Identifying Bridges for Information Spread Control in Social Networks

Michał Wojtasiewicz(✉) and Krzysztof Ciesielski

Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
m.wojtasiewicz@phd.ipipan.waw.pl, k.ciesielski@ipipan.waw.pl

Abstract. In this paper scalable method for cluster analysis based on random walks is presented. The main aim of the algorithm introduced in this paper is to detect dense subgraphs. Provided method has additional feature. It identifies groups of vertices which are responsible for information spreading among found clusters. The algorithm is sensitive to vertices assignment uncertainty. It distinguishes groups of nodes which form sparse clusters. These groups are mostly located in places crucial for information spreading so one can control signal propagation between separated dense subgraphs by using algorithm provided in this work.

Keywords: Information spread · Clusters · Scalable · Signal transfer · Random walks

1 Introduction

One of the most general occurences in the world is formation of structures of elements connected with different relations. These are called networks. Knowledge about subsets that contain information about the network has high potential of use in data mining. The most desired subsets to find in networks are called clusters. Dense subsets can be interpreted in many ways. This creates a necessity for algorithms that can cope with diversity of possible meanings. Commonness of networks in everyday life (e.g. the Internet, data sets of citings) implies using advanced methods to analyze them. The most common and natural coding method for networks are graphs. Graph structure and the way of information spread in networks are the most interesting fields of research in social network community detection. In this paper scalable method of cluster analysis based on random walks is presented. The method divides a graph into subsets, where some of them can be used for information spread control. The main aim of the algorithm presented in this paper is to detect dense subgraphs. The method provides clustering sensitive to vertices assignment uncertainty. As a result of a introduced Locally Aggregated Random Walks (LARW) algorithm one receives division which distinguishes groups of nodes responsible for signal transfer between clusters.

L.M. Aiello and D. McFarland (Eds.): SocInfo 2014 Workshops, LNCS 8852, pp. 390–401, 2015.
DOI: 10.1007/978-3-319-15168-7_48

2 Related Works

So far many algorithms for detecting communities in networks has been developed. From most popular and most frequent used techniques one has to distinguish four categories. Because of the diversity of cluster analysis problems, each of the areas is used in different situations. Choice of a method of identifying clusters should be made so that the available knowledge about the data could be used the most effectively. These are methods from categories: *bisection methods*, *hierarchical methods*, *combinatorical methods* and *spectral methods* [3]. In practical questions one mostly deals with large graphs, which frequently consist of houndreds of thousands nodes and millions of edges. In such situations there is a limited number of methods which provide a solution in a short time. This is because of complexity problems and difficulty of finding dense sets in large networks. Initial analysis, e.g. estimation of expected number of dense sets, is hard to perform as well. These are the reasons why hierarchical methods are most preffered to use in such situations. The most efficient algorithms operate on smaller sets and then agregate results with a determined stop condition. [7][6]. In this paper authors introduced a hierarchical, scalable algorithm of cluster analysis. This algorithm returns a very special division. Among standard clusters one can distinguish subgraphs which are sparse and cannot be assigned to any dense clusters. These special subgraphs enable control of signal propagation in between clusters. This subject is connected to feature of MCL algorithm and it was fully discussed in section 3.3.

Many of articles speaking about modeling or controlling the information spread in networks focus on greedy selection of vertices that have the highest influence in graph [5]. The main problem with this approach is that user starts with one most influential vertex and then greedy algorithm searches for most influential node in given neighbourhood. It can be easily seen that this kind of thinking produces very local result. Additionaly it is very probable that first most influential vertex is deep in cluster. Finding few most influential nodes in social networks in that way do not solve problem of signal propagation between clusters. An occurence similar to the feature connected to the MCL algorithm (section 3.3) was noticed in paper [2]. The author of [2] paper noticed that vertices of high degree gather more information in their neighborhood, while vertices of lower degree quickly transfer information inside the graph. It was noticed, that in dense subsets information are transfered relatively fast. It happens because such subsets have many internal edges and fewer on the outside. That creates the problem of communication between the clusters, which should be solved by initiating signal transfer on the boundaries of clusters. This is what LARW does. Interesting approach for identyfing influential veritices was presented in [1]. Authors analyzed dynamic social networks and they developed algorithm which assigns *dynamic influential value*. This coefficient is based on probability of spreading influence through time. It is calculated in a greedy way so there is again problem with local optimum. Because it takes into account information from future states of network it is useless in static case analysis. In work [4] authors introduced approach in which there can be more than one type of influence. Every node can have a *opinion* which is continous function of time. Despite that interesting approach authors assume that influence of node is given by its degree. This is not so simple. It is easy to imagine that vertex can have small degree but signal started in this vertex will propagate very fast. This will happen in situation when that node is connected to several dense clusters.

In every work mentioned above vertices were considered singly. Introduced in this paper algorithm provides division in graph where some groups of vertices can be used to signal diffusion control.

3 LARW Algorithm

3.1 Motivation

Popular way of dealing with a complex problem is to divide it into smaller parts. The point of this process is to minimize the complexity without losing key data. One has to find optimal trade-off between global and local approach. Algorithm presented in this paper is an answer to a problem of scalability of MCL method [10]. That algorithm relies on simulating random walks on network. This procedure comes down to multiplication of stochastic matrices. There is a computational problem related. Because of multiplying very large matrices one has to have huge amount of operation memory and computational power. At the beginning of the process stochastic matrix is sparse but it becomes dense after several steps. As the matrix gets more and more dense the operation memory starts to become insufficient. It regards even small graphs. Solution suggested by the authors is based on execution computations on specific subsets of graph. Dense subsets are seperated by using the MCL algorithm locally and then aggregating results. This is a hierarchical method which gives in result multilevel clustering. That division has an important advantage. Among selected clusters there are subgraphs which are not dense in a sense of internal edges. Authors have named these sparse subgraphs *bridges* and defined as follows:

Bridges are subgraphs which have less internal edges than external ones. Additionaly they have at least two neighbouring clusters and at least two of those clusters are dense.

This definifion implies that bridge can be connected to more than just two clusters and several bridges can be connected to each other. Simulations in section 4 show that the role of these bridges is transfering signal/information between clusters.

3.2 Scheme

In this section authors introduced a scheme of proposed algorithm. The scheme consists of three main steps which were discussed briefly below and can be seen on figure 1.

1. Find spanning tree $T(G)$ of a given graph G. Now find vertex v which fulfills condition:
$$V(T(G))_{min} = \underset{u \in V(T(G))}{\operatorname{argmin}} \ (deg(u)) \tag{1}$$
$$v = \underset{w \in G}{\operatorname{argmax}}(deg(w) : w \in V(T(G))_{min}) \tag{2}$$
where $V(T(G))$ is set of all vertices in graph G and $deg(v)$ is a degree of vertex v. Next, cut out neighbourhood of rank r of found vertex v. Save the rest of a graph as G'. Repeat this step for all next G' until reaching situation when all nodes are assigned to some neighbourhood. This first division will be called *initial clustering.*

2. Apply MCL method for every cluster in initial clustering. Save received results.

3. Aggregate every cluster from second step to one *supernode.* Create a new graph from supernodes and assign transition probabilities between them as a sum of probabilities between vertices from given clusters.

Whole procedure have to be repeated until graph becomes a separated set of supernodes.

First step of the scheme above contains an important rule for choosing vertices. This rule should cause a situation where vertices chosen firstly are located near borders of clusters. Neighbourhood of that vertex probably consists of vertices from different actual clusters. MCL algorithm should find out that certain initial cluster has to be divided according to borders of actual dense subsets.

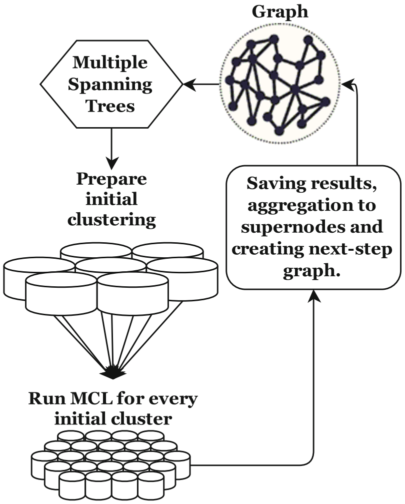
Graph
Multiple Spanning Trees
Prepare initial clustering
Saving results, aggregation to supernodes and creating next-step graph.
Run MCL for every initial cluster

Fig. 1. Flowchart of LARW algorithm

Aggregating results of clustering by grouping vertices to a supernode is a typical technique of hierarchical algorithms.

The main idea of algorithm is to recognize where in the graph are located borders and then dividing initial clusters along them. Local approach satisfies requirement of scalability of algorithm for large datasets. Hierarchical way ensures that vertices near to a border which are from different clusters will be still separated.

Scalability of algorithm is really good. LARW performs tens or hundreds times faster than MCL [10] for large sets and that advantage becomes higher with larger graphs.

3.3 MCL Feature

During work on the scalable modification of Markov Clustering Algorithm very interesting feature was revealed. Figure 2 shows behaviour of the algorithm in certain situation of three vertices.

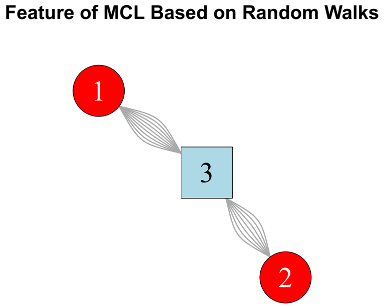

Fig. 2. MCL feature

In the figure 2 result of running MCL can be seen. Despite that there is no distinction between these three nodes, the method has found two clusters. It happened because probability mass run out very fast from vertex number *3* to other vertices. This is why MCL method decided to mark vertex *3* as a different one in a sense of probability mass distribution. Role of a node with number *3* is just to transfer random walker between vertices *1* and *2*. It is not hard to imagine that nodes from figure 2 can be groups of vertices. If one of groups hidden underneath vertex *3* form a sparse cluster and its neighbouring clusters are dense then vertex *3* is a bridge according to a definition from section 3.1. As can be seen in section 4 bridges play an important role in information spread through networks. If one wants to reach as many units as possible in shortest time then it is not recommended to start in a node which is deep in the cluster. In a situation like that signal will need a lot of steps until it travels to a different cluster. The best way is to identify bridges (if they exist) and initate signal in one or some of them. When it is wanted to target nodes only from one cluster identifying bridges will be helpful. Removing bridges adjacent to a considered cluster will make leaving that cluster more difficult. One can control signal propagation on a graph by opening and closing flow through bridges.

4 Simulations and Results

In this section results of several simulations were provided. Authors considered two directions of checking role of bridges in graphs. First direction is to compare pace of signal spreading in two situations: initialized in bridges and initialized in a cluster. Second is to analyze how important role bridges play in transferring information between neighbouring clusters. Both directions were presented in sections 4.2 and 4.3 respectively. It was difficult to make a comparison with presented LARW algorithm. This is because most of hierarchical clustering algorithms do not find any subsets which fulfill definition of bridge. Authors found one algorithm - Walktrap [6] which can find at least one bridge. Comparison of clusterings was placed in section 4.

LARW algorithm gives in a result multilevel clustering therefore for analysis of signal propagation division with highest modularity [8] was taken.

4.1 Datasets

In this section a basic statistics of chosen graphs was provided. In table 1 one can see parameters for degrees of nodes in given graphs. All of these datasets can be found on [9]. As can be seen in table 1 LARW algorithm found couple bridges. In section 4 one can see that despite of the fact there is little number of bridges they play crucial role in signal transferring.

Table 1. Statistics of analyzed graphs

Graphs	#V(G)	#E(G)	Minimum	Median	Maximum	Bridges found by LARW
Coauthorship	16264	47594	1	4	107	40
Zachary	34	75	1	3	16	1
Dolphins	62	159	1	5	12	5
Lesmis	77	254	1	6	36	1
Football	115	615	7	11	12	1
Polblog	1490	16726	0	7	351	0

4.2 Signal Initialization

First way of analyzing bridges influence is to simulate how fast signal discovers a graph when it was started in a bridge against one initialized in a cluster. To do that the Markov Chain was involved again. For every bridge authors did the same procedure:

1. Identify bridge and remove subgraph induced by vertices from considered bridge and adjacent clusters. Call it G_{sub}. Set of neighbouring clusters can contain other bridges as well.
2. Simulate signal propagation by multiplying stochastic matrices 1,2,. . .,d times where d is diameter of G_{sub}. For every cluster G_{sub} in every step calculate how many vertices were visited outside given cluster in certain number of steps.

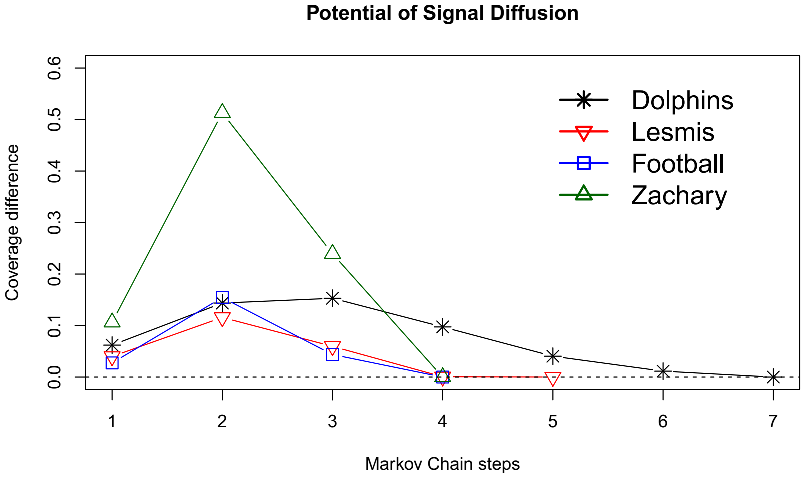

Fig. 3. Signal Initialization by LARW

Step 2. can be done by calculating fraction of positive transition probabilities from given cluster to the rest of a graph. Now it is enough to compare fractions derived from considered bridge and other clusters. For that purpose authors calculated average fraction for all clusters except bridge. Then for every G_{sub} difference between bridge fraction and average fraction from other clusters was derived. In result one recives a list of differences between visited fraction of nodes in certain number of steps. Of course number of steps as well as diameter can be different in different G_{sub}. At the end authors calculated average difference between considered fractions. Average was taken over all G_{sub}'s for every number of steps separately. In result one receives mean coverage of signal spread in graph in two situations: starting in a bridge cluster and starting in any other. Figure 3 shows results for different datasets.

Figure 4 shows comparison between information spreads induced by bridges detected by LARW and bridges detected by Walktrap algorithm.

Figure 3 proves that by initializing signal in a bridge, one will achieve higher coverage of a network than initializing it in any other cluster. All coverage differences are positive which means that signal recovers graph faster when it was started in one of found bridges. In figure 4 one can see that bridges found by Walktrap are very different from those found by LARW. As can be seen in a

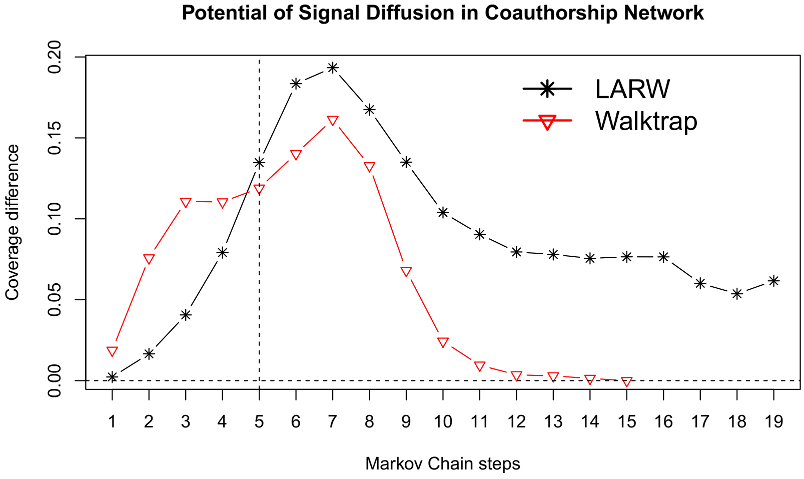

Fig. 4. Signal Initialization in Coautorship network

figure 4 Walktrap bridges are in fact parts of clusters. This is why signal is spreading very fast in first five steps and then it stucks while signal from LARW bridges recovers more and more nodes. This situation implies that Walktrap bridges are less influential after several steps of random walker.

In figure 5 one can see result of simulations when LARW did not find any bridges but Walktrap found four. These bridges are mistakes. Signal spreads faster initialized in cluster than in one of these bridges.

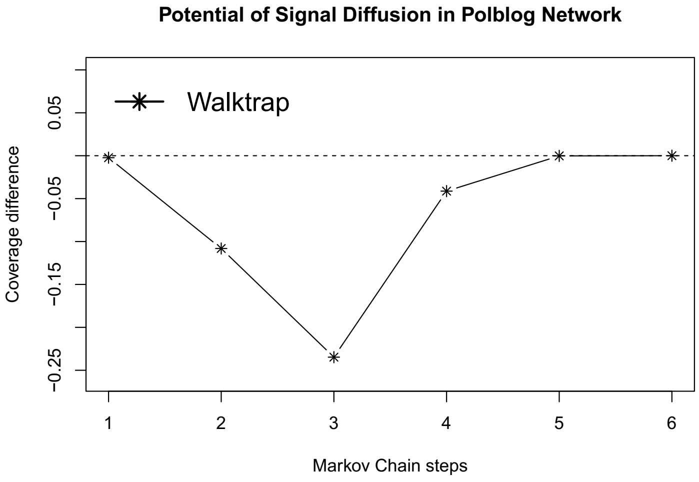

Fig. 5. Signal Initialization in Polblog network by Walktrap

4.3 Information Spread Control

Analysis of how well one can control travelling between clusters can be done by manipulating signal flow through bridges. Procedure is very similiar to one used in 4.2. The distinction is that authors computed difference in fractions of visited nodes in a clustering with bridges and without them. Potential of information flow between dense clusters through bridges in G_{sub} was calculated in 4.2. The same way of thinking was performed here but authors considered subgraph of G_{sub}. That subgraph does not have analyzed bridges. So this is a situation in which information cannot travel through a bridge. At the end average difference between fractions of visited nodes with using bridges and without them was calculated. Figure 6 shows results for several datasets. In figure 7 comparison between information spreads induced by bridges detected by LARW and Walktrap algorithm was shown.

In figure 6 one can see difference between coverage achieved with bridges and coverage without them. Positive values provide that removing bridges is a method that impede signal dispersion. When comparing figures 3 and 6 one can see that without bridges even number of steps needed to uncover the whole graph is larger. Clearly bridges are located between clusters and they transfer large amount of information.

In figure 7 comparison of quality between bridges found by Walktrap and bridges found by LARW has been shown. One can see that coverage given by LARW algorithm in step number five exceeds the one achieved by bridges from Walktrap clustering. The difference between signal spreads with and without bridges is even negative. This means that spreading information is easier without bridges found by Walktrap. This is because some of them are connected stronger to one of neighbouring clusters and should be part of them. After removing

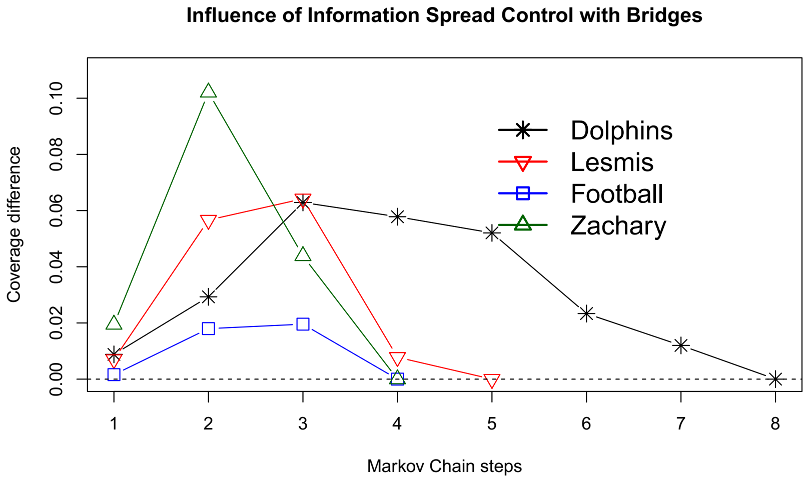

Fig. 6. Bridges Influence by LARW

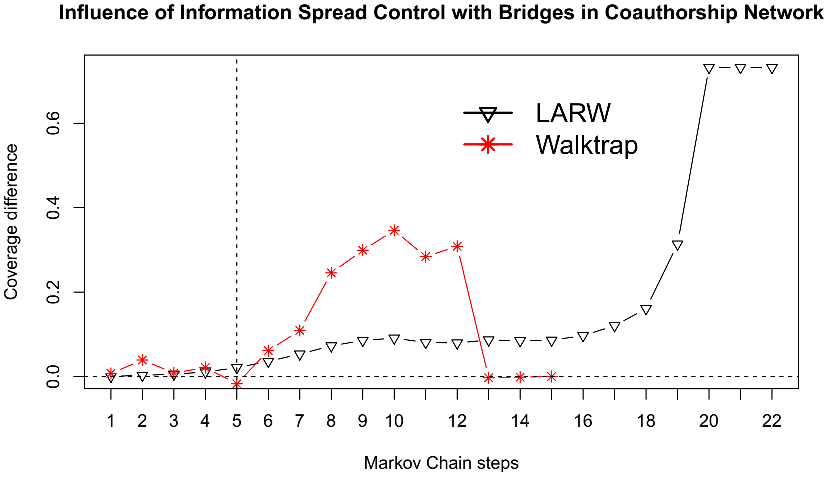

Fig. 7. Bridges Influence in Coautorship network

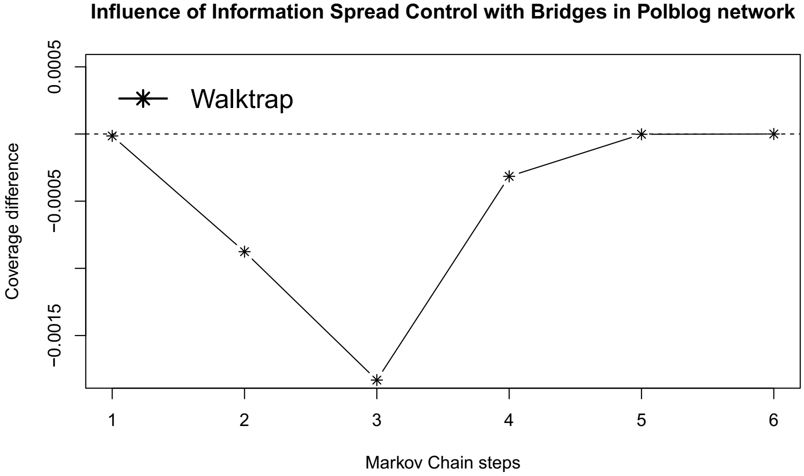

Fig. 8. Bridges Influence in Polblog network by Walktrap

bridges nodes there is simply less vertices to visit when information propagate through network.

Interesting thing is that the diameter of G_{sub}'s in case of LARW clustering becomes longer after removing bridges. It can be seen in figure 7. It means that some of found bridges are located in the most important for signal dispersion places in a graph. Influence of found bridges is so big that even after a number of steps equal to diameter of analyzed G_{sub} the difference in coverages is very high. The monotonic behaviour of coverage difference with respect to the Markov Chain steps is a consequence of splitting G_{sub} into two separated subgraphs. Clearly situation in which after a large number of steps difference did not converge to zero implies that signal started in G_{sub} without bridges stucks in one of clusters and cannot recover the rest of a graph.

In figure 8 negative influence of a signal spread is visible. After removing bridges found by Walktrap algorithm signal spreads faster than with them. This is because they are strongly connected to one of clusters. When signal starts to spread from a cluster, information has to get to this bridge which is weakly connected to other clusters. Identyfing these bridges gives nothing because in fact they are a part of a certain cluster.

5 Conclusion and Future Work

In this section several conclusions were provided. Firstly, one can easily see that the most difficult part of signal propagation is spreading information between clusters. According to definition of a cluster which is a dense subgraph one can expect to observe fast signal diffusion inside the cluster. Large number of edges forming cluster ensures that most of vertices in cluster will be reached in several steps. This is why random walker will rather stay in cluster than travel between clusters. Presented LARW algorithm provides clustering which has important probabilistic feature. It can separate groups of vertices which form a sparse cluster but should not be included in any of dense ones. Authors found out those groups are bridges defined in section 3.1 and they are responsible for transferring information between clusters. Simulations presented in section 4 shows that bridges are very important as a neighbours of dense clusters. Without them pace of signal dispersion in a graph becomes slower. Futhermore one can control spreading of information by removing certain bridges or lowering probability of passing information through them.

Clustering provided by LARW enables control of information diffusion in social networks by identyfing subgraphs crucial for transferring signal between clusters.

Algorithm presented in this work shows how much is still to be done in control of signal propagation by community detection. One of topics is a situation when LARW cannot find any bridge. Where is an optimal place for signal initialization then? One of possible solutions is to find groups of vertices from every clusters that are on borders of them. Then one has to remove number of vertices without quality loss of signal dispersion. This topic will be a part of future research. Second interesting subject is to identify bridges connected to other bridges. It is better to manipulate signal with small number of influential subgraphs. This have to be done carefully because one can easily lower resolution of influence and ability to control signal diffusion. Another important direction of research is a situation when LARW finds lots of bridges. Reduction of influence removing part of them can be huge so one has to examine importance of every of those subgraphs and choose best ones for given problem.

Acknowledgments. The paper is co-founded by the European Union from resources of the European Social Fund. Project PO KL ,,Information technologies: Research and their interdisciplinary applications”, Agreement UDA-POKL.04.01.01-00-051/10-00.

References

1. Aggarwal, C., Lin, S., Yu, P.S.: On influential node discovery in dynamic social networks. In: SDM, pp. 636–647 (2014)
2. Doerr, B., Fouz, M., Friedrich, T.: Why rumors spread fast in social networks. Communications of the ACM **55**(6), 70–75 (2012)
3. Fortunato, S.: Community detection in graphs. Complex Networks and Systems Lagrange Laboratory ISI Foundation (2010)
4. Ju, C., Cao, J., Zhang, W., Ji, M.: Influential node control strategy for opinion evolution on social networks. Abstract and Applied Analysis, Article ID 689495 (2013)
5. Kempe, D., Kleinberg, J., Tardos, É.: Influential nodes in a diffusion model for social networks. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1127–1138. Springer, Heidelberg (2005)
6. Pons, P., Latapy, M.: Computing communities in large networks using random walks (2005)
7. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. P10008 (2008)
8. Newman, M.E.J.: Modularity and community structure in networks. PNAS **103**(23), 8577–8582 (2006)
9. Newman, M.: http://www-personal.umich.edu/mejn/netdata/
10. van Dongen, S.M.: Graph clustering by flow simulation. PhD thesis, Universiteit Utrecht (2000)