# CoViFlowPro: A Community Visualization method based on a Flow Propagation Algorithm

Costas Panagiotakis
Dept. of Business Administration
TEI of Crete
72100 Agios Nikolaos, Greece
cpanag@staff.teicrete.gr

Harris Papadakis
Dept. of Informatics Engineering
TEI of Crete
71004 Heraklion, Greece
adanar@ie.teicrete.gr

Paraskevi Fragopoulou*
Dept. of Informatics Engineering
TEI of Crete
71004 Heraklion, Greece
fragopou@ics.forth.gr

## ABSTRACT

We propose a method (CoViFlowPro) for the visualization of a community of a node based on the results of a flow propagation algorithm (FlowPro) [15]. FlowPro computes the community of a node in a network without the knowledge of the structure of the entire graph resulting at the same time to a metric that is related with the probability of a node belonging to the requested community. In this work, we use this metric to visualize the community of a node on the curve of the Archimedean spiral. The novelty of CoViFlowPro is the fact that the proposed community visualization method is local and it does not require the knowledge of the entire graph as most of the existing visualization methods from the literature. Moreover, it visualizes the community of a node taking into account the significance of the node membership.

## Categories and Subject Descriptors

Information Systems Applications [**Web applications**]: Social networks

## Keywords

Community visualization, complex networks, community detection, belief propagation

## 1. INTRODUCTION

Social networks in various application domains present an internal structure, where nodes form groups of tightly connected components which are more loosely connected to the rest of the network. These components are mostly known as *communities*. Various applications like finding web communities, detecting/analyzing/visualizing the structure of social networks are just some for which community detection is important [18]. Various methods exist to identify communities: Some algorithms follow an iterative approach starting by characterizing either the entire network, or each individual node as community, and splitting [12] or merging [9] communities respectively, producing a hierarchical tree of nested communities, called *dendrogram*. Several researchers aim to find the entire hierarchical community dendrogram [12] while others wish to identify only the optimal community partition [4]. More recently used approaches aim to identify the community surrounding one or more seed nodes [19]. Some researchers try to discover distinct (non-overlapping) communities, while others allow for overlaps between communities [10].

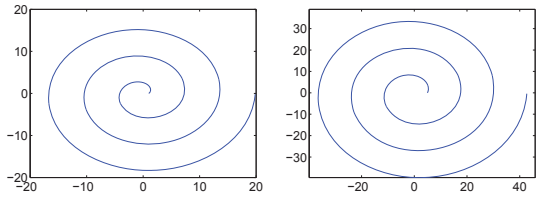
A distributed community detection algorithm (SCCD) that detects the entire community structure of a network based on interactions between neighboring nodes is proposed in [17, 18]. Extensive experiments on several benchmark graphs with known community structure and real graphs show that SCCD gives high performance results. In addition, SCCD has been used on the interactive image segmentation problem [16] outperforming other methods from the literature.

The graph visualization problem has been extensively studied in the literature and several method have been proposed to visualize large graphs [2] with applications on social and biological networks. The main goal of most of these methods is to reduce the overall edge crossings [3]. One of the most popular and widely available methods for social network visualization is the spring embedder variant of [5]. This is a force-directed algorithm in which a graph is a physical system of objects (the vertices) and springs (the edges) bind adjacent vertices together. Vertices are iteratively repositioned according to the forces of springs, as the physical system is stabilized. In [8], the authors present an edge bundling method that uses a self-organizing approach to bundling in which edges are modeled as flexible springs that can attract each other. The resulting no hierarchy-bundled graphs have clearly visible high-level edge patterns. In [7], a customized technique for identifying and visualizing community structures in social networks has been proposed that is called Vizster. Vizster presents social networks using a node-link representation, where links represent the articulated "friendship" relations between the members (nodes) of the system. The network layout is computed using a spring-embedding method, in which nodes repel each other and edges act as springs, which has the advantage of group-

*P. Fragopoulou is a collaborating researcher at the Foundation for Research and Technology-Hellas, Institute of Computer Science, 70013 Heraklion, Crete, Greece.


BICT 2014, December 01-03, Boston, United States

DOI 10.4108/icst.bict.2014.257867




(a) (b)

**Figure 1: Two examples of Archimedean spirals with (a) $a = b = 1$ and (b) $a = 5$, $b = 2$.**

ing users into identifiable communities based upon increased connectivity. Vizster also includes tools for the automatic determination of community structure using Newmanś community identification algorithm [12].

In [14], the FlowPro method is proposed that solves the single community detection problem without the knowledge of the entire graph structure. This makes possible the application of FlowPro in extremely large graphs or in cases where the graph is not given. According to the problem formulation of FlowPro, the probability of a node belonging to the requested community is analogous to its stored flow. So, the stored flow can be used as a belief (rating) of a node belonging to the community. In this work, the proposed method (CoViFlowPro) visualizes the community of a node on the Archimedean spiral [6] based on the stored flow feature. Therefore, a user of a social network for which the entire graph is unknown, is possible to use CoViFlowPro, in order to visualize his own community. The neighbors and the bridges of the initial node are depicted with different color and shape. To our knowledge CoViFlowPro is the first algorithm in the literature that visualizes a single community without the knowledge of the entire graph structure. Therefore, the main goal of this work is not to analyze the structure of the whole social network, but to provide a useful tool for a simple user of a social network, which is impossible to know the entire graph structure, to visualize his community. So, this is the main reason that the proposed visualization method uses FlowPro, since FlowPro is able to work when the entire graph structure is unknown. In addition, the Archimedean spiral can be combined with any community detection method which is able to provide a "belief" of a node belonging to the community.

## 2. METHODOLOGY

This section, presents the CoViFlowPro method. We start by briefly analyzing the FlowPro method that is firstly executed. Let $G = (V, W)$ be a graph comprising a set $V$ of nodes together with a set $W$ of edge weights. According to the problem definition of local community detection, the initial node ($s \in V$) is given and the goal is to find the set of nodes $C(s)$ that belong to the community of $s$, with $C(s) \supseteq \{s\}$. Therefore, FlowPro requires as input the initial node ($s \in V$) that searches for its community. This node has an initial flow for propagation $T(s) = |n(s)|$. More details for this method can be found in [14, 15]. The main steps of the method are described hereafter:

- In each iteration of the main process of FlowPro, the initial node propagates a flow that is shared among its neighbors.
- Each node $x$ is able to store, propagate to its neighbors, and return part of this flow to the initial node. The stored flow of node $x$ ($S(x)$) is used to visualize the community $C(s)$.
- Finally, when the algorithm converges, the flow stored at the nodes that belong to the community of the initial node is generally higher than the flow stored in the rest of the graph nodes resulting to the requested community.
- Based on the stored flow $S(x)$, FlowPro has the extra ability of removing and adding edges to $s$ in order to increase the distance for nodes $x$ that do not belong to $C(s)$ (e.g. removing bridges) and to decrease the corresponding distance for nodes $x$ that belong to $C(s)$, respectively. This property increases the converge and the performance of FlowPro.

Hereafter, a detailed description of the subsequent steps of CoViFlowPro algorithm is given. The goal of CoViFlowPro is to visualize the nodes that belong on the set $V(s) = C(s) \cup N(s)$, where $C(s)$ is the community of the node $s$ according to the FlowPro method and $N(s)$ is the set of the neighbors of $s$. Initially, the nodes of $V(s)$ are classified into the following three sets $G_1(s)$ (neighbors of node $s$ and members of $C(s)$), $G_2(s)$ (non-neighbors of node $s$ and members of $C(s)$) and $G_3(s)$ (neighbors of node $s$ and non-members of $C(s)$):

$$G_1(s) = \{u \in V(s) : u \in C(s) \cap N(s)\} \quad (1)$$

$$G_2(s) = \{u \in V(s) : u \in C(s) \cap \overline{N(s)}\} \quad (2)$$

$$G_3(s) = \{u \in V(s) : u \in \overline{C(s)} \cap N(s)\} \quad (3)$$

where $\overline{C(s)}$ is the subset of the nodes of $V(s)$ that do not belong to $C(s)$ and $\overline{N(s)}$ is the subset of the nodes of $V(s)$ that do not belong to $N(s)$. This classification is used in order to discriminate the neighbors ($G_1(s)$), non-neighboring members ($G_2(s)$) of $C(s)$ and the bridges of the initial node $s$, ($G_3(s)$), so that they are visualized differently. The nodes of $G_1(s)$ are depicted with blue circles, the nodes of $G_2(s)$ are depicted with red circles and the nodes of $G_3(s)$ are depicted with gray squares.

Subsequently, the nodes of set $V(s)$ are placed on the Archimedean spiral [6], according to their stored flow. An Archimedean spiral is characterized by a fixed arm width along its windings. The spiral function can be written in polar coordinates as follows:

$$r(u) = a + b \cdot \theta(u) \quad (4)$$

where $\theta(u)$ is computed by the stored flow $S(u)$ of node $u$. $a$ and $b$ are the two parameters of the spiral that control the minimum radius of the spiral and the distance between successive turnings, respectively. Fig. 1 depicts two Archimedean spirals with $a = b = 1$ and $a = 5$, $b = 2$. In both examples it holds that $\theta \in [0, 6\pi]$. In our experiments we have used $a = 5$ and $b = 2$. The nodes of set $V(s)$ are sorted in descending order according to their stored flow.

The nodes are also plotted with the same order. Therefore, $\theta(u)$ is computed based on $\theta(u-1)$ according to the following Equation 5.

$$\theta(u) = \theta(u-1) + 2\pi\frac{S(u-1) - S(u)}{S(1)} + \frac{\pi}{r(u-1)} \qquad (5)$$

where $\theta(1) = 0$. The quantity $2\pi\frac{S(u-1)-S(u)}{S(1)}$ is related to the position of node $u$ on the Archimedean spiral. $S(1)$ is the maximum value of the stored flow. It holds that the nodes that have high flow stored, which means that they have high probability to belong to the community, are plotted close to the origin, while the rest of the nodes and especially the bridges are plotted farther from the origin. Finally, the quantity $\frac{\pi}{r(u-1)}$ is added in order to avoid intersections between the nodes of almost equal stored flow.

## 3. EXPERIMENTAL RESULTS

The proposed method has been tested on a variety of 208 benchmark graphs with known community structure as well as on several real graphs. The benchmark graphs have been also used in community detection algorithms FlowPro [15] and SSCD [18], since the ground truth communities of these graphs are known. These benchmark graphs were generated randomly given the following set of parameters: The number of nodes $N$ of the graph, the number of communities $Comm$ of the graph, the (average) degree of nodes $degree$ and the ratio of local links (intra-community links) to node degree $local/degree$. The real graphs are obtained from [11], [1]. A demonstration of the proposed method is given in [13].

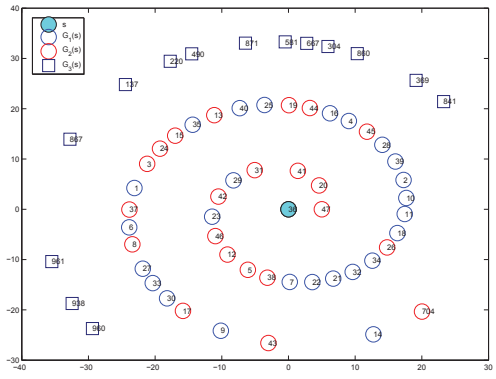


**Figure 2: An example of visualization results of CoViFlowPro method on a synthetic graph.**

Figs. 2 and 3 show three results of the CoViFlowPro on a synthetic (Fig. 2) and two real graphs (Figs. 3(a) and 3(b)). On the synthetic graph, the error between the resulting community of the FlowPro and the ground truth data is less than 3%. According to the CoViFlowPro visualization rules, the community members are depicted with blue circles if they are neighbors of s ($G_1(s)$), otherwise they are depicted with red circles ($G_2(s)$). The bridges are depicted with gray squares.

In Fig. 2, a synthetic graph with $N = 1000$, $Comm = 10$, $degree = 20$ and $local/degree = 0.65$ has been used. In this example the community size is 48 nodes. The initial node ($s = 36$) has 39 neighbors. Fourteen of them are detected as bridges. According to CoViFlowPro, only 2 out of the first 10 most important nodes of the community of $s$ are also neighbors of $s$, while the node with the highest belief to belong to $C(s)$ has label 47. Concerning the structure of $C(s)$ that can be determined by visually observing (without thresholds) the largest angle distances between successive nodes in the spiral, we can group community nodes into three categories, the *strong community nodes*, which belong to the community with very high belief: the first three nodes (47, 20 and 41), the *medium community nodes*: the next 43 nodes, and finally, the *weak community nodes*, which belong to the community with low belief: the last 4 nodes (9,43,14,704).

In Fig. 3(a), the undirected real graph ego-Facebook [11] with 747 nodes and 30.025 edges has been used. In this example the detected community has 67 nodes. The initial node ($s = 21$) has 54 neighbors. Two of them are detected as bridges. According to CoViFlowPro, it seems that five out of the first ten most important nodes of the community of $s$ are also neighbors of $s$. Concerning the structure of $C(s)$, it seems that $C(s)$ is very compact, since only the last three nodes of the community (209, 509 and 70) can be classified as weak community nodes. In Fig. 3(b), the directed real graph Wiki-Vote [1] with 7.115 nodes and 103.689 edges has been used. In this example the detected community has 52 nodes. The initial node ($s = 42$) has 44 neighbors. Two of them are detected as bridges. According to CoViFlowPro, it seems that the first eighteen most important nodes of the community of $s$ are also neighbors of $s$. Concerning the structure of $C(s)$, it seems that the community is not compact. The first twenty nodes of the community are strong community nodes, while the rest can be classified as weak community nodes.

## 4. CONCLUSIONS

We presented a community visualization algorithm which is based on a flow propagation method and on the Archimedean spiral. The stored flow of FlowPro method has been used as a belief that a node belongs to the community. The novelty of the proposed approach is the fact that CoViFlowPro is local, fully distributed, and it does not require the knowledge of the entire graph as many of the existing methods in the literature. Thus, the application of CoViFlowPro is possible in extremely large graphs or in cases where the entire graph is unknown like in most social networks presenting well the community structure and detecting bridges of the initial node.The proposed algorithm has been tested on a large number of synthetic graphs with known community structure and in graphs derived from real social networks proving its effectiveness. In our experimental results we show that the visualization of $C(s)$ is quite helpful in the analysis of the community structure. Under different community structures and sizes, the use of Archimedean spiral provides an efficient way to visualize the community and it can be also combined with any community detection method that provides a "belief" of a node belonging to the community.


## Acknowledgments

This research has been partially co-financed by the European Union (European Social Fund - ESF) and Greek na-

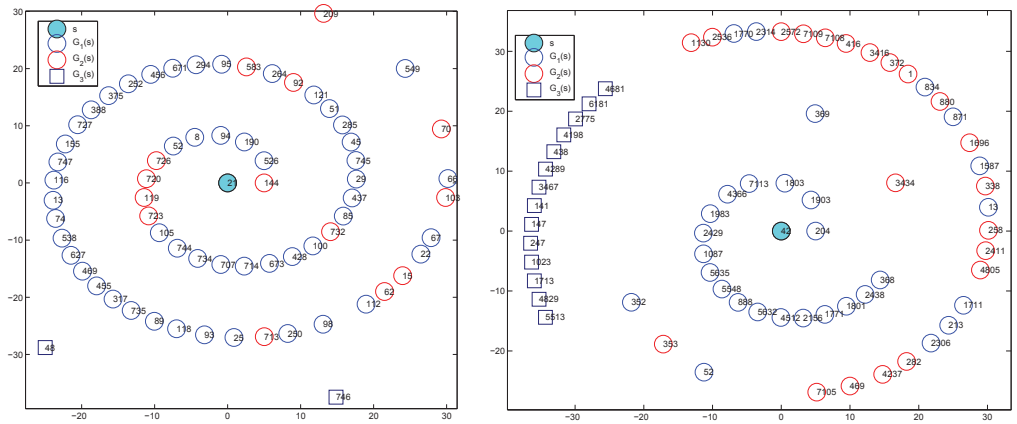



**Figure 3: Two examples of visualization results of CoViFlowPro method on real graphs.**

tional funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Programs: ARCHIMEDE III-TEI-Crete-P2PCOORD.

## 5. REFERENCES

[1] Network databases - University of Notre Dame (http://www3.nd.edu/).

[2] U. Brandes, N. Indlekofer, and M. Mader. Visualization methods for longitudinal social networks and stochastic actor-oriented modeling. *Social Networks*, 34(3):291–308, 2012.

[3] W. Cui, H. Zhou, H. Qu, P. C. Wong, and X. Li. Geometry-based edge clustering for graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1277–1284, 2008.

[4] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee. Self-organization and identification of web communities. *IEEE Computer*, 35:66–71, March 2002.

[5] T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.

[6] C. Fumeaux, D. Baumann, and R. Vahldieck. Finite-volume time-domain analysis of a cavity-backed archimedean spiral antenna. *IEEE Transactions on Antennas and Propagation*, 54(3):844–851, 2006.

[7] J. Heer and D. Boyd. Vizster: Visualizing online social networks. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 32–39. IEEE, 2005.

[8] D. Holten and J. J. Van Wijk. Force-directed edge bundling for graph visualization. In *Computer Graphics Forum*, volume 28, pages 983–990. Wiley Online Library, 2009.

[9] D. Katsaros, G. Pallis, K. Stamos, A. Vakali, A. Sidiropoulos, and Y. Manolopoulos. Cdns content outsourcing via generalized communities. *IEEE Transactions on Knowledge and Data Engineering*, 21:137–151, 2009.

[10] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015+, March 2009.

[11] J. Leskovec and R. Sosič. SNAP: A general purpose network analysis and graph mining library in C++ http://snap.stanford.edu/snap, June 2014.

[12] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, Feb 2004.

[13] C. Panagiotakis, H. Papadakis, and P. Fragopoulou. FlowPro webpage (https://sites.google.com/site/netdilab/research/flowpro) and (https://sites.google.com/site/costaspanagiotakis).

[14] C. Panagiotakis, H. Papadakis, and P. Fragopoulou. Flowpro: A flow propagation method for single community detection. In *IEEE Consumer Communications and Networking Conference*, 2014.

[15] C. Panagiotakis, H. Papadakis, and P. Fragopoulou. Local community detection via a flow propagation method. In *International World Wide Web Conference*, 2014 (under review).

[16] C. Panagiotakis, H. Papadakis, E. Grinias, N. Komodakis, P. Fragopoulou, and G. Tziritas. Interactive image segmentation based on synthetic graph coordinates. *Pattern Recognition*, 46(11):2940 – 2952, 2013.

[17] H. Papadakis, C. Panagiotakis, and P. Fragopoulou. A distributed algorithm for community detection in large graphs. In *Int. Conf. on Advances in Social Network Analysis and Mining*, 2013.

[18] H. Papadakis, C. Panagiotakis, and P. Fragopoulou. Distributed community detection in complex networks using synthetic coordinates. *Journal of Statistical Mechanics: Theory and Experiment*, 3, 2014.

[19] S. Papadopoulos, A. Skusa, A. Vakali, Y. Kompatsiaris, and N. Wagner. Bridge bounding: A local approach for efficient community discovery in complex networks. Technical Report arXiv:0902.0871, Cornell University LIbrary, Feb 2009.