A data reduction approach using hypergraphs to visualize communities and brokers in social networks

Luís Cavique, MAS-BioISI, FCUL and Universidade Aberta, Portugal
Nuno C. Marques, CITI and FCT, Universidade Nova Lisboa, Portugal
António Gonçalves, INESC-ID and EST-IP Setúbal, Portugal

Abstract

The comprehension of social network phenomena is closely related to data visualization. However, even with only hundreds of nodes, the visualization of dense networks is usually difficult. The strategy adopted in this work is data reduction using communities. Community detection in social network analysis is a very important issue and in particular detection of community overlapping. In this approach, the information extracted from social networks transcends cohesive groups, enabling the discovery of brokers that interact among communities. In order to find admissible solutions in hard problems, relaxed approaches are used. Quasi-cliques are generated, and partition is found using a partial set covering heuristic. The proposed method allows the identification of communities and actors that link two or more groups. In the visualization process, the user can choose different dimension reduction approaches for the condensed graph. For each condensed structure a hypergraph can be drawn, identifying communities and brokers.

Keywords

graph mining, data reduction, community detection, brokerage, hypergraphs

1 Introduction

Social networks are usually represented with graph theory, where the set of vertices corresponds to ‘actors’ (i.e. people, companies or social actors) and the set of edges corresponds to ‘ties’ (i.e. relationships, associations or links).

Social Network Analysis, which studies the interaction between individuals and organizations, is a very relevant technique that has emerged in modern sociology. See Scott (1991) and Wasserman, Faust (1995) for the theoretical basis and key techniques in social networks. In classic social network analysis two types of metrics can de differentiated – the structural metrics and the community metrics (Scott 1991; Wasserman, Faust 1995). The structural metrics include the centrality measures and deal with the whole network, whereas the community metrics comprehend the network partition into communities and affiliations of overlapping subgroups.

More recent works in network science (Newman 2010; Barabási 2016) include new metrics like degree distribution and degree correlation in addition to dynamical processes in networks.

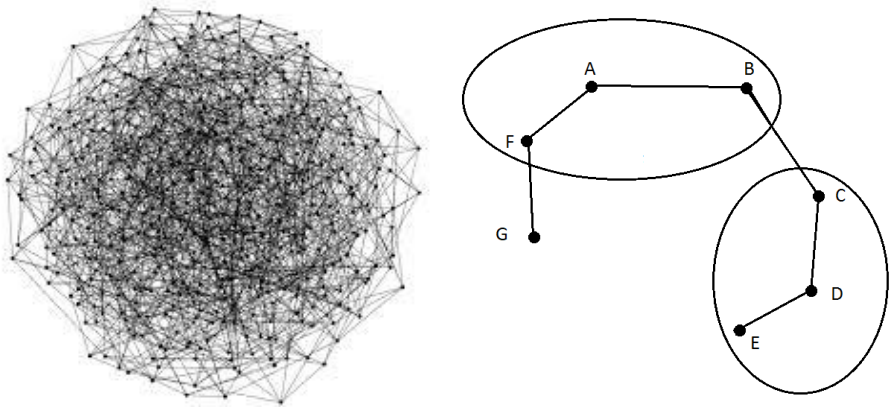
The visualization of a small number of vertices can easily be mapped. However, the visualization of the whole graph becomes incomprehensible when the number of vertices and edges increases. The growth of the available data becomes incompatible with the full mapping of the network, resulting in a pressing need for new metrics and visualization tools to explore dense social network structures.

Objectives

Since competitive advantages come from information access and control, brokers that span structural holes create new opportunities for the network and compensations for themselves (Burt 1992, Burt 2005). Brokerage represents not only a source of competitive advantage for individuals but also a relevant concept in understanding the knowledge-based organizations. Therefore, we believe these characteristics provide the justification for the study of the brokerage subject.

Community detection is one of the most important features in social networks and many algorithms have been proposed to find network partitions (Girvan and Newman 2002; Fortunato 2010; Cruz et al. 2014; Liu et al. 2018). However, the specific discovery of brokers between partitions remains scarce.

In this work we aim to find not only the communities but also the brokers. We also intend to create a model that allows the visualization of communities and brokers in dense networks. In a dense network, Figure 1.a, it is very difficult to identify any element. However, with a condensed structure, Figure 1.b, communities (A to G) and brokers (links e.g. AB, CD and hyperlinks e.g. ABF, CDE) can clearly be visualized in a hypergraph.



(a) 'fur ball' network (b) proposed method

Figure 1. Given a dense network (a) the proposed method generates a condensed structure with communities and brokers that are visualized in hypergraph (b)

Synonyms

In this study we use the words 'network' and 'graph' as synonyms. The terms 'cluster', 'community', 'dense group', 'connected component', 'clique', 'quasi-clique' or 'k-clique' have a similar meaning, while 'broker' and 'influential node' are also identical. Equally, 'condensed graph', 'condensed structure', 'clustered graph' and 'network partition' have the same significance.

Contributions
The main contribution of this work is the transformation of clusters into vertices of a weighted hypergraph, where the edges correspond to brokers. This document details and extends previous works (Cavique et al. 2009; Cavique et al. 2012; Cavique et al. 2014). The novelty of this work includes the following contributions:

- Quality: a method to condense networks using relaxed structures such as quasi-cliques and partial covers, which allows finding better results in community detection benchmark datasets.
- Usability: the proposed method uses a parameter k that tunes the level of aggregation of the condensed graph.
- Visualization: an innovative form of visualization of communities and brokers using hypergraphs.

Organization
This document is organized as follows. In Section 2, we present the related work and background information. In Section 3, we present the proposed method which detects the communities and discovers brokers in social networks. In the visualization of the condensed structures a hypergraph is used, and a numeric example is shown. In Section 4, the computational results are presented. Finally, in the last section we draw some conclusions.

2 Related Work and Background Information

In this work data needs to be reduced into a condensed graph in order to identify communities and brokers in a social network. We use graph theory to define condensed graph: the condensed graph G_c of graph G is obtained by contracting all the arcs and nodes into a single node, in every strongly connected component.
In this section, firstly we present the related work in community detection and layout of clustered graphs. Then, we introduce the background information that combines the areas of brokerage in social network analysis, with sub-graph discovery and the minimum set covering optimization problem.

2.1 Community Detection
In this particular area of social networks, authors do not always use the same keywords, so there are several synonyms for community detection. This work is related to community detection/finding/mining/discovery, graph clustering, graph summarization, graph partition and graph compression.

Community or group can be defined as a set of nodes with similarity. A partition is a sub-division of a graph into groups of vertices such that each vertex is assigned to one group.

One of the first studies is given by the Kernighan and Lin (1970) algorithm, which finds a partition of the nodes by dividing the set into two disjoint subsets A and B of

equal size, such that the sum of the weights of the edges between nodes in A and B is minimized.

The Girvan and Newman (2002) method has been applied in recent years to social networks. This method successively deletes edges of high 'betweenness', and then recalculates all 'betweenness', breaking each component into smaller components.
Other studies, based on physics, introduced the concept of clique percolation (Derenyi et al. 2005), where the network is viewed as a union of cliques.

Blondel et al. (2008) transform clusters into vertices of a weighted graph, where the edges correspond to bridges.

A detailed review about community detection in graphs, which includes traditional methods, modularity-based methods, spectral algorithms, methods based on statistical inference and multiresolution method can be found in Fortunato (2010).

The SlashBurn application (Kang, Faloutsos 2011) is designed for large datasets with fast processing times and low disk space requirements.

The BigClam method (Yang, Leskovec 2013) combines overlapping community detection as well as hierarchically nested structures in large networks.

The HyCoM application (Araujo et al. 2014) uses a Minimum Description Length to guide the community discovery process, with no user-defined parameters and an adequate scalability.

Cruz et al. (2014) present an interesting method that combines community detection and visualization of brokers using colors to identify groups.

The recent method CondenNSe (Liu et al. 2018) finds 30-50% more compact networks than baselines, with up to 75-90% fewer structures. In this work of graph summarization, the super-graphs have the same meaning as the condensed graphs.

Most community detection algorithms are not fully adapted to the user needs regarding different levels of node aggregation and their visualization. In our work a parameter k tunes the level of aggregation of the network, allowing different visualizations.

2.2 Visualization of Clustered Graphs Layouts

Clustered graph layout is a sub-area of graph drawing which considers the meta information of a graph and does not alter its structure.

Eades and Feng (1997) present one of the first attempts to visualize clustered graphs in multiple abstraction levels. The algorithm allows a three-dimensional visualization of different layers of data aggregation. The extended work of Eades and Huang (2000) provides an architecture to handle clustered graphs using animation methods.

The work of Li and Takatsuka (2004) follows the same clustered graph definition as Eades and Feng (1997) and proposes a new approach by adding filtering to geometric distortion of the clustered graph.

Bourqui et al. (2007) present an algorithm to lay out clustered weighted graphs using Voronoi diagrams. The method allows the visualization of groups avoiding overlap, leading to drawings that identify groups and adjacency relations.

The recent work of Didimo and Montecchiani (2014) presents an algorithm to compute a two-dimensional layout of hierarchically clustered graphs, running in a parallel environment, with a fast performance.

Most clustered graphs compute recursively in different hierarchical levels the grouping of vertices. However, the interactions among communities are not a priority. In our work the notion of levels persists and the brokerage concept is added.

2.3 Brokerage in Social Network Analysis

The visualization using a graph, called sociogram, was initially presented by Moreno (1934). This scientific area of sociology tries to explain why alliances and conflicts are generated in groups, how the group structure affects the group efficacy, how the leadership emerges and how diffusion of innovation works.

Erdos, who was one of the most prolific mathematicians, wrote over 1500 papers with more than 500 co-authors. Erdos represents the number zero in his social network and the researchers who worked with him are called Erdos number of 1. Erdos' number of 1 co-authors are called Erdos number of 2, and so on, building one of the oldest small world networks known. The work of Erdos and Renyi (1959) presents interesting random graph properties. An interesting example of the "Erdos Number" can be found in Grossman et al. (2007).

A relevant development in the structure of social networks came from an experiment by the American psychologist Stanley Milgram (1967). Milgram's experiment consisted in sending letters from people in Nebraska, in the Midwest, to be received in Boston, on the East Coast, where people were instructed to pass on the letters, by hand, to someone else they knew. The letters that reached the destination were passed on by an average of six people. Milgram concluded the experiment showed that, on average, Americans are no more than six steps away from each other. This experiment led to the six- degree concept of separation and the notion of small world.

In the late 1960s, while working on his Ph.D., Mark Granovetter interviewed people who had recently changed jobs, to find out how they had found their new jobs. Surprisingly, the information about the new jobs had come from distant acquaintances instead of close friends. The concept of strong and weak ties (Granovetter 1973) introduced a novel principle in social networks. Weak ties are valuable because they will more likely be the source of novel information and openness to new opportunities. On the other hand, strong ties intensify group cohesion and maintenance of group identity. This led to the Triadic Closure property, which establishes that if the node has strong ties to two neighbors, these neighbors must have at least a weak tie between them. The property is based on the fact that if two people have a friend in common, it is probable that they will become friends in the future (Easley and Kleinberg 2010).

Sociograms can easily represent a small network, not exceeding dozens of nodes. When the number of nodes increases it is more difficult to draw the network. With the advent of computer programs, several social network analysis packages have become available including NEGOPY (Richards, Rice 1981), (Richards 1995) and UCINET (Borgatti et al. 2002).

The package NEGOPY, short for 'negative entropy', uses an adjacency matrix of the graph and identifies four network structures. The group consists of at least three strongly connected nodes. The liaison is a node which belongs to a group but links two or more groups. A dyad is a pair of linked nodes. Finally, an isolated node is an unconnected node. A practical application using the NEGOPY package and census

data related to the use of contraceptives was carried out by Rogers and Kincaid (1981) in the study of communication networks of Korean women in rural villages.

A new interest was revived with the Watts and Strogatz model, published in the Nature journal (Watts, Strogatz 1998), which studies graphs with small-world properties and power-law degree distribution.

Connectivity of Graphs and Brokerage

One important concept in community structure metrics is the notion of connectivity of the graph. In this sub-section node-connectivity and line-connectivity in graphs is discussed (Harary 1969).

In a connected graph all nodes are reachable. On the contrary, a graph is disconnected if there is no path between any pair of nodes. A graph with only one node is also considered connected. A graph G can lose its connectivity when specific lines or nodes are deleted.

A node *n* is a cut-point (or cut-vertex) if the number of components in G_1 is fewer than the number of components in G_2 that results from deleting node *n*. Analogously, a bridge (or line-cut) is an edge that is critical to the connectedness of the graph. The line *l* is a line-cut if the number of components in G_1 is fewer than the number of components in G_2 that results from deleting line *l*.

Generalizing, a k-line-cut is a set of lines that, if deleted, disconnects the graph. A bridge is a 1-line-cut. Similarly, k-cut-point is a set of nodes that, if removed, makes the graph disconnected.

Following the Triadic Closure property, R.S. Burt (1992) developed a complementary approach coined 'Structural Holes', referring to the absence of links in a connected organization. He also introduced the term brokerage, meaning nodes that connect two dense groups.

The concepts of node-connectivity and line-connectivity in graphs discussed by Harary (1969) are directly related to the liaison by Richards and Rice (1981) and *structural holes* by Burt (1992).

Figure 2 shows two ways of spanning structural holes, using a bridge or a broker. Structural hole, bridge and broker can be defined as follows:

- structural hole refers to the lack of edges between components, or communities;
- bridge is an edge whose removal increases the number of components in the network;
- broker or cut-vertex is a vertex whose deletion increases the number of components in the network.

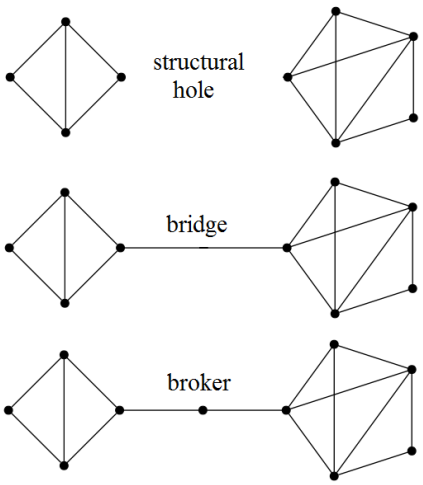


Figure 2. Structural hole, bridge and broker

Influential nodes, like brokers and nodes forming bridges, have a central position in the network and are connected to other communities to maximize their influence (Aggarwal et al. 2012).

Brokers can play many different roles. The differences between coordinator, gatekeeper, consultant, representative and liaison can be found in Long et al. (2013):

- coordinator: a person who links connections within the same group;
- gatekeeper: a person who controls the information that passes into or out of the group;
- consultant: a person who takes a central lead with clusters who are not directly linked;
- representative: a person who links his group to an outside group;
- liaison: a person who links different outside clusters without having prior loyalty to either.

Note that the term 'liaison' given by Long et al. (2013) does not have the exact same meaning given by Richards and Rice (1981).

Brokerage across structural holes provides new opportunities and "good ideas". Structural holes between business unit organizations are common, leading to compensations for the broker. Two types of broker strategies must be referred: the altruistic broker, the *tertius iungens*, which means "the third who joins" the communities and its opposite, the egoistic broker, the *tertius gaudens*, meaning "the third who benefits". For many years, the centrality of the actors in networks has been an important issue in social network analysis.

The central node, or hub, can be found using different measures: degree centrality, "betweenness" centrality, closeness centrality or eigenvector centrality. The brokers have some similarities with hubs, who also score high in terms of centrality. However, the "centrality" of the brokers lies between different communities instead of the hubs that belong to a specific community.

2.4 Cliques and Relaxed Cliques

Graph theory has many applications and has been used in different fields. Berge (1970) established the present notation in graph theory that is used in this section.

In this notation, an undirected graph is represented by G=(V,E), where $E \subseteq [V]^2$ is a pair in which V(G) represents the set of vertices or nodes, and E(G), the set of edges. An edge can also be represented by $\{i, j\} \in E(G)$, where *i* and *j* are the two connected vertices. The number of vertices V(G) can be represented by |V(G)| and the graph called of order *n* if V(G)={1,2,…,n} and so, |V(G)|=n. The number of edges *m* is given by the cardinality of E(G), i.e. |E(G)|. If two vertices are joined by an edge, they are adjacent.

A graph G'=(V',E') is a sub-graph of the graph G=(V,E) if $V' \subseteq V$ and $E' \subseteq E$. Given a subset of V, called S, then G'=G-S denotes the sub-graph G' induced from G by deleting all vertices in S and their incident edges. A sub-graph G_1 is said to be complete, if there is an edge for each pair of vertices.

A clique is a complete sub-graph, meaning that each member has direct ties with every other member or node. A clique is maximal if it is not contained in any other clique. The maximum clique, $\omega(G)$, is the clique with maximum cardinality in graph G. The Maximum Clique is an NP-hard problem. In order to find a lower bound for the maximization problem, the heuristic proposed by Johnson (1974) and the meta-heuristic that uses Tabu Search, developed by Soriano and Gendreau (1996) can be used. Following works can be found in Cavique et al. (2001) and Cavique and Luz (2009).

The clique structure must contain an edge for each pair of vertices. This constrained structure presents many restrictions in real life because it is uncommon in social networks. Therefore, alternative approaches for more relaxed cohesive groups were suggested, such as k-clique, k-clan/k-club and k-plex (Scott 1991). The quasi-clique is also a relaxation of a clique, which ignores the lack of some edges in the complete sub-graph. In this work a combination of k-cliques and quasi-cliques is developed.

Relaxations: k-clique, k-clan/k-club and k-plex

Distance and diameter are useful metrics in social networks. The length of the shortest path between vertices u and v in G is denoted by the distance d(u,v). The diameter of G is given by diam(G)= max d(u, v) for all $u,v \in V$.

Luce (1950) introduced the dense group based on distance called k-clique, where k is the maximum path length between each pair of vertices. A k-clique is a subset of vertices C such that, for every $i, j \in C$, the distance $d(i, j) \leq k$. The 1-clique is identical to a clique because the distance between the vertices is one edge. The 2-clique is the maximal complete sub-graph with a path length of one or two edges. The path distance equal to two can be exemplified by the "friend of a friend" connection in social relationships. In social websites, like LinkedIn, each member can reach his own connections as well as the ones two and three degrees away. The increase of value k corresponds to a gradual relaxation of the criterion of clique membership. See Figure 3.

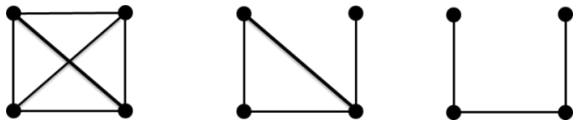


Figure 3. Examples of a 4 node graph of 1-clique, 2-clique and 3-clique

In graph theory, the k^{th} power of graph G returns a new graph G^k where each pair of vertices is adjacent when their distance in G is at most k. To find all the maximal k-cliques in the graph, the k^{th} power of graph G can be used associated with the maximum clique algorithm. The transformation process adds edges to reach length k between every pair of nodes.

A limitation of the k-clique concept is that some vertices may be distant from the group. The distance between two nodes may correspond to a path involving nodes that do not belong to the k-clique. To overcome this handicap the cohesion groups based on diameter called k-club and k-clan were introduced. To find all k-clan, all the k-cliques S^i must be found first, and then the restriction diam(G[S])≤ k applied to remove the undesired k-cliques. In Figure 4, the left graph shows the 2-clique {1,2,3,4,5} is not a 2-clan because d(4,5)=3. The path 4-6-5 is not possible as node 6 does not belong to the sub-graph with the 2-cliques. Another approach to these diameter models is the k-club, which is defined as a subset of vertices S such that diam(G[S])≤ k. In the same graph two 2-cliques can be found: {1,2,3,4,5} and {2,3,4,5,6}, one 2-clan: {2,3,4,5,6} and three 2-clubs: {1,2,3,4}, {1,2,3,5} and {2,3,4,5,6}.

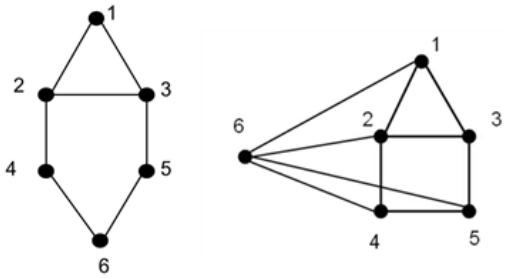


Figure 4. 2-clans, 2-clubs (left) and 3-plex (right).

An alternative way of relaxing a clique is the k-plex concept which considers the vertices' degree. The degree of a vertex of a graph is the number of edges incident to the vertex and is denoted by deg(v). The maximum degree of graph G is the maximum degree of its vertices and is denoted by Δ(G). On the other hand, the minimum degree is the minimum degree of its vertices and is denoted by δ(G). A subset of vertices S is a k-plex, if the minimum degree in the induced sub-graph $\delta(G[S]) \geq |S| - k$. In Figure 4, the right graph shows |S|=6 and the degree of vertices 1, 2, 3, 4 and 5 does not exceed the value 3. Thus, the minimum degree in the induced sub-graph δ(G[S]) is 3. For |S|=6, k=3 is obtained.

Quasi-cliques

In this sub-section general properties of quasi-cliques in an undirected graph G = (V,E) are described. Let S be the set of vertices of the sub-graph induced by S in G. In a complete sub-graph S the number of edges is equal to the combinations (|S|, 2) or |S|.(|S|-1)/2.

Quasi-cliques are almost complete sub-graphs with a specified edge density γ. Quasi-cliques can also be referred as γ-cliques, where the real parameter γ is $0 < \gamma \leq 1$. For γ =1, the 1-clique corresponds to a complete subgraph.

Two definitions can be found in the literature for quasi-cliques. One considers the density of the whole graph (Abello et al. 2002) and the other verifies the minimum degree for each vertex (Pei et al. 2005).

For Abello et al. (2002) the number of edges in γ-clique should be larger than the product of the edge density γ by the number of edges of the complete graph with |S| vertices. In other words, the induced graph S is a quasi-clique, if $|E(S)| \geq \gamma . \binom{|S|}{2}$.

For Pei et al. (2005) for every vertex, in the sub-graph S, the vertex degree should be greater than the product of the edge density λ by $|S| - 1$. In other words, the induced graph S is a quasi-clique, if $min\_deg(S) \geq \lfloor \lambda . (|S| - 1) \rfloor$.

The work of Brutano et al. (2008) reuses the previous definitions and defines two parameters λ and γ, with $0 < \lambda \leq \gamma \leq 1$, where λ reveals a local view and γ a more general one. The induced graph S is a (λ, γ)-quasi-clique, if, and only if, the two following conditions are ensured: $min\_deg(S) \geq \lfloor \lambda . (|S| - 1) \rfloor$ and $|E(S)| \geq \gamma . \binom{|S|}{2}$.

The maximum γ-clique problem is to find a γ-clique with the maximum cardinality in graph G.

2.5 Set Partition and Set Covering Problem

A partition is a sub-division of a graph into groups of vertices such that each vertex is assigned to one group. The mathematical formulation of the Set Partition problem can be stated as follows, where matrix [a(i,j)] stores the information about different communities and for each variable x, a cost can be associated by using a vector $[c_j]$:

minimize $f = \sum c_j . x_j$
subject to $\sum a_{i,j} . x_j = 1$
and $x_j \in \{0,1\}$ $\quad j=1,\ldots,n$

Since the constraint $\sum a_{i,j} . x_j = 1$ is very restricted, some problems are not solvable, such as when a node is shared by two or more communities.

A more relaxed problem that allows a node to share two components is the Minimum Set Covering problem. In the mathematical formulation the sign of the constraint is replaced for equal or greater instead of just equal, allowing the existence of over-covered nodes

The optimization problem that finds the minimum number of columns that covers all the rows is the Minimum Set Covering problem. For each attribute x, a cost can be associated by using a vector $[c_j]$, allowing a cost differentiation among attributes. The matrix and the cost vector are then used in the set covering problem, defined as:

minimize $f=\sum c_j.x_j$
subject to $\sum a_{i,j}.x_j \geq 1$
and $x_j \in \{0,1\}$ $j=1,\ldots,n$

The Minimum Set Covering problem finds the minimum number of columns/clusters that covers all the rows/constraints/nodes. In the objective function for each attribute x_j, a cost can be associated by using a vector $[c_j]$, allowing a cost differentiation among attributes. Using the matrix representation, constraints are supported by matrix [a(i,j)].

We used the Minimum Set Covering problem to find the minimum number of clusters that covers all the nodes in our previous works (Cavique et al. 2009; Cavique et al. 2012; Cavique et al. 2014) where the costs c_j do not vary, i.e. $c_j=1,\forall j$.

Example: Given a graph with 6 nodes and 8 clusters with costs = {1,1,1,1,1,1,1,1}, with the following structure {(1,3), (2,3,4), (2,4,5), (2,3,4,5), (1,2,3,6), (5,6), (2,5,6), (1,4,5)}, where each node should be covered by at least one cluster, find the minimum number of clusters that covers all the nodes. The integer linear programming formulation is as following:

Minimize $z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8$
subject to:

node 1: $x_1 + x_5 + x_8 \geq 1$
node 2: $x_2 + x_3 + x_4 + x_5 + x_7 \geq 1$
node 3: $x_1 + x_2 + x_4 + x_5 \geq 1$
node 4: $x_2 + x_3 + x_4 + x_8 \geq 1$
node 5: $x_3 + x_4 + x_6 + x_7 + x_8 \geq 1$
node 6: $x_5 + x_6 + x_7 \geq 1$

$x_j \in 0,1$, $j = 1,\ldots,8$

The solution returns the global cost z =2 and $x_5=x_8=1$ and $x_j=0$ for the other variables. The minimal cover is {5,8}, corresponding to clusters 5 and 8. Similar optimal solutions are given by {3,5} or {4,5}.

The Minimum Set Covering problem is widely studied in Combinatorial Optimization, with many computational resources which implement quasi-exact algorithms and heuristic approaches.

The minimum set covering heuristic, proposed by Chvatal (1979), repeats the process by choosing the line with fewer elements, followed by choosing the column with the best ratio between column cost and the number of covered lines.

To solve highly constrained covering problems relaxed approaches can be adopted, such as the partial set covering (Bilal et al. 2014), where it is not necessary to cover all the elements.

In Section 3 the concepts of k-quasi-clique and partial set covering will be used in the proposed algorithm.

3 The Proposed Model

Although there are several community detection algorithms, there are few studies about the linkage between them, namely concerning issues related to brokerage. In this section we present a new approach to simplify the visualization of the network based on a hypergraph, associated with parameter *k* that tunes the level of aggregation of the network.

Our proposed model is depicted in Figure 5, where the circles represent the processes and the storage symbols represent the data. Based on the original network the community detection algorithm returns the condensed graph with communities and brokers. The visualization process represents the condensed structure in a hypergraph.

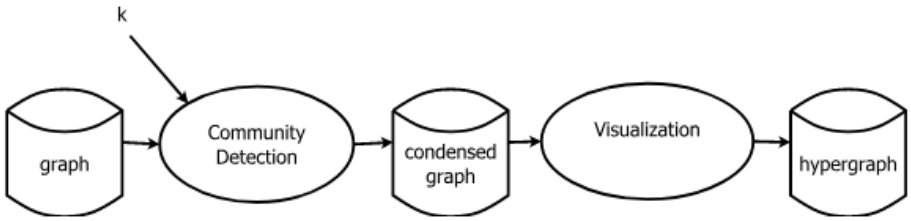


Figure 5. Proposed Method

The community detection algorithm firstly generates a set of k-quasi-cliques and stores them in the matrix $[a_{i,j}]$.Then, a partial set covering algorithm runs returning the minimum set of k-quasi-cliques.

Parameter *k* tunes the aggregation of the graph. The higher the value of *k* is, the greater the zoom in the graph is.

In section 3 the community detection algorithm is detailed, the visualization in the hypergraph is presented and a numeric example illustrates the method.

3.1 Community detection: generation of k-quasi-cliques

The transformation of a graph G(V,E) into a graph such that for every $i,j \in V$, the distance $d(i,j) \leq k$, is called the power graph G^k. To create a power graph G^k, the Floyd (1962) algorithm is used to find all the shortest paths in a graph, in an implementation with a worst-case complexity time of $O(|V|^3)$. Then, for each pair of edges with a distance less or equal to k, a new edge is added in the graph. Searching for maximal k-cliques in graph G is the same as searching for maximal cliques in the power graph G^k, i.e., k-clique(G) = clique(G^k).

To implement the first phase of Algorithm 1, the generation of several k-quasi-cliques, we use some previous work. Part of the described work in this subsection can be found in previous works (Cavique et al. 2002; Cavique and Luz 2009).

We define A(S) as the set of vertices which are adjacent to vertices of a current solution S. Let n=|S| be the cardinality of clique S and $A^k(S)$ the subset of vertices with k arcs incident in S. A(S) can be divided into subgroups $A(S) = \cup A^k(S)$, $k=1,...,n$. The cardinality of the vertex set |V| is equal to the sum of the adjacent vertices A(S) and the non-adjacent ones $A^0(S)$, plus |S|, resulting in $|V| = \Sigma|A^k(S)|+n$,

k= 0,...,n. For the quasi-clique problem, given $0 < \lambda \leq \gamma \leq 1$, the admissible subgroup is $A^{quasi}(S) = \bigcup_{i=\lambda..n}^{i=n} A^i(S)$, with n=|S|. For a given solution S, we define a neighborhood N(S) if it generates a feasible solution S'. In this work we are going to use only one neighborhood structure.

We consider the following notation:
N (S, γ) – neighborhood structure, where the quasi-clique edge density is $0 < \gamma \leq 1$
$N^+(S, \lambda, \gamma) = \{S' : S' = S \cup \{v^i\}, v^i \in A^{quasi}(S)\}$, where $A^{quasi}(S) = \bigcup_{i=\lambda..n}^{n} A^i(S)$ and n=|S|
S – the current solution
S*– the highest cardinality maximal clique found so far
G^k – power graph

Algorithm 1: Constructive Heuristic for the Maximum Quasi-Clique Problem
Input: power graph G^k; start sub-graph S; parameter γ
Output: maximal clique S*;

1. S*=S; λ = γ
2. while S is not maximal
2.1. choose S' from $N^+(S, \lambda, \gamma)$;
2.2. update S=S';
3. end while;
4. if edge density of N(S, γ) is valid then S*=S;
5. return S*;

In Algorithm 1, the neighborhood structure N(S, γ) is repeatedly used, by adding nodes until a maximal quasi-clique is reached. Parameters λ and γ are used in different phases. Parameter λ is used first, to obtain A(S), the nodes which are adjacent to the current nodes, and parameter γ, is used in the end to validate the edge density of the quasi-clique.

Comparing this version to previous ones, since we are using approximated algorithms to improve performance, we did not use other neighborhood structures to remove or swap nodes. For the same reasons, meta-heuristic features were not implemented.

In this work, to generate a diversified set of maximal k-quasi-cliques a multi-start algorithm is used, which calls for the Constructive Heuristic for the Maximum Clique Problem. Each generated k-quasi-clique corresponds to a column in the matrix $[a_{i,j}]$ referred previously in the partition and set covering problem mathematical formulations.

3.2 Community detection: partial set covering

In the partial set covering problem not-covered lines are allowed. Thus, each line corresponds to one of three possible states:

- not-covered or isolated node
- covered once or member of a cluster
- covered more than once, over-covered or broker

The desirable state is when the line is covered once, where the node is part of a single community. On the other hand, not-covered or over-covered lines are undesirable states, which correspond to isolated nodes or nodes that belong to two or more communities. So, the proposed objective function is defined by:

cost = not-covered + over-covered.

The optimal solution is when all lines are covered only once, obtaining a cost equal to zero.

To implement Algorithm 2, the Constructive Heuristic for Partial Covering is presented. The input for the partial cover is a matrix [a(i,j)] where the lines correspond to the nodes of the graph and each column is a k-partial-clique that covers a certain number of nodes. The input parameter $0 < \varphi \leq 1$ refers to the percentage of covered lines, so the heuristic will return at least (φ. I) covered lines, where *I* represents the total number of lines in matrix [a(i,j)]. We consider the following notation:

[a(i,j)] – input matrix with line i (or node i) and column j (or k-quasi-clique j)
[c(j)] – input vector of the cost of each column
I – number of lines of [a(i,j)]
J – number of columns of [a(i,j)]
R – remaining columns
S – the current solution
S* – the best cover solution
iter – number of iterations
cost(S) – objective function defined previously
parameter φ - percentage of covered lines

Algorithm 2: Constructive Heuristic for Partial Set Covering
Input: matrix [a(i,j)], vector [c(j)], parameter φ
Output: set covering S* with communities and brokers

1. $S^*=\emptyset$; iter=0;
2. while at least (φ.I) lines are not covered
2.1. R=[a(i,j)];
2.2. choose the best line $i^* \in R$ such as $|a(i^*,j)|=\min |a(i,j)| \ \forall j$;
2.3. choose the best column j* that covers line i* with minimal over-cover;
2.4. update $R=R \setminus a(i,j^*) \ \forall i$;
2.5. update $S=S \cup \{j^*\}$;
2.6. iter=iter+1;
3. end while;
4. sort cover S by descending order of costs;
5. for each S_i do if ($S \setminus S_i$ is still a partial φ-cover) then $S=S \setminus S_i$;
6. if (cost(S) < cost(S*)) S*=S;
7. return S*;

The Constructive Heuristic for Partial Set Covering reuses the Cover Heuristic (Chvatal 1979). As already stated in this section, meta-heuristic features were not implemented in this version.

For each iteration, a line is chosen to be covered and the best column that covers that line. Then, solution S and the remaining vertex R, are updated. The chosen line is usually the line that is more difficult to cover, i.e., the line that corresponds to fewer columns. The chosen column is the column that covers a larger number of lines, which includes the chosen line. When choosing the column, over-cover minimization will be considered. After reaching the cover set, the next step is removing redundancy, by sorting the cover set in descending order of cost and checking if each k-clique column is essential.

In this section, two relaxed algorithms are described. Since approximated features are used, it is not appropriate to optimize with meta-heuristics at the same time. These heuristics have low algorithmic complexity and the multi-start procedure can be decomposable, that is, it can run in a computer parallel environment.

The solution obtained is minimum partial set covering S*, where each column corresponds to a community. The over-covered lines coincide with brokers in the social network and the not-covered lines are isolated nodes.

3.3 Algorithm ComDetection (k, γ, φ)

Our community detection algorithm ComDetection (k, γ, φ), Algorithm 3, is divided into two phases and uses three parameters k, γ and φ.

Firstly, it generates a set of k-quasi-cliques and stores them in the matrix $[a_{i,j}]$. The parameter k is used to transform graph G into a power graph G^k and the parameter γ to run the maximal quasi-clique algorithm. Each generation of a k-quasi-clique is saved in a new column of matrix $[a_{i,j}]$.

Secondly, ComDetection reuses matrix $[a_{i,j}]$ and runs a partial set covering algorithm. Some values of parameter φ can be tested to find the best objective function. The minimum number of k-γ-cliques obtained corresponds to communities and brokers, which will be represented in a hypergraph. The algorithm can be specified as follows:

parameter k – defines the graph level of aggregation present in the power graph G^k
parameter γ – edge density, $0 < \gamma \leq 1$, in the relaxed quasi-clique problem
parameter φ – percentage of covered lines in the partial set covering

Algorithm 3: Algorithm ComDetection (k, γ, φ)
Input: graph G, parameters k, γ, φ
Output: communities and brokers information

1. Column generation of k-quasi-cliques
-- Input: graph G, parameters k, γ
-- Output: matrix $[a_{i,j}]$
1.1. Graph transformation into a power graph G^k
1.2. Add column to matrix $[a_{i,j}]$ using a new start:
1.2.1. Run the maximal quasi-clique algorithm (Algorithm 1)

2. Partial minimum set covering
-- Input: matrix $[a_{i,j}]$, parameter φ
-- Output: communities and brokers information
2.1. Run the partial minimum set covering algorithm with $[a_{i,j}]$ (Algorithm 2)

This work considers the common elements between partitions (over-covered elements or brokers) and elements that do not belong to any partition (not-covered elements or isolated nodes) and can be formulated as the partial φ-set covering problem with k-γ-cliques.

The algorithmic complexity for the column generation of k-quasi-cliques is given by $O(M.N^2)$ where M is the number of columns generated and N is the number of vertexes of the network. To generate the maximal cliques, a heuristic is used with $O(N^2)$. A detailed study of the time complexity of heuristics for the maximum clique can be found in Johnson (1979).

Given a matrix with M columns and N vertexes to be covered, for the partial minimum set covering, we use the approximating algorithm in Chvatal (1979) which provides the polynomial time complexity of O(ln N + 1), where N is the size of the set to be covered.

The final time complexity of algorithm ComDetection (k, γ, φ) is given by $O(M.N^2)$+ O(ln N+1).

3.4 Hypergraph Visualization

The visualization of dense networks with more than one hundred nodes is usually difficult or impossible. In this sub-section we describe a method which uses k-cliques and allows different levels of aggregation, from k=1 to k=diameter of the network. For each visualization with dimension k, a condensed graph is created, and a hypergraph can be mapped, identifying communities and brokers. To choose k, the number of communities and the capabilities of the software for visualizing hypergraphs should be considered. The visualization of different hypergraphs, varying k, allows the user to acquire a deeper knowledge of the network. In order to formalize the visualization method, we apply Algorithm 4.

Algorithm 4: Network visualization
Input: G, γ, φ
Output: hypergraph (k, γ, φ)
1. calculate the diameter of the graph
2. for k=1 to k=diameter: condensed graph = ComDetection (k, γ, φ)
3. choose k and map the hypergraph (k, γ, φ)

Each condensed graph G_c(k, γ, φ) stores the information about communities C and brokers, for vertices that overlap two or more communities, as shown in Table 1. Vertex 4 belongs to two communities C1 and C2, so it is called a broker. Brokers can cover more than two communities, for instance vertex 5 covers three communities. In G_c=(V,C) the information of the edges of G=(V,E) is not available.

Table 1. Condensed graph storing the quasi-cliques information

community vertice	C1	C2	C3	broker
1		1		
2		1		
3		1		
4	1	1		1
5	1	1	1	1
…	…	…	…	…

A graph can represent nodes connecting two communities. However, to link more than two communities a hypergraph is needed.

Hypergraph H=(V, E) (Berge 1970) consists of a set of vertexes V ={v_i: i=1, …, p}and a family E = {E_j : j = 1,…, q} of different subsets of the set of vertexes. E_j sets are called edges of a hypergraph or hyper-edges.

A hypergraph can be defined by an incidence matrix M(H) = [m_{ij}], i=1,…,p, j=1,…,q, where: $m_{ij} = \begin{cases} 1 \; if \; v_i \in Ej, \\ 0 \; if \; v_i \notin Ej. \end{cases}$

The transformation of the condensed graph G_c=(V,C) into a hypergraph is the following: communities C correspond to hyper-vertices and brokers B correspond to hyper-edges that link the communities in the hypergraph H=(C, B).

3.5 Numeric Example

A running example with 17 nodes, shown in Figure 6, illustrates how to obtain the visualization of the condensed network and the respective communities and brokers. The first network metric to consider is the diameter of the graph. In network17 the diameter is equal to 5.

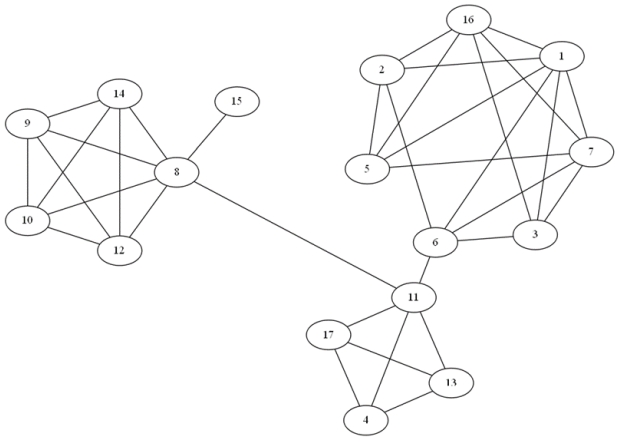


Figure 6. Network with 17 nodes and diameter 5

Given the diameter equal to 5 for network17, the algorithm ComDetection, with parameters γ=0.9 and φ=0.9, run 5 times, varying k. Table 2 presents the number of communities found for different values of k. The communities have a similar

structure, given by the value k, in other words, each community is a maximal k-clique, so the communities found are k-clique communities. The partition of the network must be composed by similar components, in this case, k-cliques are maximal cliques of a graph G^k. In the example, the number of communities decreases from 4 to 1. As expected, when k is equal to the diameter only one community can be found.

Table 2. Number of communities with ComDetection (k, 0.9, 0.9)

graph	#nodes	diameter	# communities				
			k=1	k=2	k=3	k=4	k=5
network17	17	5	4	3	2	1	1

In this running example, we choose k=2. The number of communities are 3, {A, B, C}, as shown in Table 3. The set covering problem returns 3 columns, or 3 communities, which cover all the vertices, from 1 to 17. Some of the vertices are over-covered, like nodes 6, 8 and 11, which means these nodes belong to more than one community. As stated before, the actors that belong to different communities are called brokers. In this example nodes 6 and 8 are related to 2 communities and node 11 is linked with 3 communities.

Table 3. Condensed graph provided by ComDetection (2, 0.9, 0.9)

community vertice	A	B	C	brokers of 2 communities	brokers of 3 communities
1			1		
2			1		
3			1		
4		1			
5			1		
6		1	1	1	
7			1		
8	1	1		1	
9	1				
10	1				
11	1	1	1		1
12	1				
13		1			
14	1				
15	1				
16			1		
17		1			

Figure 7, on the left, shows the incidence matrix of a hypergraph, where the columns correspond to the hyper-edges and the lines to the communities. In this example, communities {A, B, C} are reduced to vertices in the hypergraph. The brokers are the hyper-edges that associate the communities. Figure 7, on the right, shows the hypergraph that condenses the information of the algorithm ComDetection (2,0.9,0.9).

hyper-edge community	e6	e8	e11
A		1	1
B	1	1	1
C	1		1

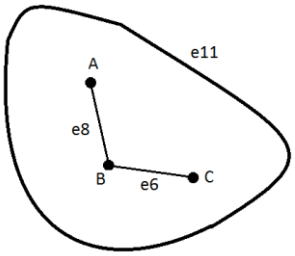


Figure 7. Incidence matrix (left) and hypergraph of ComDetection (2, 0.9, 0.9) (right)

Comparing the hypergraph with the original network, it is clear nodes 6 and 8 are close to two communities and node 11 is central in the network. Node 11 is central because there is a path of length 2 reaching every node of the graph.

The cardinality of the set of all hyper-edges incident to vertex v_i is called the degree of vertex v_i and is denoted as deg v_i. In the example the deg(A)=deg(C)=2 and deg(B)=3. A graph can be seen as a particular case of a hypergraph with degrees of all the edges equal to 2.

In this section the visualization procedure was presented. Given a network and chosen dimension k, the partition into k-clique communities is performed. As a result the brokers are found, and the condensed network can be represented in a hypergraph. This approach allows the user to discover the structure of the network, by reducing data and allowing the visualization of a condensed view.

4 Computational Results

To implement the computational results of these algorithms, some choices such as the computational environment, the performance measures and the datasets must be made.

The computer programs were written in C language and the Dev-C++ compiler was used. The computational results were obtained from an Intel Core Duo CPU 3.00 GHz processor with 4.00 GB of main memory running under the Windows 10 operating system.

Two performance measures are considered: (i) the computational time and the quality of the community detection algorithms and (ii) the quality of the visualization of the condensed network and the ability to identify brokers. For the first performance

measure the LFR benchmark datasets (Lancichinetti et al. 2008) are used and for the second measure a set of families of graphs is used.

4.1 Computational runtime and community detection performance

Lancichinetti, Fortunato and Radicchi (2008), introduced the LFR family of graphs to test algorithms in the identification of communities in networks, which considers the heterogeneity in the distribution of node degrees and of community sizes. In the generation of the datasets for the LFR benchmark we use the minimum number of parameters, as follows:

- N – number of nodes,
- deg – average degree of the nodes,
- max_deg – maximum degree,
- μ – mixing parameter.

The mixing parameter, μ, varies between 0 and 1, $0 \leq \mu \leq 1$, where each node shares $1-\mu$ of its links with the nodes of the same community and the fraction μ with the nodes of other communities.

Six datasets were generated from N=500 until N=5000, with similar average degree, max degree and mixing parameter, μ. The calculation of the diameter is important in this work, where the same value equal to 3 was found for all datasets.

The runtime of the two algorithms of ComDetection is shown in Table 4. Given the high computational runtime of the multi-start algorithm and since the decomposition of the algorithm is possible, it could run in a parallel computer environment. On the other hand, the partial-cover algorithm performs very well in the ComDetection (1, γ, φ), with a runtime of 6 seconds for N=5000.

Table 4. Runtime of the LFR dataset with k=1

LFR dataset				ComDetection (1, γ, φ)	
N	deg	max deg	diameter	quasi-clique generator time (sec)	partial cover time (sec)
500	50	70	3	2	1
1000	100	140	3	17	1
2000	200	280	3	264	4
3000	300	420	3	980	5
4000	400	560	3	3701	5
5000	500	700	3	4931	6

The polynomial regression $y = 0.0003x^2 - 0.46466x + 117.04$ with a $R^2=0.9669$ is found for the time complexity of the quasi-clique generator.

The quality of the community detection algorithm is reported in Table 5. In the generation of the LFR dataset we use parameter μ equal to 0.10, and the number of communities retrieved by the generator is between 8 and 9.

As the maximum clique and the set covering are highly constrained problems, we used relaxations of each problem. The quasi-cliques with parameter γ and the partial

covering with parameter φ, are the components of the algorithm ComDetection (k, γ, φ). To run ComDetection (1, γ, φ) we use k=1 and vary the parameters γ and φ. We reuse the work of Brutano et al. (2008) for the parameters of the quasi-cliques, by using λ and γ, such that, $0 < \lambda \leq \gamma \leq 1$, where λ reveals a local parameter and γ a general one. Running Algorithm 2, to generate 1-γ-cliques, the local parameter λ was set equal to 0.60 and the general parameter γ was assigned equal to 0.80.

In the partial set covering problem, the distance to the optimal solution is given by %Opt= 1 - $\frac{cost}{N}$, where the cost, or objective function, was defined by the sum of the not-covered lines plus the over-covered ones, and N corresponds to the total number of lines. After running Algorithm 2, the partial set covering, the best values for φ are between 0.85 and 0.90 and the correspondent %Opt is around the meaningful value of 88%, as presented in Table 5.

The quality of the community detection algorithm is also measured by comparing the given number of communities and the communities found by the algorithm, which forms a perfect match, as shown in Table 5 by comparing the number of given communities and the number of found communities.

Let Q=(V,E) be the quasi-clique obtained with Algorithm 1. To check the parameter μ of the LFR generator, we calculate μ' as follows:

$$\mu'(Q(V,E)) = 1 - \frac{|E|}{\sum_i \deg(v_i)\,/2}$$

In Table 5 we can verify that μ and μ' are the same size, but they have different values, respectively 0.10 and 0.16.

Table 5. Quality of the solutions of the LFR dataset with k=1

LFR dataset			ComDetection (1, γ, φ)						
N	μ	# given communities	γ	λ	φ	objective function	% Opt	# found communities	μ'
500	0.10	9	0.80	0.60	0.90	30	94%	9	0.15
1000	0.10	9	0.80	0.60	0.90	54	95%	9	0.16
2000	0.10	8	0.80	0.60	0.85	548	73%	8	0.16
3000	0.10	9	0.80	0.60	0.90	348	88%	9	0.16
4000	0.10	8	0.80	0.60	0.85	496	88%	8	0.16
5000	0.10	9	0.80	0.60	0.90	360	93%	9	0.16

4.2 Visualization of the condensed network

To validate the quality of the visualization of the condensed network and the ability to identify brokers we use different families of graphs.

The first two families of graphs contain some clique instances from the second DIMACS (1995) challenge. These include the "brock" graphs, which contain cliques hidden within much smaller cliques, increasing the difficulty of discovering complete subgraphs, and the "c-fat" graphs which are a result of fault diagnosis data.

The “CS-phd”, “erdos” and “roget” datasets were selected from the Pajek networks library (Batagelj and Mrvar 2006). The CS-phd network contains the ties between Ph.D. students and their advisors in computer science, where each edge links an advisor to a student. In the Erdös network, each node corresponds to a researcher, and two nodes are adjacent if the researchers published together. The graphs are named “erdos-x-y”, where “x” represents the last two digits of the year the graphs were created, and “y”, the maximum distance from Erdös to each vertex in the graph. The Roget network is based on Roget's Thesaurus published in 1852.

The LFR graph family, whose experiments were described in the previous sub-section, was also included.

Table 6. Number of communities for different graph families

graph	#nodes	diameter	# communities												
			k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=9	k=10	k=11	k=18	k=28	k=40
brock200_1	200	2	24	1											
brock200_2	200	2	26	1											
brock400_1	400	2	26	1											
brock400_2	400	2	23	1											
c-fat200-1	200	18	16	11	8	8	7	7	6	5	4	2	1		
c-fat200-2	200	9	9	7	5	4	3	3	3	1					
c-fat500-1	500	40	16	12	9	7	7	6	6	4	4	3	3	1	1
CSphd	1882	28	76	47	36	30	26	21	19	16	13	9	7	1	
erdos-97-1	472	6	9	8	7	7	4	1							
erdos-98-1	485	7	8	8	7	5	1	1	1						
erdos-99-1	492	7	8	8	7	7	1	1	1						
LFR-1000	1000	3	9	1	1										
LFR-2000	2000	3	8	1	1										
roget	1022	10	59	36	16	9	1	1	1	1	1				

As stated in Algorithm 4, to better understand the power graph G^k of each instance and the features of the family of graphs, we run the algorithm for different values of k. For the analysis of each graph, we consider the number of nodes, the diameter, and the number of communities with k varying from 1 to the diameter, as shown in Table 6.

The k-sequence with the number of communities identifies each family of graphs and is very promising in social network analysis. Each sequence returns a different pattern for each family of networks.

The brock graphs, known as hiding cliques, have a diameter equal to 2, and to cover the graph, 1-set of 2-cliques is enough. On the other hand, the c-fat graphs have large diameters, generating long k-sequences.

For erdos-97-1 and erdos-99-1, with the diameter of 11, networks are covered with only one k-clique, with k=6 and k=7 respectively, repeating the solution until k=diameter. The same happens with Roget network, with a diameter of 10, where k>=5 reaches one community. Alternatively, in the CS-phd network the number of communities decreases slowly until k=diameter.

Finally, LFR graphs, with diameter equal 3, present a pattern similar to the brock graphs.

Visualization of graph c-fat-200-1

This sequence with k dimensions allows the final user to choose the most appropriate dimension. To exemplify the visualization method with a larger graph than in the previous section, we choose graph c-fat200-1 and k=3. With dimension k=3, 8 communities are retrieved, which falls within the mapping rule of thumb of 7±2 communities. After running ComDetection (3, 0.8, 0.95), 8 communities are found, and the incidence matrix of the hypergraph is shown in Table 7. The number of brokers of each hyper-edge is denoted by the hyper-edge weight.

Table 7. Incidence Matrix of hypergraph of c-fat200-1 for ComDetection (3, 0.8, 0.95)

node \ hyper-edge	eAB	eBC	eCD	eDE	eEG	eAF	eFH	eABF
A	1					1		1
B	1	1						1
C		1	1					
D			1	1				
E				1	1			
F						1	1	1
G					1			
H							1	
weight	10	2	22	2	3	10	2	12

In Figure 8, for the hyper-edges with only two nodes a simplified representation of the hypergraph is used, adopting a line between the nodes.

Figure 8.a represents the hypergraph for k3-c-fat200-1 with eight communities using parameters γ=0.8 and φ=0.95. Most of the hyper-edges link two communities and only hyper-edge eABF links three communities.

Varying parameter k, after running ComDetection (9, 0.8, 0.95), in Figure 8.b, the hypergraph for k9-cfat200-1 with parameters γ=0.8 and φ=0.95 is shown with five communities.

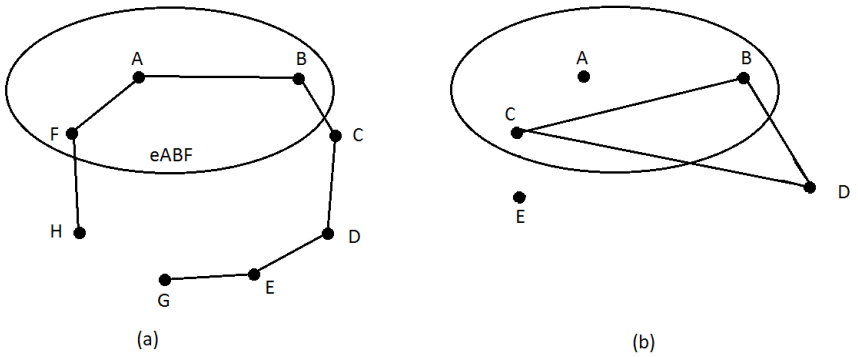


Figure 8. Hypergraph for k3-c-fat200-1 and k9-c-fat200-1

Threats and opportunities can be found in the analysis of the hypergraph. The threats are associated with network resilience, for instance, hyper-edges with a low weight, like eBC, eDE and eFH, in Figure 8.a, seem more susceptible than the others. The opportunities are related to structural holes, for instance, community E, in Figure 8.b, seems very distant from the other groups, and therefore a broker which links the communities could bring new advantages.

5 Conclusions

The comprehension of social network phenomena is closely related to data visualization. The aim of this work was to create a method to identify communities and brokers. In this document three major contributions are highlighted and presented in three distinct levels:

- Quality: the computational experiments show competitive results in community detection with the LRF benchmark datasets;
- Usability: algorithm ComDetection (k, γ, φ) allows the network reduction with different *k* levels;
- Visualization: an innovative approach allows the transformation of the condensed structure into a hypergraph.

To measure the quality of the community detection algorithm, in the computational results, the LFR benchmark was used with very competitive results.

As network partition is a hard problem, we used problem relaxations. The two-phase algorithm ComDetection (k, γ, φ) uses three parameters – parameter k tunes the dimension of visualization, parameter γ is the density of the quasi-clique and parameter φ the percentage of the partial covering set. The algorithm first generates a set of k-quasi-cliques. Then a partial set covering heuristic is applied to identify groups and influential nodes. The social networks analysts have often referred the problematic of structural holes and brokerage. Automatic procedures are limited in finding, not only the communities but also the actors that play within them. In this work the data extracted from social networks goes further than the study of communities, allowing the finding of the brokers that interact among groups.

In the visualization process, the final user can choose the most appropriate k-value, from the k-sequence, and draw the respective hypergraph that condenses the network information. In the analysis of the hypergraph, strategies for threats and opportunities should be considered.

Most community detection algorithms do not consider different levels of node aggregation and their visualization. On the other hand, the clustered graphs layout applications do not regard the interactions among communities. We believe our method combines community detection and clustered graph layout features. In our work a parameter *k* tunes the level of aggregation of the network, allowing different drawings, where brokers interact with communities.

Acknowledgements
The first author would like to thank the FCT UID/Multi/04046/2013 for its support.

REFERENCES

1. Abello J., M. Resende, S. Sudarsky, 2002, Massive quasi-clique detection, Proceedings of the Latin-American Symposium on Theoretical Informatics, pp. 598–612.
2. Aggarwal C.C., S. Lin, P.S. Yu, 2012, On Influential Node Discovery in Dynamic Social Networks, Proceedings of the Twelfth SIAM International Conference on Data Mining, pp. 636-647.
3. Araujo, M., S. Gunnemann, G. Mateos, C. Faloutsos, 2014, Beyond blocks: Hyperbolic community detection, in Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), Nancy, France.
4. Barabási, A.-L., 2016, Network Science, Cambridge University Press.
5. Batagelj V., A. Mrvar, 2006, Pajek datasets, URL: http://vlado.fmf.uni-lj.si/pub/networks/data/
6. Berge C., 1970, Graphes et Hypergraphes, Dunod, Paris.
7. Bilal, N., P. Galinier, F. Guibault, 2014, An iterated-tabu-search heuristic for a variant of the partial set covering problem, Journal Heuristics 20, pp. 143–16.
8. Blondel, V. D., J.-L. Guillaume, R. Lambiotte and E. Lefebvre, 2008, Fast unfolding of communities in large networks, Journal Statistical Mechanics, P10008.
9. Borgatti, S., M. Everett, L. Freeman, 2002, UCINET for Windows: Software for social network analysis.
10. Bourqui, R., D. Auber, P. Mary, 2007, How to draw clustered-weighted graphs using a multilevel force-directed graph drawing algorithm, in Proceedings of the 11th International Conference on Information Visualization, IV'07, pp. 757 –764.
11. Brunato M., H.H. Hoos, R. Battiti, 2008, On Effectively Finding Maximal Quasi-cliques in Graphs, In: Maniezzo V., Battiti R., Watson JP. (eds) Learning and Intelligent Optimization, LION 2007, Lecture Notes in Computer Science, vol 5313. Springer, Berlin, Heidelberg.
12. Burt R.S., 1992, Structural Holes: The Social Structure of Competition, Harvard University Press.
13. Burt, R.S., 2005, Brokerage and closure: an introduction to social capital, New York: Oxford University Press.

14. Cavique L., A.B. Mendes, J.M.A. Santos, 2009, An Algorithm to Discover the k-Clique Cover in Networks, in Progress in Artificial Intelligence, L. Seabra Lopes et al. (Eds.), EPIA 2009, LNAI 5816, Springer-Verlag Berlin Heidelberg, pp. 363-373.
15. Cavique L., A.B. Mendes, J.M.A. Santos, 2012, Clique Communities in Social Networks, in Quantitative Modelling in Marketing and Management, World Scientific Publisher, edited by Luiz Moutinho and Kun-Huang Huarng, pp. 469-490.
16. Cavique L., C. Rego, I. Themido, 2002, A scatter search algorithm for the maximum clique problem, In Ribeiro, C. e Hansen, P. (Eds.) Essays and Surveys in Meta-heuristics, Kluwer Academic Pubs., Dordrecht, The Netherlands, pp. 227-244.
17. Cavique L., C.J. Luz, 2009, A heuristic for the stability number of a graph based on convex quadratic programming and tabu search, Journal of Mathematical Sciences, 161 (6), pp. 944-955.
18. Cavique, L., N.C. Marques and J.M.A. Santos, 2014, An Algorithm to Condensed Social Networks and Identify Brokers, Advances in Artificial Intelligence, IBERAMIA, Lecture Notes in Computer Science 8864, Springer, ISBN 978-3-319-12026-3, pp. 331-343.
19. Christofides N., J. Paixão, 1993, Algorithms for large scale set covering problems, Annals of Operations Research, vol. 43(5), pp. 259-277.
20. Chvatal V., 1979, A greedy heuristic for the set-covering problem, Mathematics of Operations Research, vol. 4, pp. 233–235.
21. Cruz, J.D., C. Bothorel, F. Poulet, 2014, Community detection and visualization in social networks: Integrating structural and semantic information., ACM Transactions Intelligent Systems and Technology, vol. 5, 1, Article 11, 26 pages, January.
22. Derenyi I., G. Palla, T. Vicsek, 2005, Clique Percolation in Random Networks, Physical Review Letters, vol. 94(16), pp. 160202.
23. Didimo, W., F. Montecchiani, 2014, Fast layout computation of clustered networks: algorithmic advances and experimental analysis, Inf. Sci., vol. 260 (1), pp. 185-199.
24. DIMACS, 1995, Maximum clique, graph coloring, and satisfiability, Second DIMACS implementation challenge, URL http://dimacs.rutgers.edu/ Challenges/.
25. Eades, P., Feng, Q.-W., 1997, Multilevel visualization of clustered graphs, in Graph Drawing, S. North, Ed., Lecture Notes in Computer Science, vol. 1190, Springer, pp. 101–112.
26. Eades, P., Huang, M. L., 2000, Navigating Clustered Graphs using Force-Directed Methods, Journal of Graph Algorithms and Applications, vol. 4 (3), pp. 157-181.
27. Easley D., J. Kleinberg, 2010, Networks, Crowds and Markets: Reasoning About a Highly Connected World, Cambridge University Press.
28. Erdos, P., Renyi A., 1959, On Random Graphs. I., Publicationes Mathematicae 6, 290–297.
29. Floyd, R.W, 1962, Algorithm 97: Shortest Path, Communications of the ACM, vol. 5(6), pp.345.
30. Fortunato S., 2010, Community detection in graphs, Physics Reports 486, pp. 75-174.
31. Girvan M. and M. E. Newman, 2002, Community structure in social and biological networks, Proceedings of the National Academy of Science U.S.A., 99(12), pp. 7821-7826.

32. Granovetter M., 1973, The strength of weak ties, American Journal of Sociology, 78, pp.1360–1380.
33. Grossman, J., P. Ion, R.D. Castro, 2007, The Erdos number Project, URL http://www.oakland.edu/enp/.
34. Harary, F., 1969, Graph theory, Addison-Wesley Publishing Company.
35. Hespanha, J.P., 2004, An efficient matlab algorithm for graph partitioning, Department of Electrical and Computer Engineering, University of California, Santa Barbara.
36. Johnson, D.S., 1974, Approximation Algorithms for Combinatorial Problems, Journal of Computer and System Sciences vol. 9, pp.256-278.
37. Kang U., C. Faloutsos, 2011, Beyond 'Caveman Communities': Hubs and Spokes for Graph Compression and Mining, in Proceedings of the 11th IEEE International Conference on Data Mining (ICDM), Vancouver, Canada, pp. 300–309.
38. Karypis, G., V. Kumar, 1999, Multilevel k-way Hypergraph Partitioning. In Proceedings of the IEEE 36th Conference on Design Automation Conference (DAC), New Orleans, LA.
39. Kernighan B.W., S. Lin, 1970, An efficient heuristic procedure for partitioning graphs, Bell Systems Technical Journal, vol. 49, pp. 291–307.
40. Lancichinetti, A., S. Fortunato, and F. Radicchi, 2008, Benchmark graphs for testing community detection algorithms, Physical Review, The American Physical Society, E 78(4), 046110, pp. 1-5.
41. Li, W., M. Takatsuka, 2004, Adding filtering to geometric distortion to visualize a clustered graph on small screens, in Proceedings of the Australasian Symposium on Information Visualisation (APVis'04), vol. 35. Australian Computer Society, pp. 71–79.
42. Liu, Y., T. Safavi, N. Shah, D. Koutra, 2018, Reducing large graphs to small supergraphs: a unified approach, Social Network Analysis and Mining, vol.2018,1.
43. Long, J.C, F.C. Cunningham, J. Braithwaite, 2013, Bridges, brokers and boundary spanners in collaborative networks: a systematic review, BMC Health Services Research 13:158.
44. Luce, R.D., 1950, Connectivity and generalized cliques in sociometric group structure, Psychometrika, vol.15, pp. 159-190.
45. Milgram, S., 1967, The Small World Problem, Psychology Today, 1(1), pp. 60-67.
46. Moreno, J.L., 1934, Who Shall Survive? Nervous and Mental Disease Publishing Company, Washington DC.
47. Newman, M.E.J., 2010, Networks: An Introduction, Oxford University Press.
48. Pei J., D. Jiang, A. Zhang, 2005, On mining cross-graph quasi-cliques, In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'05).
49. Richards, W.D. (1995), NEGOPY 4.30 Manual and Users's Guide, School of Communication Simon Fraser University, Burnaby, Canada.
50. Richards, W.D., R.E. Rice, 1981, The NEGOPY network analysis program, Social Networks, vol. 3, 2, pp. 15-223.
51. Rogers, E.M., D.L. Kincaid, 1981, Communication Networks: Toward a New Paradigm for Research, New York, Free Press.
52. Scott J., 1991, Social Network Analysis: Handbook, SAGE Publications Ltd.

53. Soriano P., M. Gendreau, 1996, Tabu search algorithms for the maximum clique, In: Johnson, D.S.; Trick, M.A. (Eds.). Clique, Coloring and Satisfiability, Second Implementation Challenge DIMACS, American Mahematical Society, pp. 221-242.
54. Wasserman, S., K. Faust, 1995, Social Network Analysis: Methods and applications, Cambridge University Press.
55. Watts, D.J., Strogatz, S.H., 1998, Collective dynamics of small-world networks, Nature 393(6684), 409–10.
56. Yang, J., J. Leskovec, 2013, Overlapping community detection at scale: a nonnegative matrix factorization approach, in Proceeding of the 6th ACM International Conference on Web Search and Data Mining (WSDM), ACM.