Metadata of the chapter that will be visualized in SpringerLink

Book Title	Advances in Artificial Intelligence -- IBERAMIA 2014
Series Title	
Chapter Title	An Algorithm to Condense Social Networks and Identify Brokers
Copyright Year	2014
Copyright HolderName	Springer International Publishing Switzerland

Corresponding Author	Family Name	**Cavique**
	Particle	
	Given Name	**Luís**
	Prefix	
	Suffix	
	Division	
	Organization	LabMAg and Universidade Aberta
	Address	Lisboa, Portugal
	Email	lcavique@uab.pt
Author	Family Name	**Marques**
	Particle	
	Given Name	**Nuno C.**
	Prefix	
	Suffix	
	Division	CITI and FCT
	Organization	Universidade Nova Lisboa
	Address	Lisbon, Portugal
	Email	nmm@fct.unl.pt
Author	Family Name	**Santos**
	Particle	
	Given Name	**Jorge M. A.**
	Prefix	
	Suffix	
	Division	
	Organization	Universidade de Évora
	Address	Évora, Portugal
	Email	jmas@uevora.pt

Abstract

In social network analysis the identification of communities and the discovery of brokers is a very important issue. Community detection typically uses partition techniques. In this work the information extracted from social networking goes beyond cohesive groups, enabling the discovery of brokers that interact between communities. The partition is found using a set covering formulation, which allows the identification of actors that link two or more dense groups. Our algorithm returns the needed information to create a good visualization of large networks, using a condensed graph with the identification of the brokers.

Keywords (separated by '-') Data mining - Graph mining - Social networks - Condensed network - Brokerage

An Algorithm to Condense Social Networks and Identify Brokers

Luís Cavique[1(✉)], Nuno C. Marques[2], and Jorge M.A. Santos[3]

[1] LabMAg and Universidade Aberta, Lisboa, Portugal
lcavique@uab.pt
[2] CITI and FCT, Universidade Nova Lisboa, Lisbon, Portugal
nmm@fct.unl.pt
[3] Universidade de Évora, Évora, Portugal
jmas@uevora.pt

Abstract. In social network analysis the identification of communities and the discovery of brokers is a very important issue. Community detection typically uses partition techniques. In this work the information extracted from social networking goes beyond cohesive groups, enabling the discovery of brokers that interact between communities. The partition is found using a set covering formulation, which allows the identification of actors that link two or more dense groups. Our algorithm returns the needed information to create a good visualization of large networks, using a condensed graph with the identification of the brokers.

Keywords: Data mining · Graph mining · Social networks · Condensed network · Brokerage

1 Introduction

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

The visualization of a small number of vertices can be easily mapped. However, when the number of vertices and edges increases, the visualization of the whole graph becomes incomprehensible as the large amount of available data in corporations and governments becomes incompatible with the complete drawing. There is a pressing need for new metrics and pattern recognition tools to explore and visualize large social networks.

Brokers can be defined as actors that work within communities. Although different methods are used to find network partitions [14], the specific discovery of brokers between partitions is scarce [21]. In this work we are interested in finding, not only the communities, but also the brokers. We also intend to show that the visualization of the brokers in a condensed graph undoubtedly simplifies the work of the social network analyst when dealing with large networks.

A.L.C. Bazzan and K. Pichara (Eds.): IBERAMIA 2014, LNAI 8864, pp. 331–343, 2014.
DOI: 10.1007/978-3-319-12027-0_27

In Figure 1, a small network is shown. A social network analyst should define the communities and the actors within them. The dense group {1,3,4,5} can be easily identified, but the rest of the network is more sparse. Three groups of nodes can also be recognized, although a formal definition of community is needed. In this work we define community as a k-clique [20].

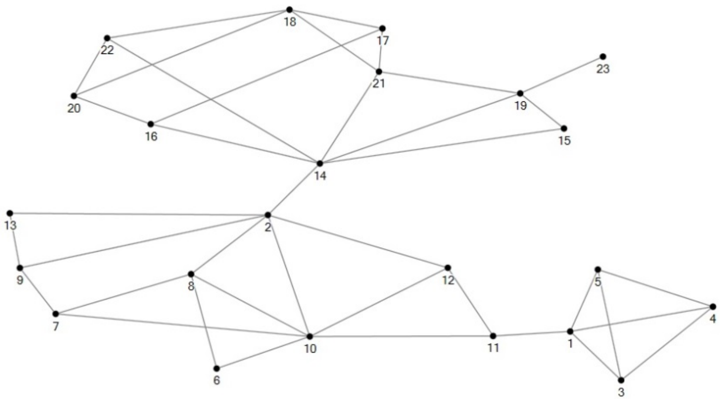

Fig. 1. Network for the running example

A strategy to condense the graph associated with the identification of brokers is presented in this work. The partition strategy is found using a set covering formulation with k-cliques [3], which allows over-covered nodes and isolated nodes.

In this study the words “network” and “graph” are used as synonyms. The terms “community”, “dense group”, “connected component”, “clique” or “k-clique” are also equivalents.

In Section 2, we present the concept of brokerage in social networks and the two combinatorial problems related to the k-clique covering. In sub-section 2.1 structural holes, bridges and brokers in social networks are presented. In sub-section 2.2 the generation of cliques and k-cliques is introduced. And finally, in sub-section 2.3 the set covering problem is defined. In Section 3, we present the two phase algorithm to condense the graph and discover brokers in social networks. In Section 4, the computational results are presented. Finally, in Section 5 we draw some conclusions.

2 Related Work

The graph theory related work in this paper combines the areas of graph visualization, graph mining, social network concepts, sub-graphs and graph partition.

The aim of graph visualization techniques is to achieve the comprehension of the data by providing intuitive layouts associated with interactive functionalities. In Tarawneh et al. [25] five main areas in graph layout algorithms are referred: node-link layouts, tree layout, matrix visualization, 3D layout and nodes-and-edges clustering. The goal of clustering techniques is to reduce visual disorder in the final layout. Reducing the number of elements, edges and/or nodes, will increase the clarity of the visualization. Clustering algorithms can be divided into two groups: edge compression and nodes compression. The edge clustering approach [13] replaces individual edges with edges connected to groups of nodes. Modular decomposition and power-graph decomposition [22], are the most well-known edge clustering techniques. The second group, nodes clustering, based on a specified criteria divides the graph into different sets, and then reduces each set to a node. In Auber et al. [1] the clustering algorithms are applied to small-word networks.

The area of Graph Mining has seen significant growth in the last decade [7]. The use of cliques is also referred to in Du et al. [12]. A similar concept of brokerage is the communities overlapping in networks, when nodes belong to multiple dense groups [26].

In our work we use clustering techniques in order to find partition, but with an additional purpose – to find influential nodes that link two or more partition sets, the brokers. In order to condense the network and to discover brokers, three different concepts are combined: the brokerage in social networks and two combinatorial problems – the maximal k-clique generation and the set covering problem.

2.1 Social Network Concepts

In the late 1960s, while working on his Ph.D., Mark Granovetter interviewed people who had recently changed jobs, in order to come to a conclusion as to how they had found their new jobs. Surprisingly, he realized that the information about the new jobs had come from distant acquaintances instead of close friends. The concept of strong and weak ties [17] introduced a novel principle in social networks. Weak ties are valuable because they will more likely be the source of novel information, surprise and openness to new worlds. On the other hand, strong ties intensify group cohesion and the persistence of group identity. This resulted in 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 [14].

Following this issue, R.S. Burt [2] developed a complementary approach coined Structural Holes, referring to the absence of links in a connected organization. He also introduced the concept of brokerage, signifying nodes that connect two dense groups.

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:

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

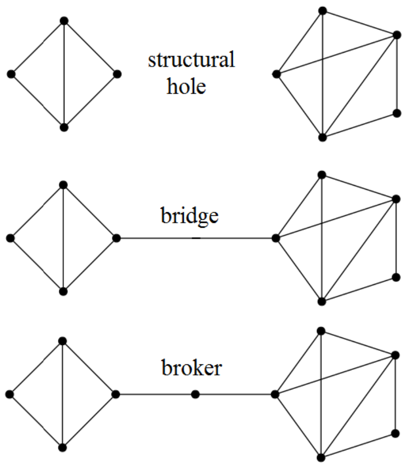

Fig. 2. Structural hole, bridge and broker

2.2 Generating k-Cliques

In order to find dense groups, components or communities, we are going to use the concept of k-clique.

Given an undirected graph G=(V, E), where V denotes the set of vertices (or nodes) and E, the set of edges (or arcs), graph G_1= (V_1, E_1) is called a sub-graph of G, if $V_1 \subseteq V$, $E_1 \subseteq E$ and for every edge (v_i, v_j)$\in$ E_1, the vertices $v_i, v_j \in V_1$. A sub-graph G_1 is said to be complete, if there is an edge for each pair of vertices. A complete graph is also called a clique.

A clique is maximal if it is not contained in another clique, while the maximum clique is the clique with maximum cardinality. The Maximum Clique is an NP-hard problem. In order to find a lower bound for the maximization problem, the heuristics proposed by Johnson [18] and the meta-heuristic that uses Tabu Search [16], developed by Soriano and Gendreau [24] can be used. Following works can be found in Cavique et al. [6] and Cavique and Luz [4].

The clique structure is very restrictive, since there must be an edge for each pair of vertices. More relaxed approaches were suggested in social sciences, such as k-cliques, k-clans, k-clubs and k-plexs [23].

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, we use the k^{th} power of graph G in such a way that we can reuse an already well-known algorithm, the maximum clique algorithm. The transformation process adds edges to reach length *k* between every pair of nodes.

In Figure 2, each dense group is a 2-clique sub-graph, since the maximal distance between any pair of vertices is equal to 2.

Graph G^2 is equivalent to Granovetter's Triadic Closure, where a new edge is inserted because the distance between the pair of nodes is equal to 2. In this approach strong ties correspond to friends, and weak ties correspond to a friend's friend.

2.3 Network Partition

Community or dense 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, Lin [19] algorithm, which finds a partition of the nodes by dividing the data 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.

Recent studies, based on physics, introduced the concept of clique percolation [11], where the network is viewed as a union of cliques.

The Girvan-Newman [15] 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 [14].

A more relaxed problem that allows a node to share two components is the 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 Set Covering problem. The matrix $[a_{i,j}]$ stores the information about the different communities and for each attribute *x*, a cost can be associated by using a vector c_j. 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 $x_j \in \{0,1\} \qquad j=1,\ldots,n$

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

The set covering heuristic, proposed by Chvatal [8], repeats the process by choosing the line with fewer elements, followed by the choice of the column with the best ratio considering the cost of the column and the number of lines covered. This constructive heuristic is improved by using a Tabu Search heuristic that removes the most redundant columns and re-builds a new solution [5].

3 The Two-Phase Algorithm

In order to simply the visualization of the network, a condensed network is a graph where some of the nodes represent communities. In this condensed network the nodes that correspond to k-cliques are shrunk.

Although there are several partition algorithms, there are few studies about the linkage between them, namely concerning issues related to brokerage. In this paper we present a new approach that takes into account the common elements between partitions (over-covered) and elements that do not belong to any partition (isolated nodes), formulated as the set covering problem with k-cliques.

The Two-phase Algorithm, firstly, generates several k-cliques ($[a_{i,j}]$columns) from the network and secondly, runs the set covering problem in order to find the minimum number of k-cliques which cover all the vertices. The algorithm can be specified as follows.

Algorithm 1: The Two-phase Algorithm
Input: network/graph G, distance k
Output: condensed network/graph G_C^k
1) Generate maximal k-cliques columns
2) Run the set covering problem

This work is an extension of the work presented by Cavique et al. [3]. The novelty of this work is the identification of brokers and the generation of condensed networks which allows a clear visualization in large networks.

3.1 The Set Covering Problem with k-Cliques

In this work, to generate a large set of maximal k-cliques a multi-start algorithm is used, which calls for the Tabu Heuristic for the Maximum Clique Problem. To each generated k-clique will correspond a column of the matrix $[a_{i,j}]$ referred in the partition and Set Covering problem formulations.

To implement the first step of Algorithm 1, the generation of several k-cliques, we use some previous work. Finding a maximal clique in a k-graph is the same as finding a maximal k-clique in a graph. Part of the described work in this sub-section can also be found in Cavique et al. [6] and Cavique and Luz [4].

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 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 three neighborhood structures. We consider the following notation:

$N^+ (S) = \{S': S'= S \cup\{v^i\}, v^i\in A^n(S)\}$
$N^- (S) = \{S': S'= S \setminus\{v^i\}, v^i\in S\}$
$N^0 (S) = \{S': S'= S \cup\{v^i\}\setminus\{v^k\}, v^i\in A^{n-1}(S), v^k\in S\}$
S – the current solution
S*– the highest cardinality maximal clique found so far
T– the tabu list
N(S) – neighborhood structures

Algorithm 2: Tabu Heuristic for the Maximum Clique Problem
Input: k-Graph; start sub-graph S;
Output: maximal clique S*;
1. T=∅; S*=S;
2. while not end condition
2.1. if (N^+(S)\T ≠ null) choose the maximum S';
2.2. else if (N^0(S)\T ≠ null) choose the maximum S'; update T;
2.2.1. else choose the maximum S' in N^-(S); update T;
2.3. update S=S';
2.4. if (|S|>|S*|) S*=S;
3. end while;
4. return S*;

To implement the second step of Algorithm 1, the input for the k-clique cover is a matrix where the lines correspond to the vertices of the graph and each column is a k-clique that covers a certain number of vertices. We consider the following notation:

$[a_{i,j}]$ – input matrix with j columns
$[c_j]$ – vector of the cost of each column
T– the tabu list
R – remaining columns
S – the current solution
S*– the best solution

Algorithm 3: Tabu Heuristic for the k-Clique Covering
Input: [a(i,j)], [c(j)]
Output: the cover S*
1. T=∅; S*=∅;
2. while not end condition
2.1. R=[a(i,j)]\T; S=∅;
2.2. while R ≠ ∅ do
2.2.1. choose the best line i*∈R such as |a(i*,j)|=min |a(i,j)| ∀j;
2.2.2. choose the best column j* that covers line i*;
2.2.3. update R,S,T: R=R\a(i,j*) ∀i; S=S∪{j*}; update T;
2.3. end while;
2.4. sort the cover S by descending order of costs;
2.5. for each Si do if (S\Si is still a cover) then S=S\Si;
2.6. if (cost(S)< cost(S*)) S*=S;
3. end while;
4. return S*;

Each iteration of the inner cycle of the heuristic chooses a line to be covered. The best column which covers the line, updates solution S and the remaining R columns. The chosen line is usually the line that is more difficult to cover, i.e. the line which corresponds to fewer columns. After reaching the cover set, the second step is to remove redundancy, by sorting the cover in descending order of cost and checking if each k-clique is really essential. The outer cycle improves the constructive heuristic using a Tabu Search strategy that removes some already used columns and re-builds a new solution.

3.2 Numeric Example

In this sub-section the running example, initialized in section 1, is presented, in order to show the condensed graph and the brokers.

Following the rule of three degrees of influence, i.e., our friends' friends' friends affect us, proposed by Christakis and Fowler [9], we are going to use the 3-cliques, equivalent to a power graph G^3. In Figure 3, three 3-cliques partially cover the given graph. We can verify that in each k-clique the maximum distance between two nodes is equal to 3, the value of *k*.

In Figure 4, on the left, matrix [ai,j] shows that nodes 2,10,12 and 14 are over-covered and node 23 is an isolated node. Also in Figure 4, on the right, the condensed graph is composed by three communities: A, B and C.

In the condensed graph, the communities are represented by squares, and the brokers, represented by dots.

Table 1 shows the data extracted from the figures:

- the cost of the solution is equal to the number of columns, and is also equal to the number of communities in the social network, i.e., equal 3;
- in the solution, 22 covered nodes can be found;
- the over-covered nodes are 4 and they represent the brokers in the network;
- one node is isolated, which is a node in the periphery of the social network;

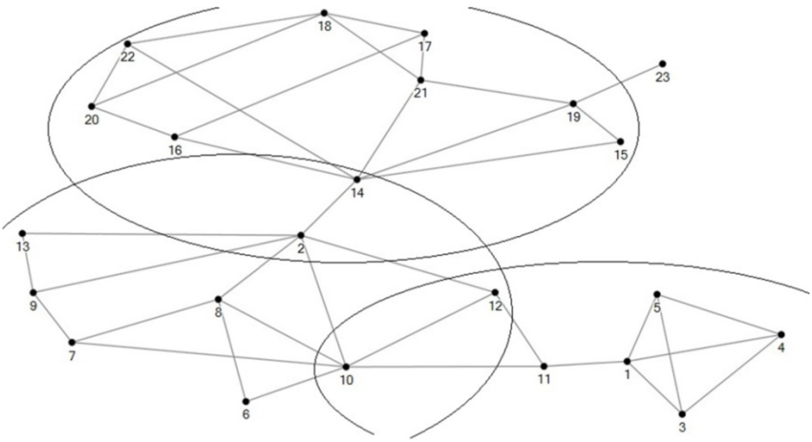

Fig. 3. Three k-cliques, with k=3, covers the graph partially

	A	B	C	isolated
1			1	
2	1	1		
3			1	
4			1	
5			1	
6		1		
7		1		
8		1		
9		1		
10		1	1	
11			1	
12		1	1	
13		1		
14	1	1		
15	1			
16	1			
17	1			
18	1			
19	1			
20	1			
21	1			
22	1			
23				1

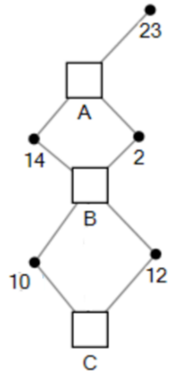

Fig. 4. The problem solution and the respective condensed graph G_C

Table 1. Information Extracted from Figures 3 and 4

|x| = number communities	covered nodes	over-covered = brokers	isolated nodes
3 (A,B,C)	22	4	1

In this paper, the network is shrunk into a condensed network, where each community is reduced to a single node and the over-covered nodes are called brokers, as shown in Figure 4.

4 Computational Results

To implement the computational results of this algorithm some choices such as the computational environment, the datasets and the performance measures 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 AMD 1.90 GHz processor with 8.00 GB of main memory running under the Windows 7 Home Premium operating system.

To validate the proposed method, two groups of datasets were used, the Erdös graphs and some cliques from the DIMACS [10] benchmark instances. In the Erdös graphs, 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 second group of graphs contains some clique instances from the second DIMACS challenge. These include the "brock" graphs, which contain cliques "hidden" within much smaller cliques, increasing the difficulty of discovering cliques in these graphs. The "c-fat" graphs are a result of fault diagnosis data.

The performance measures taken into account are the computational time and the quality of the solution, namely the quality of the visualization of the condensed network.

4.1 Computational Time

We selected 4 brock datasets, 3 c-fat datasets and 3 erdos datasets. For each dataset we tested from k=1 to k=7. For large *k* values, only one community was found and there were no brokers. In table 2, the chosen instances with the number of nodes, the diameter and the computational time are shown.

Although both algorithms are NP-hard, the less than 120 seconds computational time seems acceptable. The polynomial time complexity of the heuristics and the reduction of the optimization parameters insure the presented results.

Table 2. Datasets and Run Time for Kmax=7

Graph	nr nodes	diameter	time (seconds)
brock200-1	200	2	18
brock200-2	200	2	19
brock400-1	400	2	61
brock400-2	400	2	60
c-fat200-1	200	18	14
c-fat200-2	200	9	15
c-fat500-1	500	40	29
erdos-97-1	472	6	71
erdos-98-1	485	7	65
erdos-99-1	492	7	70

4.2 Quality of the Solutions

To test the visualization of the dataset erdos-97-1dataset was chosen. In Table 3, the k variable, number of communities |x|, the total number of covered nodes and the isolated nodes can be found.

Table 3. Solutions for erdos-97-1

k	|x| = number communities	covered	over-covered = brokers	isolated nodes
k1	49	204	27	90
k2	12	170	26	137
k3	4	164	20	143
k4	3	277	0	30
k5	1	305	0	2

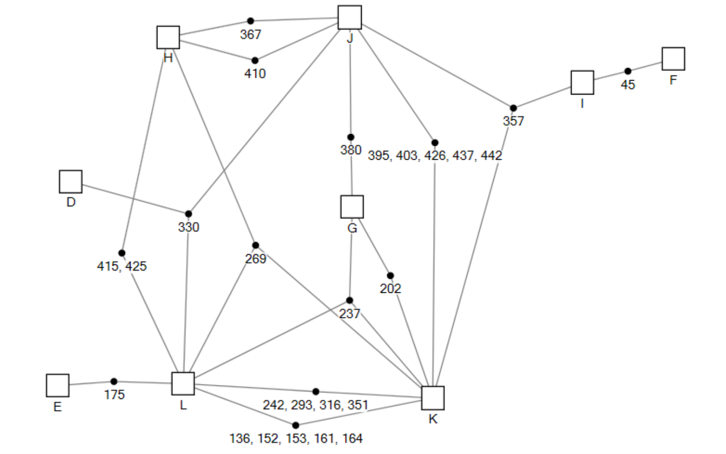

Fig. 5. Condensed erdos-97-1-k2 graph

Solution k2 was chosen with 12 communities allowing a good visualization of the network. In Figure 5, the condensed erdos-97-1-k2 graph is presented, where, the bridges were purposely omitted to simplify the visualization. Only 9 of the 12 communities are shown, the other three communities are connected by bridges.

The purpose of this experiment is not to highlight the exact solution, but rather to provide a good visualization of the network. The shrinking of the communities allows a general visualization of the network, the identification of the dense groups and the finding of the brokers. The analysis of the condensed networks also allows discovering the structural holes between communities.

5 Conclusions

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. The data extracted from social networking goes beyond the structure of communities, allowing the finding of the brokers that interact between groups. In this paper the finding of communities and the identification of their related brokers is shown.

The community partition can be relaxed for the set covering problem allowing brokers, i.e. over-covered actors. With this purpose in mind we defined community as a k-clique and the community partition as a set covering problem with k-cliques allowing over-covered and existence of isolated nodes.

The algorithm returns a condensed graph, which allows a new visualization of large networks. The communities and brokers in a condensed graph undoubtedly simplify the work of the social network analyst. The proposed visualization of the condensed network not only clearly shows the brokers, but also allows the social network analyst to detect structural holes, in order to enhance the hidden structures of the network.

Acknowledgement. The first author would like to thank the FCT support in the Funding of Strategic Projects with Public Interest promoted by Associated Laboratories and RD Units, PEst-OE/EEI/UI0434/2011.

References

1. Auber, D., Chiricota, Y., Jourdan, F., Melançon, G.: Multiscale visualization of small world networks. In: Proceedings of the Ninth annual IEEE conference on Information visualization, INFOVIS 2003, Washington, DC, USA, pp. 75–81 (2003)
2. Burt, R.S.: Structural Holes: The Social Structure of Competition. Harvard University Press (1992)
3. Cavique, L., Mendes, A.B., Santos, J.M.A.: An Algorithm to Discover the k-Clique Cover in Networks. In: Lopes, L.S., Lau, N., Mariano, P., Rocha, L.M. (eds.) EPIA 2009. LNCS, vol. 5816, pp. 363–373. Springer, Heidelberg (2009)
4. Cavique, L., Luz, C.J.: A heuristic for the stability number of a graph based on convex quadratic programming and tabu search. Journal of Mathematical Sciences **161**(6), 944–955 (2009)
5. Cavique, L., Rego, C., Themido, I.: Subgraph Ejection Chains and Tabu Search for the Crew Scheduling Problem. Journal of Operational Research Society **50**(6), 608–616 (1999)
6. Cavique, L., Rego, C., Themido, I.: A scatter search algorithm for the maximum clique problem. In: Ribeiro, C., Hansen, P. (eds.) Essays and Surveys in Meta-Heuristics, pp. 227–244. Kluwer Academic Pubs., Dordrecht (2002)
7. Chakrabarti, D., Faloutsos, C.: Graph mining: Laws, generators, and algorithms. ACM Computing Surveys **38**(1), 1–69 (2006)
8. Chvatal, V.: A greedy heuristic for the set-covering problem. Mathematics of Operations Research **4**, 233–235 (1979)
9. Christakis, N., Fowler, J.: Connected: The surprising power of networks and how they shape our lives, Back Bay Books/Little, Brown and Company. Hachette Book Group (2011)
10. DIMACS: Maximum clique, graph coloring, and satisfiability. Second DIMACS implementation challenge (1995), http://dimacs.rutgers.edu/Challenges/
11. Derenyi, I., Palla, G., Vicsek, T.: Clique Percolation in Random Networks. Physical Review Letters **94**(16), 160202 (2005)
12. Du, N., Faloutsos, C., Wang, B., Akoglu, L.: Large human communication networks: patterns and a utility-driven generator. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 269–278 (2009)
13. Dwyer, T., Riche, N.H., Marriott, K., Mears, C.: Edge Compression Techniques for Visualization of Dense Directed Graphs. IEEE Transactions on Visualization and Computer Graphics **19**, 2596–2605 (2013)
14. Easley, D., Kleinberg, J.: Networks, Crowds and Markets: Reasoning About a Highly Connected World. Cambridge University Press (2010)
15. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA **99**(12), 7821–7826 (2002)
16. Glover, F., Laguna, M.: Tabu Search. Kluwer Academic Publishers, Boston (1997)
17. Granovetter, M.: The strength of weak ties. American Journal of Sociology **78**, 1360–1380 (1973)

18. Johnson, D.S.: Approximation algorithms for combinatorial problems. Journal of Computer and Systems Sciences **9**(9), 256–278 (1974)
19. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. Bell Systems Technical Journal **49**, 291–307 (1970)
20. Luce, R.D.: Connectivity and generalized cliques in sociometric group structure. Psychometrika **15**, 159–190 (1950)
21. Nooy, W., Mrvar, A., Batagelj, V.: Exploratory Social Network Analysis with Pajek. Cambridge University Press (2005)
22. Royer, L., Reimann, M., Andreopoulos, B., Schroeder, M.: Unraveling Protein Networks with Power Graph Analysis, in Berg, Johannes. PLoS Computational Biology **4**(7), e1000108 (2008), doi:10.1371/journal.pcbi.1000108
23. Scott, J.: Social Network Analysis: A Handbook. SAGE Publications Ltd. (2000)
24. Soriano, P., Gendreau, M.: Tabu search algorithms for the maximum clique. In: Johnson, D.S., Trick, M.A. (eds.) Clique, Coloring and Satisfiability, Second Implementation Challenge DIMACS, pp. 221–242. American Mathematical Society (1996)
25. Tarawneh, R.M., Keller, P., Ebert, A.: A General Introduction To Graph Visualization Techniques. In: Garth, C., Middel, A., Hagen, H. (eds.) Proceedings of IRTG 1131, Visualization of Large and Unstructured Data Sets Workshop, pp. 151–164 (2011)
26. Yang, J., Leskovec, J.: Overlapping Community Detection at Scale: A Nonnegative Matrix Factorization. In: ACM International Conference on Web Search and Data Mining (WSDM), pp. 587–596 (2013)