \in C} \frac{1}{O_u O_v} \left[A_{uv} - \frac{d_u d_v}{2m} \right] \tag{1}$$

with $C_{\mathcal{N}}$ the set of communities in $\mathcal{N}$; O_u the number of communities to which the node u belongs and A_{uv} is the element of the adjacency matrix representing the network.

F1 score. Let $\mathcal{N} = (V, E)$ be a network, and $\hat{C}$ (respectively C^*) the set of (respectively ground truth) communities associated to $\mathcal{N}$. The average F1 score measure aims to quantify the level of correspondence between C^* and $\hat{C}$. More precisely, we need to determine which $C_i \in C^*$ corresponds to which $\hat{C}_i \in \hat{C}$. The F1 score is defined as the average of F1 score of the best matching ground-truth community to each detected community, and the F1 score of the best matching detected community to each ground-truth community [20]. More formally, this function is defined as follows:

$$\frac{1}{2} \left(\frac{1}{|C^*|} \sum_{C_i \in C^*} F_1(C_i, \hat{C}_{g(i)}) + \frac{1}{|\hat{C}|} \sum_{\hat{C}_i \in \hat{C}} F_1(C_{g'(i)}, \hat{C}_i) \right) \tag{2}$$

where the best matching g and g' is defined as follows: $g(i) = \arg\max_j F_1(C_i, \hat{C}_j)$, $g'(i) = \arg\max_j F_1(C_j, \hat{C}_i)$, and $F_1(C_i, \hat{C}_j)$ is the harmonic mean of Precision and Recall.

Normalized Mutual Information. This metric adopts the criterion used in information theory to compare the detected communities and the ground-truth communities. Normalized Mutual Information has been proposed as a performance metric for community detection (see [12] for details). It provides a real number between zero and one that gives the similarity between two sets of sets of objects. The Normalized Mutual Information is written as:

$$\frac{H(X) + H(Y) - H(X, Y)}{(H(X) + H(Y))/2} \tag{3}$$

where $H(X)(H(Y))$ is the entropy of the random variable $X(Y)$ associated to the partition $C'(C'')$, whereas $H(X, Y)$ is the joint entropy. This variable is equal 1 only when the two partitions C' and C'' are exactly coincident.

IV. Triangle-Driven Community Detection Using Satisfiability Problem

A. Cohesive Subgraphs

A cohesive subgraph is a pivotal vehicle for the analysis of massive graphs [26]. It has been used for finding communities and spam link farms in web graphs, graph visualization, real-time story identification, DNA motif detection in biological networks, finding correlated genes, epilepsy prediction, to name a few. The problem of mining cohesive subgraphs is one of the typical graph mining tasks that has attracted a lot of attention. The most basic, trivial subgraph, is the *triangle*. Many social networks are abundant in triangles, since typically friends of friends tend to become friends themselves [27], [28]. This phenomenon is observed in other types of networks as well (biological, online networks etc.) and is one of the main reasons which gave rise to the definitions of the transitivity ratio and the clustering coefficients of a graph in complex network analysis.

Inspired by these observations about triangle, we design a novel framework which we call Cohesive Subgraph using SAT problem. The main idea is to exploit triangle structure to detect overlapping communities. The idea of using triangle structure to obtain communities is not new. For instance, [29] recently designed an algorithm based on triangle structure to discover disjoints communities. In this paper, however, we design a highly expressive framework for overlapping community detection in complex networks.

The first step of our work is to propose a novel theoretical concept of dense subgraph structure based on triangles which we called *Cohesive Subgraph* where all triangles in the subgraph are mutually neighborhood or neighborhood by transitivity (see Fig.1 for a visualized toy example). The intuition behind our idea is the following: the larger the number of triangles of a given node closes with its neighbors, the higher the probability that there is a lot of connection between them and therefore they form a community structure. Indeed, our assumption relies on the idea that two neighborhood nodes are more probable to belong to the same community, if they belongs mutually to a large number of triangles so that the mutual friendships between pairs of connected nodes in the same community is maximized.

Moreover, the set of communities can overlap, for instance, we can find nodes in several communities. We show later how to extract nodes of each community. More formally, we will start by the core idea of our contribution.

Let us start by some technical definitions. The reason for which we give these notions in this section is to show that using overlapping triangles can help to find dense structures in large graphs.

*Definition 2 (**Neighborhood of a triangle**):* Let $\mathcal{N} = (V, E)$ be an undirected network. We say that two triangles t_1 and t_2 in $\mathcal{N}$ are neighbors if t_1 and t_2 share at least one node. For a triangle t, the set of its neighboring triangles is denoted by $\Gamma(t)$.

Example 1: Let us consider the network $\mathcal{N} = (V, E)$ of Fig.1. For the triangle $t = \{1, 7, 8\}$, we have $\Gamma(t) = \{\{1, 8, 11\}, \{8, 9, 11\}, \{8, 9, 10\}, \{1, 2, 6\}, \{1, 2, 3\}, \{1, 2, 5\}, \{1, 3, 5\}, \{1, 3, 6\}, \{1, 5, 6\}\}$.

Now, we are looking for neighbors of a given node forming at least one triangle with that node. For that, let us first define the neighborhood triangles of a given node as follows.

*Definition 3 (**Neighboring triangles of a node**):* Let $\mathcal{N} = (V, E)$ be an undirected network and u a node s.t. $u \in V$. The neighboring triangles of u, denoted by $\mathcal{T}(u)$, are the triangles in the neighborhood of u.

We denote by $\Gamma(u)$ the set of nodes formed by neighboring triangles of u.

That is, we look for neighbors of the node u which are connected to each other *via* triangles.

Proposition 1: Let $\mathcal{N} = (V, E)$ be an undirected graph s.t. $u \in V$. Then, $\forall t_1, t_2 \in \mathcal{T}(u)$, t_1 and t_2 are mutually neighbors.

Example 2: Let us consider again the network $\mathcal{N} = (V, E)$ depicted in Fig. 1. Then, the neighborhoods of the node $u = 8$ are the following nodes: $\Gamma(u) = \{1, 7, 9, 10, 11\}$.

Now, on the basis of the definitions of neighborhood of a triangle and neighboring triangles of a node, we define a simpler topological structure and one that is more tractable and can be used as a proxy for extracting overlapping communities as follows.

*Definition 4 (**Cohesive Subgraph**):* Let $\mathcal{N} = (V, E)$ be an undirected network. A cohesive subgraph $\mathcal{N}' = (V', E')$ is a subgraph of $\mathcal{N}$ such that $\forall u \in V'$, $\mathcal{T}(u) \neq \emptyset$ and for all two triangles t_1 and t_2 in $\mathcal{N}'$, either t_1 and t_2 are neighbors (w.r.t Definition 2), or t_1 and t_2 are neighbors by transitivity [1].

That is, a cohesive subgraph is a graph containing only triangles mutually neighborhood or mutually neighborhood by transitivity; and each node is contained within at least one triangle in the graph. Notice that a cohesive subgraph aims to capture the cohesion in social interactions in networks in the light of triangles, since triangle connectivity is strictly stronger than connectivity. Notice that in [30], Huang et al. introduced a structure called k-truss community based on adjacency between triangles. Such structure was defined as the maximal k-truss subgraph [2] with an additional constraint on edge connectivity, i.e., any two edges in a community either belong to the same triangle, or are reachable from each other through a series of adjacent triangles.

So, our principle of community discovery is to start with a seed of well-connected nodes, here one unique initial node, and expand the reachability of this node to include other nodes, nodes that are connected to the seed node already in the community. More precisely, given one node u we aim to find the cohesive subgraph containing u, in which each node is triangle connected with other nodes.

An important thing to note is that, the cohesive subgraph detection task can be solved by modeling it as a propositional satisfiability problem. This allows us to benefit from the recent advances in SAT and its optimisation variants. Thus the user specifies what the problem with the different constraints required and a general purpose solver determines how to solve the problem. This strategy is inspired by a recent work on finding the best k-linked centred communities in a network using SAT problem [31]. In the following subsection, we introduce our SAT-based encoding, which enables to discover the cohesive subgraphs as the set of overlapping communities for a given large graph.

B. Discovering Cohesive Subgraph Community via SAT

In this subsection, we study how to process overlapping communities on large networks. For this, we provide an encoding of the problem of discovering cohesive subgraphs using the propositional satisfiability problem. As mentioned previously, we first design a simple node which is then considered as a seed node in each cluster. Then, we find the cohesive subgraph around this seed node. Let us note that by translating the problem of overlapping community detection to an equivalent SAT problem, we can directly benefit from the recent tremendous advances in SAT, and the constant stream of innovations in this extremely active research field.

In order to discover such cohesive subgraphs, we have to use propositional variables that allow us to capture for each

[1] Two triangles t_1 and t_2 are neighbors by transitivity iff there exists a triangle t_3 s.t. t_1 and t_3 are neighbors and t_2 and t_3 are neighbors.

[2] A k-truss is the largest connected subgraph in which every edge is a part of (reinforced by) at least $(k - 2)$ triangles within the subgraph.

node the set of its neighborhoods, its neighboring triangles, and so on. More precisely, for our SAT encoding we associate each node u with a propositional variable denoted x_u where $x_u \in \{0, 1\}$. The key idea is that the variables assigned to 1 represent the seed nodes, i.e., $S = \{u \in V \mid \mathcal{I}(x_u) = 1\}$. We now describe our SAT-based encoding using such propositional variables.

Our SAT-based encoding consists of the following set of constraints. Given a network $\mathcal{N} = (V, E)$, the first propositional formula expresses the fact that if a node $u \in V$ is a seed node ($\mathcal{I}(x_u) = 1$), then all the neighborhood of u can not be a seed node.

$$\bigwedge_{u \in V} (x_u \rightarrow \bigwedge_{v \in V | v \in \Gamma(u)} \neg x_v) \tag{4}$$

It is worth noticing that the constraint (4) can be expressed by a set of binary clauses [3]:

$$\bigwedge_{u \in V} \bigwedge_{v \in V | v \in \Gamma(u)} (\neg x_u \vee \neg x_v) \tag{5}$$

The second constraint allows us to force the selection of a seed node for each cohesive subgraph. To achieve this, we use the following formula:

$$\bigwedge_{u \in V} \bigvee_{v \in V | v \in \Gamma(u)} x_v \tag{6}$$

Obviously, the formula $(4) \wedge (6)$ may have many candidate solutions (i.e. models). However, choosing an arbitrary model do not always guarantee a best partition of a network into communities. To alleviate this problem, we will consider an objective function to be optimized (by minimizing) on the space of solutions. Let us consider example in Fig.1 . Minimizing the objective function leads to two communities found inside the network and all triangles in each community are mutually neighborhood or neighborhood by transitivity. Then, our cohesive subgraph discovery optimization problem can be formulated as follows:

$$\min \sum_{u \in V} x_u \qquad \text{subject to } (4) \wedge (6) \tag{7}$$

After finding the seed nodes, we still have to determine whether a node u belongs to the cohesive subgraphs or not depending on its connectivity to that seed nodes. To achieve this, we use the following formula that assigns nodes of the network to communities where they belong to, i.e., nodes which are connected to each other via triangles.

$$\bigwedge_{u \in S} \bigvee_{v \in V | v \in \Gamma(u)} x_v \tag{8}$$

where S denote the set of seed nodes in $\mathcal{G}$.

Proposition 2: If the constraints $(4) \wedge (6) \wedge (8)$ are satisfied, then for all $u \notin S$ there exists $v \in S$ s.t. $u \in \Gamma(v)$.

Proposition 2 ensures that if $(4) \wedge (6) \wedge (8)$ admits a model $\mathcal{I}$, then the nodes corresponding to the variables assigned to 1 ($\{u \in V \mid \mathcal{I}(x_u) = 1\}$) are the seed nodes and the network can be partitioned into $|S|$ cohesive subgraphs. Also, Proposition 2 shows that if a node u is not a seed node then there is always a seed node that covers this node. The communities can then be constructed by finding the nodes neighborhood to each seed node. Obviously, the formula $(4) \wedge (6) \wedge (8)$ may admits many candidate solutions (i.e. models).

Example 3: Let us consider again the undirected network $\mathcal{N} = (V, E)$ depicted in Fig.1. Our SAT-based encoding can lead to the following solution: $\mathcal{I} = \{\neg x_1, \neg x_2, x_3, \neg x_4, \neg x_5, \neg x_6, \neg x_7, x_8, \neg x_9, \neg x_{10}, \neg x_{11}\}$. For that solution, $\mathcal{S} = \{3, 8\}$ are the seed nodes of the cohesive subgraphs. After finding the seed nodes, we still have to determine whether a node u belongs to the cohesive subgraph or not depending on its seed node. So for that solution, $\mathcal{N}$ can be partitioned into the two communities $C_1 = \{1, 2, 3, 4, 5, 6\}$ and $C_2 = \{1, 7, 8, 9, 10, 11\}$.

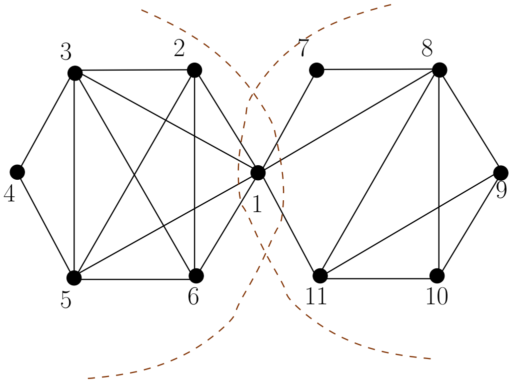

Fig. 1: A simple undirected network

C. Algorithms

This section is devoted to our SAT based overlapping communities framework which is summarized by Algorithm 1 and Algorithm 2. More precisely, the first phase of our approach is to detect the neighborhood triangles of each node from the original network. To do so, Algorithm 1 iterates through every edge (u, v) in the network and if there exits nodes that are linked to the two endpoints for that edge, then the two nodes u and v participate in some common triangles. So, as a result, u will be part of the neighboring triangles of v and v itself becomes part of the neighboring triangles of u.

Once we have the set of neighboring triangles of each node of $\mathcal{N}$, we construct the set of overlapping communities from $\mathcal{N}$ as presented by Algorithm 2. To discover communities, Algorithm 2 is based on propositional satisfiability. It takes as input the network $\mathcal{N}$ and the set of neighboring triangles of each node computed by Algorithm 1, and returns the set of overlapping communities. Algorithm 2 proceeds as follows: First, we generate the corresponding optimization problem that can be represented as a Partial MaxSAT problem (line 1). Then, a state-of-the-art Weighted Partial MaxSAT solver

[3] A binary clause is a clause formed with at most two literals.

Algorithm 1: Neighboring Triangles Computation based on Edge-Iterator

```
Input: A network $\mathcal{N} = (V, E)$
Output: The set of neighboring triangles of $u$ and $v$
for $(u, v) \in E$ do
    if $N_u \cap N_v \neq \emptyset$ then
        $\Gamma(u) \leftarrow \Gamma(u) \cup \{v\}$;
        $\Gamma(v) \leftarrow \Gamma(v) \cup \{u\}$;
    end
end
return $\Gamma(u), \Gamma(v)$
```

WPM3 is used to get an optimal solution (i.e. model) $\mathcal{I}$ (line 2). Next, the seed nodes are determined from the obtained model (lines 4-7). Using such seed nodes, the next step is to build communities by finding the nodes that are neighbors (according to Definition 3) of each seed node (lines 10-12).

Algorithm 2: CSSAT (Cohesive Subgraph Discovering using SAT)

```
   Input: A network $\mathcal{N} = (V, E)$, $\Gamma(u)\ \forall\ u \in V$
   Output: A set of overlapping communities $C$
1  $\Phi = encodeToPartialMaxSAT(\mathcal{N})$;
2  $\mathcal{I} = solve(\Phi)$ ;
3  $S \leftarrow \emptyset$; $C \leftarrow \emptyset$;
4  for $u_x \in \mathcal{I}$ do
5      if $\mathcal{I}(u_x) == 1$ then
6          $C_u \leftarrow \{u\}$;
7          $S \leftarrow S \cup C_u$
8      end
9  end
10 for $C_u \in S$ do
11     for $w \in \Gamma(u)$ do
12         $C_u \leftarrow C_u \cup \{w\}$
13     end
14     $C \leftarrow C \cup C_u$
15 end
16 return $C$
```

V. Performance Evaluation

A. Experiment Settings

In this section, we present an experimental evaluation of our proposed approach. It was conducted on nineteen networks that cover a variety of application areas and are briefly described in columns 1 and 2 of Table I, II and III. All networks in Table I and Table II have ground-truth communities. We have also chosen some large networks (Facebook, DBLP, Amazon, Youtube, Google, Stanford.edu, Notre Dame, Arxiv-Condensed-Matter, High Energy Physics, and Arxiv Astro Physics taken from SNAP [32]) to show the scalability of our model.

We evaluate the performance of our approach by comparing it with the following most prominent state-of-the-art community detection algorithms: (i) *Scalable Community Detection* (SCD) [29], (ii) *Clique Percolation Method* (CPM) [33], (iii) *Cluster Affiliation Model for Big Networks* (BIGCLAM) [20], and (iv) *Detecting Highly Overlapping Communities with Model-Based Overlapping Seed Expansion* (MOSES) [34]. For the CPM algorithm, we use the cliques of size equal to 4.

Our proposed method, referred to as CSSAT, was written in Python. Given an input network as a set of edges, our algorithm starts by generating the corresponding optimization problem represented as a Partial MaxSAT problem. To solve this problem, we consider the state-of-the-art Weighted Partial MaxSAT solver WPM3 (best solver at the last MaxSAT competition [4]) [35]. As finding the optimal solution is NP-hard, in our experiment, we consider the first solution (not necessarily optimal) returned by the solver WPM3. For our experimental study, all algorithms have been run on a PC with an Intel Core 2 Duo (2 GHz) processor and 2 GB memory. We imposed 1 hour time limit for all the methods. Last, we use the symbol (-) in Tables I, II and III when the corresponding method is not able to scale on the considered network under the time limit.

B. Comparison with Baseline Algorithms

Results on ground-truth communities. After finding communities in a given network, we gauge the performance of each community that an algorithm has discovered and check whether a ground-truth community has been successfully identified. Table I and Table II summarize the evaluation results, with F1 and NMI scores of all algorithms on each network. Experiments show that our method outperforms every baseline, in most cases, by an interesting margin as shown by the average F1 Scores and the average NMI Scores reported in the last line of Table I and Table II. Interestingly, it can be seen that CSSAT produces more accurate average w.r.t. the ground-truth setting than all the other baseline algorithms. In terms of average F1 scores, CSSAT outperforms SCD by $40,90\%$, BIGCLAM by $85,22\%$, MOSES by 24% and CPM by 51.02%. In terms of NMI scores, we can see that CSSAT achieves the best performance in 7 networks among 10 and with a higher margin, globally, for each network. More precisely, notice that CSSAT outperforms CPM by $95,97\%$, BIGCLAM by $101,03\%$, SCD by $41,81\%$ and MOSES by $54,76\%$.

As a summary, the experimental results confirm that CSSAT method achieves the overall best performance in terms of the accuracy of the detected overlapping communities.

Results on modularity metric. Table III reports the performance comparison between our CSSAT approach and the considered methods in terms of modularity metric. We observe that across all datasets and modularity metric, CSSAT yields the best performance in 6 out of 9 networks. We also observe that CSSAT gives an important improvement against the baselines in five large networks High Energy Physics, Facebook, Notre Dame, Stanford.edu and Google.

In terms of average performance, CSSAT outperforms CPM by $57,89\%$, BIGCLAM by $24,62\%$, MOSES by $54,76\%$ and SCD by $5,26\%$. On the Arxiv General Relativity, Energy Physics Theory and Arxiv Astro Physics datasets, our method remain relatively competitive with the best baseline. A possible explanation

[4] http://maxsat.ia.udl.cat/introduction/

TABLE I: F1 Score results on ground-truth communities

Network	nodes/edges	SCD	MOSES	CPM	BIGCLAM	CSSAT
KARATE [36]	34/78	0.572	0.528	0.439	0.369	**0.764**
DOLPHIN [37]	62/159	0.308	0.331	0.325	0.206	**0.425**
RISK MAP [38]	42/83	0.680	0.067	0.120	0.594	**0.841**
PILGRIM [39]	34/128	0.360	0.780	0.667	0.537	**0.839**
BOOK [40]	105/441	0.320	0.405	**0.485**	0.201	0.394
RAILWAY [41]	301/1 224	0.326	0.419	0.344	0.390	**0.560**
FOOTBALL [40]	115/615	0.695	**0.854**	0.365	0.611	0.432
DBLP [32]	317 080/1 049 866	0.303	0.363	**0.413**	0.091	**0.413**
AMAZON [32]	334 863/925 872	0.383	0.472	0.402	0.153	**0.482**
YOUTUBE [32]	1 134 890/2 987 624	0.233	0.119	–	0.018	**0.259**
Average	-	0.418	0.475	0.390	0.318	**0.589**

TABLE II: NMI Score results on ground-truth communities

Network	nodes/edges	SCD	MOSES	CPM	BIGCLAM	CSSAT
KARATE [36]	34/78	0.363	0.260	0.221	0.182	**0.516**
DOLPHIN [37]	62/159	0.144	0.163	0.175	0.081	**0.201**
RISK MAP [38]	42/83	0.558	0.055	0.030	0.427	**0.724**
PILGRIM [39]	34/128	0.468	0.575	0.546	0.376	**0.641**
BOOK [40]	105/441	0.199	0.155	**0.273**	0.105	0.153
RAILWAY [41]	301/1 224	0.093	0.157	0.132	0.138	**0.391**
FOOTBALL [40]	115/615	0.420	**0.762**	0.223	0.436	0.221
DBLP [32]	317 080/1 049 866	0.145	0.149	**0.198**	0.031	**0.198**
AMAZON [32]	334 863/925 872	0.157	**0.221**	0.201	0.032	0.210
YOUTUBE [32]	1 134 890/2 987 624	0.049	0.012	–	0.0006	**0.050**
Average	-	0.275	0.252	0.199	0.194	**0,390**

TABLE III: Modularity Score results

Network	nodes/edges	SCD	MOSES	CPM	BIGCLAM	CSSAT
Arxiv-Condensed-Matter [32]	23 133/93 439	0.339	0.371	0.360	0.343	**0.378**
Arxiv General Relativity [32]	5 242/14 484	0.589	0.505	0.459	**0.608**	0.522
Energy Physics Theory [32]	9 877/25 973	**0.423**	0.374	0.291	0.355	0.382
Arxiv Astro Physics [32]	18 772/198 050	**0.408**	0.232	0.153	0.233	0.226
High Energy Physics [32]	12 008/118 489	0.231	0.279	0.262	0.294	**0.300**
Facebook [32]	4 039/88 234	0.433	0.325	–	0.391	**0.486**
Notre Dame [32]	325 729/1 090 108	0.401	0.421	0.439	0.381	**0.497**
Stanford.edu [32]	281 903/1 992 636	0.293	0.445	–	0.432	**0.502**
Google [32]	875 713/4 322 051	0.476	0.329	–	0.001	**0.492**
Average	-	0, 399	0.364	0.266	0.337	**0.420**

for this phenomenon is that the WPM3 solver don't return the optimal solution for these datasets.

Evaluating scalability. Finally, we evaluate the scalability of our community detection method by measuring the CPU time (see Figure 2). Notice that our algorithm needs few seconds to generate all communities for small networks such as Karate, Dolphin, etc. From the results in Figure 2, it can be seen that our algorithm takes few seconds (less than 200 seconds) to generate all communities for all networks with a number of edges less than 500 000. For all large networks having one million of edges and more, our algorithm makes less than 1200 seconds to generate all communities.

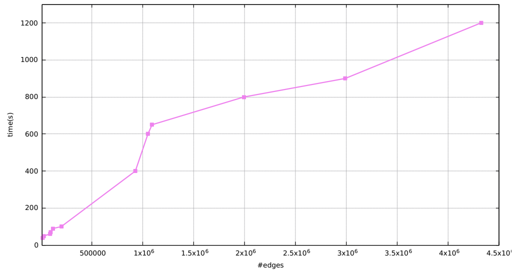

Fig. 2: Evaluating scalability for large graphs

VI. Conclusion

In this paper, we proposed a new framework for detecting overlapping communities in real-world networks. Our method is based on discovering clusters in networks based on triangles to capture the cohesion in social interactions. Then, we demonstrated that our problem can be expressed as a Partial Max-SAT optimization problem. Our approach is characterised by being free of any parameters including the prior number of the expected communities and independent of any additional measures to decide the community structure. Experimental results showed that our approach outperforms the state-of-the-art methods in accurately discovering network communities. These performances are obtained while looking for the first non necessarily optimal solution of the underlying optimisation problem.

As a future work, we intend to develop a parallel version to even improve the performance of our optimisation based approach. Finally, we plan to extend our proposed framework to deal with the problem of community search.

References

[1] T. Guns, S. Nijssen, and L. D. Raedt, "Itemset mining: A constraint programming perspective," *Artif. Intell.*, vol. 175, no. 12-13, pp. 1951–1983, 2011.

[2] S. Gilpin and I. N. Davidson, "Incorporating SAT solvers into hierarchical clustering algorithms: an efficient and flexible approach," in *KDD*, 2011, pp. 1136–1144.

[3] I. Davidson, S. S. Ravi, and L. Shamis, "A sat-based framework for efficient constrained clustering," in *SDM*, 2010, pp. 94–105.

[4] J. Leskovec, K. J. Lang, and M. W. Mahoney, "Empirical comparison of algorithms for network community detection," in *International Conference on World Wide Web, WWW*, 2010, pp. 631–640.

[5] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, pp. 75–174, 2010.

[6] G. Palla, I. Dernyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *nature*, vol. 435, pp. 814–818, 2005.

[7] I. J. Farkas, D. Abel, G. Palla, and T. Vicsek, "Weighted network modules," *New Journal of Physics*, vol. 9, 2007.

[8] J. M. Kumpula, M. Kivela, K. Kaski, and J. Saramaki, "Sequential algorithm for fast clique percolation," *Physical Review*, vol. E 78, 2008.

[9] C. Shi, Y. Cai, D. Fu, Y. Dong, and B. Wu, "A link clustering based overlapping community detection algorithm," *Data & Knowledge Engineering*, vol. 87, pp. 394 – 404, 2013.

[10] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, p. 761764, 2010.

[11] T. S. Evans, "Clique graphs and overlapping communities," *Journal of Statistical Mechanics: Theory and Experiment*, 2010.

[12] A. Lancichinetti, S. Fortunato, and J. Kertesz, "Community detection algorithms: A comparative analysis," *New Journal of Physics*, vol. 11, 2009.

[13] D. Jin, B. Yang, C. Baquero, D. Liu, D. He, and J. Liu, "A markov random walk under constraint for discovering overlapping communities in complex networks," *New Journal of Physics*, vol. 5, 2011.

[14] A. Padrol-Sureda, G. Perarnau-Llobet, J. Pfeifle, and V. Muntés-Mulero, "Overlapping community search for social networks," in *International Conference on Data Engineering*, 2010, pp. 992–995.

[15] H. Shen, X. Cheng, K. Cai, and M. Hu, "Detect overlapping and hierarchical community structure in networks," *Physica A*, vol. 388, no. 8, pp. 1706–1712, 2009.

[16] I. M. Kloumann and J. M. Kleinberg, "Community membership identification from small seed sets," in *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, 2014, pp. 1366–1375.

[17] S. Gregory, "Finding overlapping communities using disjoint community detection algorithms," in *International Workshop on Complex Networks*, 2009, pp. 47–61.

[18] J. Xie, B. K. Szymanski, and X. Liu, "SLPA: uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process," in *IEEE International Conference on Data Mining*, 2011, pp. 344–349.

[19] W. Chen, Z. Liu, X. Sun, and Y. Wang, "Community detection in social networks through community formation games," in *International Joint Conference on Artificial Intelligence, Barcelona*, 2011, pp. 2576–2581.

[20] J. Yang and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach," in *ACM International Conference on Web Search and Data Mining*, 2013, pp. 587–596.

[21] H. Zhang, I. King, and M. R. Lyu, "Incorporating implicit link preference into overlapping community detection," in *AAAI Conference on Artificial Intelligence*, 2015, pp. 396–402.

[22] J. Yang, J. J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," *CoRR*, vol. abs/1401.7267, 2014.

[23] S. Fortunato, "Community detection in graphs," *CoRR*, vol. abs/0906.0612, 2009.

[24] J. Leskovec, D. P. Huttenlocher, and J. M. Kleinberg, "Predicting positive and negative links in online social networks," in *International Conference on World Wide Web, WWW*, 2010, pp. 641–650.

[25] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, p. 026113, Feb. 2004.

[26] W. Stanley and K. Faust, *Social network analysis: Methods and applications*. Cambridge university press, 1994.

[27] C. E. Tsourakakis, U. Kang, G. L. Miller, and C. Faloutsos, "DOULION: counting triangles in massive graphs with a coin," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 837–846.

[28] H. Park, F. Silvestri, U. Kang, and R. Pagh, "Mapreduce triangle enumeration with guarantees," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 2014, pp. 1739–1748.

[29] A. Prat-Pérez, D. Dominguez-Sal, and J. Larriba-Pey, "High quality, scalable and parallel community detection for large real graphs," in *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, 2014, pp. 225–236.

[30] X. Huang, H. Cheng, L. Qin, W. Tian, and J. X. Yu, "Querying k-truss community in large and dynamic graphs," in *SIGMOD*, 2014, pp. 1311–1322.

[31] S. Jabbour, N. Mhadhbi, B. Raddaoui, and L. Sais, "A sat-based framework for overlapping community detection in networks," in *Advances in Knowledge Discovery and Data Mining - 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea, May 23-26, 2017, Proceedings, Part II*, 2017, pp. 786–798.

[32] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

[33] B. Adamcsek, G. Palla, I. J. Farkas, I. Derényi, and T. Vicsek, "Cfinder: locating cliques and overlapping modules in biological networks," *Bioinformatics*, vol. 22, no. 8, pp. 1021–1023, 2006.

[34] N. H. Aaron McDaid, "Detecting highly overlapping communities with model-based overlapping seed expansion," Mar. 2010, overlapping, scalable, community finding algorithm.

[35] C. Ansótegui, F. Didier, and J. Gabàs, "Exploiting the structure of unsatisfiable cores in maxsat," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015, pp. 283–289.

[36] Z. W.W., "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, pp. 452–473, 1977.

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

[38] J. Cheng, M. Leng, L. Li, H. Zhou, and X. Chen, "Active semi-supervised community detection based on must-link and cannot-link constraints," *PLoS ONE*, vol. 9, no. 10, pp. 1–18, 10 2014.

[39] Brian, B. Dickinson, W. Valyou, and Hu, "A genetic algorithm for identifying overlapping communities in social networks using an optimized search space," *Social Networking*, vol. Vol.02No.04, pp. 1–9, 2013.

[40] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *PROC.NATL.ACAD.SCI.USA*, vol. 99, p. 7821, 2002.

[41] T. Chakraborty, S. Srinivasan, N. Ganguly, A. Mukherjee, and S. Bhowmick, "On the permanence of vertices in network communities," in *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, 2014, pp. 1396–1405.