Comparing Two Local Methods for Community Detection in Social Networks

Sarka Zehnalova*, Milos Kudelka Jr.†, Milos Kudelka*, Vaclav Snasel*
*VSB - Technical University of Ostrava
17. listopadu 15, 708 33
Ostrava, Czech Republic
Email:{sarka.zehnalova.st,milos.kudelka,vaclav.snasel}@vsb.cz
† Institute of Information Theory and Automation
Pod Vodarenskou vezi 4, 182 08
Praha, Czech Republic
Email:kudelka@utia.cas.cz

***Abstract*—One of the most obvious features of social networks is their community structure. Several types of methods were developed for discovering communities in the networks, either from the global perspective or based on local information only. Local methods are appropriate when working with large and dynamic networks or when real-time results are expected. In this paper we explore two such methods and compare the results obtained on the sample of a co-authorship network. We study how much may detected communities vary according to the method used for computation.**

***Keywords*-social networks, community detection, DBLP**

I. Introduction

Many systems in nature or society may be represented as networks [1]. Typical examples include World Wide Web (WWW), social networks (academic collaboration records, citation network, friendship network) or biological networks (neural networks, protein networks and food webs). Complex interactive systems like social networks are then best described by weighted networks, where the strength of the link indicates the amount of collaboration between the vertices.

There has been a considerable interest recently in the idea of communities and algorithms for their detection. The concept of community is rather intuitive, so the definition is quite vague and still a subject of debate [2], [3], [4]. Universally accepted qualitative definition states that community is a collection of vertices with dense internal connections, but sparser connections to other communities. These communities are also referred to as modules or clusters [5], [6]. Alternatively, one might define communities as the output of a community detection procedure [4], [7], which means that different techniques for detecting communities may lead to slightly different yet equally valid results [8].

In this paper, we review two different algorithms, which apply the known approaches of local community detection. The aim is to compare the results they give on the co-authorship network from the DBLP dataset[1]. For the purpose of our experiment, we have used a weighted network implemented in our Forcoa.NET[2] system [9], where also one of the algorithms is used for community detection.

The rest of this paper is organized as follows. In section II, we discuss the related work. The algorithms used in our experiment are briefly described in III. In section IV, we focus on the experiment and on its results. Section V concludes the paper.

II. Related work

A. Community detection

Recently, many algorithms for detecting communities have been proposed. Traditional techniques are graph partitioning and hierarchical clustering based on similarity measures [10], [11]. Such methods usually identify all communities in an unweighted, undirected network, assigning each vertex to one community. The knowledge of the context of a whole network and even the total number of communities is required. The fixed number and size of communities are the constraints on community detection, as well as non-overlapping communities. Various metric functions have been proposed to help solve these problems. Many community-finding algorithms are based on maximizing the quantity known as modularity [7], [12], [13], [14], but any algorithm using modularity requires complete knowledge of the entire network.

When the task is just to identify a community to which a particular vertex belongs, global algorithms are impractical. Additional motivation for local methods comes from networks that require a rather demanding generation or exploration with a crawler [11]. Also, because the knowledge of the structure of the whole network may not be available, local algorithms were considered [15].

B. Local methods

Several local methods exist [8], [16], [17], [18], [19], [20], they start the search from a random vertex, and then gradually merge neighboring vertices one-at-a-time by optimizing a measure metric. In the paper [17] they tried to avoid the

[1]http://www.informatik.uni-trier.de/ ley/db/

[2]http://www.forcoa.net

need of working with an entire social network during the community search. The details of this algorithm are presented in Section III. Clauset [18] proposed a fast agglomerative algorithm that maximizes a measure called *local modularity* in a greedy fashion. Bagrow et al. [8] proposed an alternative method to detect local communities, which consists of an l-shell spreading outward from a starting vertex. The result depends on a starting vertex and a predefined threshold l, where l is the distance from the starting vertex to all shell vertices.

III. Preliminaries

In this section we describe the common approach used in local methods for community detection and we present an overview of the two algorithms used in our experiment.

A. Problem definition

As a *network* we understand an undirected weighted graph G with N vertices and E edges. *Local network community* is a set of densely connected vertices from this network.

Local methods usually work with the terms *community core - C, community boundary - B* and *community shell - S* similarly as is shown in Fig 1 and communities are usually generated from a random starting vertex or a set of vertices (a community base). A *local community expansion* is an iterative process, in which only base vertices are considered to be a community, while the other network vertices are gradually examined and the vertices that follow certain criteria then progressively expand the community.

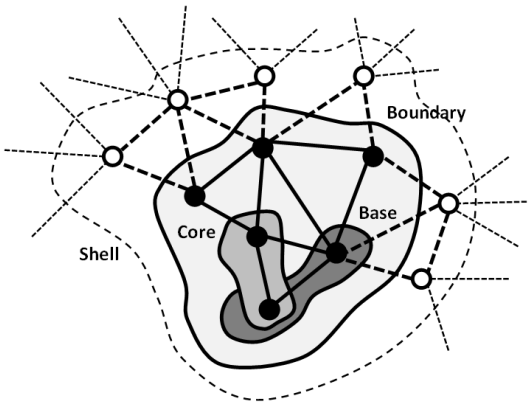

Fig. 1. Local community illustration - following terms are used: community base (a starting set of vertices), community core C, community boundary B, community shell S.

B. Local Algorithm 1 - Based on Dependency

In our previous paper [21] we presented and tested our algorithm for local community detection. This algorithm uses dependency of a vertex as a measure.

Measuring the Dependency of a Vertex

We understand dependency as a generally asymmetric measure describing a relationship between two vertices of the undirected network. Consider the situation in Figures 2a and 2b, where vertices x and y share an edge of weight 3. The vertices in the first figure have no additional neighbours. In the second figure the vertex y is adjacent to three additional vertices and the weight of the edge between them is 1. The intuition behind the term *dependency* says, that the relation between vertices in Figure 2a is balanced, while the situation in Figure 2b is different - the vertex y is less dependent on the vertex x, than vice versa. Figure 2c contains two additional edges (with weight 2) between the vertex x and two different vertices. In this situation, vertex x is no longer so highly dependent on vertex y because of the two new edges. When thinking about dependency in Figure 2c, we should also consider that the new edges include common neighbours of the vertices x and y(which mediate some part of the dependency on the vertex y).

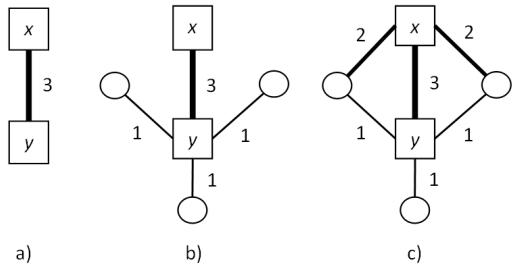

Fig. 2. Examples of dependency between two vertices.

Dependency of a Vertex on a Set of Network Vertices

The relation of dependency of one vertex on another could be generalized as a relation of dependency of vertex x on a set of n vertices $Y = \{y_1, y_2, \ldots, y_n\}$. For details see [21].

Let x not be an isolated network vertex. Then the dependency $D(x, Y)$ of vertex x on set Y of vertices is defined as:

$$D(x,y) = \frac{\sum_{y_i \in N(x,Y)} W(x,y_i) + \sum_{e_i \in \mathrm{Adj}(x,Y)} W(e_i) \cdot R(e_i)}{\sum_{e_i \in E(x)} W(e_i)} \tag{1}$$

$$R(e_i) = MAX_{y_j \in Y(v_i)} \left(\frac{W(y_j, v_i)}{W(e_i) + W(y_j, v_i)} \right). \tag{2}$$

Community Detection Based on Vertex Dependency

Every vertex in the community (excluding max. one base vertex) is dependent on the rest of community vertices.

Definition 1. *(Community base) A community base is a starting set of n vertices appropriately chosen in advance, which by definition belongs to the community and which meets the following criteria:*

1) *It is a biconnected subgraph.*
2) *At least $(n-1)$ vertices have to be dependent on the other base vertices.*

Definition 2. *(Recognition of a network vertex) An unrecognized network vertex becomes a recognized one, if during the process of local expansion, it meets the following criteria:*

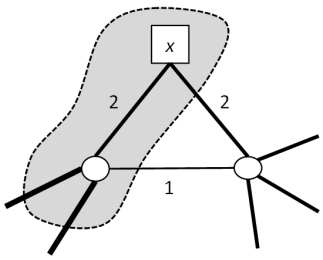

Fig. 3. Examples of dependency between two vertices.

1) *It is adjacent to at least two different community vertices.*
2) *It is dependent on the other community vertices.*

By using dependency to indicate affiliation with a community, it can occur that a community vertex can be dependent on a vertex that does not belong to the community. For example, Figure 3 shows vertex x dependent on each of its two neighbours (it shares a weight 2 edge with both of them and does not have any other edges). Thus vertex x creates a basis with each of these neighbours. If we detect a community above this basis, only the basis will constitute the community (as there is no other vertex that meets the requirements for affiliation to the community). Therefore the vertex becomes a part of two communities. However, it would be more natural if each of these two bases generated an identical community with three vertices. Therefore we establish the term *community closure*, which solves this situation.

Definition 3. *(Community Closure) Community closure is a set of community vertices, each of which qualifies:*

1) *At least one of the community vertices is dependent on it.*
2) *It is a neighbour of at least one basis or closure vertex.*

A community is further understood as a local community, including its closure.

Detection of Communities Around One Vertex

Unless it is said otherwise, a community base is considered to be two vertices connected by an edge, while at least one of the vertices is dependent on the other one. This couple should be called *the edge base*. The purpose of implementing the edge base is an effective detection of more communities.

If we want to detect all of the communities that are based on vertex a (and are dependent on it and its surroundings), then it is essential to detect the communities for all edge bases of vertex a and to remove the duplicities.

Remarks

- The two detected local communities L_1 and L_2 can have vertices in four different set relations:
  1) $L_1 = L_2$,
  2) $L_1 \cap L_2$,
  3) $L_1 \subset L_2$ resp. $L_2 \subset L_1$,
  4) $L_1 \cap L_2 = X, X \neq \emptyset, X \subset L_1, X \subset L_2$

input : social network G, start vertex n_0
input : empty community core C, empty community shell S
input : community boundary B, equal to community base
output: community L, $L = C \cap B$
Add n_0 to B, add all neighbours of n_0 to S
while *at least* 1 *vertex from* S *has been recognized* **do**
  1. Move the vertices of the boundary B which do not have neighbours outside the community L, to the core C
  2. Refill the shell S with new neighbours of the vertices added to the boundary B that are outside the community L
  3. Calculate the dependency on the other vertices for each vertex of the shell S
  4. Move to boundary B every vertex from shell S, which meets the criteria for recognition
  5. Create community closure from remaining shell vertices and refill the shell S with new neighbours of the vertices added to the closure
end
for *every vertex from* S **do**
  1. Move vertext to community closure if it meets the criteria
  2. Refill the shell S with new neighbours of the vertices added to the closure
end

Algorithm 1: Based on dependency

  Detected communities can be nested or coverlapping.
- If a starting base of a community is an edge base, the result of the detection algorithm is a biconnected subgraph.

C. Local Algorithm 2 - Iterative Local Expansion

This algorithm searches communities according to a definition of a community. The way the community is defined is the most important thing for their search. In the article [17], authors evaluate quality of the community by the sharpness of its boundary. In other words, for a group of vertices to be a community, its boundary must be somewhat sharp. To measure sharpness we first have to define what a boundary is. The boundary vertex is such a vertex that is connected by an edge with at least one vertex both from outside and from inside of the community. And all the boundary vertices constitute the boundary. Then the sharpness R of the boundary is measured as a ratio of edges that lead from the boundary into the community and edges that lead both in and out of it.

Apart from the boundary, there are two other important sets. The first one is called D (discovered) with vertices that belong to the community. The second one is S (shell) which contains all the vertices that are connected with at least one boundary vertex. Algorithm starts from a single vertex, so at the beginning the boundary and discovered sets contain only the first vertex and the shell contains all of its neighbours. In

each step, the algorithm counts hypothetical sharpness R for each vertex in shell. Then the vertex with the highest value of R (if the new R is higher than the current R) is added to the community and the three sets are updated. Algorithm repeats this process until the current R cannot be raised anymore. Further description of this algorithm is beyond the scope of this paper, for the details see [17].

In the article, authors work only with unweighted graph, but a lot of today's social networks can be easily transformed to weighted. The algorithm can be quite easily adapted to weighted networks. Just instead of the number of edges will algorithm work with the sum of their weights.

In the article [22] T. Opsahl presented the new weighted variants of algorithms for vertex centrality measurement. He came with an idea to combine unweighted and weighted types of degree of a vertex and of shortest paths in a single calculation.

Let us define C as number of edges and W as sum of edges either in a shortest path or in a degree of a vertex. Then his combined formula looks like:

$$O = C^{(1-alpha)} \cdot W^{alpha} \tag{3}$$

where $alpha$ is the number that affects the importances of both parts of the formula. When $alpha$ is 1, this formula simulates common weighted variant and when $alpha$ is 0, the common unweighted variant is simulated. Values between 0 and 1 changes the ratio of how both parts affect the result. Values lesser than 0 or greater than 1 make the particular part affect the result in a negative way, e.g. for values higher than 1 the vertex (path) with less edges is favored over the one with more edges even if the sums of weights of their edges are the same.

In this paper, we have made a modification of the iterative local expansion algorithm described in [17] using the Opsahl's idea. Instead of measuring just number of edges or just sum of their weights, we used Opsahl's formula to combine them during the community search.

IV. Experiment

A. Dataset

For our experiment, we have used a part of a data collection of a weighted network from our Forcoa.NET[3] system. This weighted network is based on the DBLP dataset[4] of publications from the field of computer science. These data contain highly relevant information about publication activity from the period of nearly fifty years and are freely available[5]. Total number of authors was 1,060,175 with 6,450,138 edges. After we have performed a network denoising based on forgetting concept [23], the set used in experiment contained 96,172 authors and 67,854 edges in total, however for local methods only the immediate surroundings of a selected author was necessary to examine.

[3]http://www.forcoa.net

[4]http://www.informatik.uni-trier.de/ ley/db/

[5]http://dblp.uni-trier.de/xml/

B. Settings

In this section the two algorithms described in III are used for community detection on a sample of weighted network. Both algorithms started with a preselected vertex, for Algorithm 1 we have chosen the threshold of 0.5, which corresponds with natural intuition, for measuring the affiliation of a vertex to a community. If we further consider that vertex x is dependent on vertex y, respectively on the set of vertices Y, we have to assume that it holds: $D(x, y) \geq 0.5$, resp. $D(x, Y) \geq 0.5$.

For Algorithm 2 we have selected $alpha$ as 0.5, so both the weights and the connectivity between the vertices were equally taken into consideration.

C. Results

When starting from a particular single vertex, communities detected by each of the algorithms were always slightly different, for example see community around 'Vaclav Snasel' and 'H. Vincent Poor[6]' in Fig. 4 and Fig. 5. Both algorithms may start also from an edge or a set of vertices. In this case, the main difference between the algorithms was that when starting from an edge or set of vertices that belonged to previously detected community, Algorithm 2 detected the same community, but Algorithm 1 detected a set of overlapping sub-communities and a nested set of sub-communities. We have listed those sub-communities in Tab. I and Tab. II. From our experience we may corroborate, that in case of author 'Vaclav Snasel', the detected sub-communities correspond to research sub–teams.

V. Conclusions

In our paper, we compared two local methods of community detection in weighted networks. We observed differences in detected communities, because of different definitions of community within the algorithms. The results of comparison showed that both algorithms detected very similar communities. The first algorithm has stronger requirements for adding vertex into a community (especially the need to be connected to at least other two community vertices). Because of that, it is capable of detecting overlapping and nested sub-communities, which according to our observation within our research group, corresponds very well with the reality. This algorithm is also used in our public on-line Forcoa.NET system.

In our future research we will focus on analysing properties shared by vertices in detected communities and on the comparison with other algorithms for community detection.

Acknowledgment

This paper was supported by the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic; by the SoftComp: Development of human resources in research

[6]H. Vincent Poor was selected as an author with most records in DBPL database

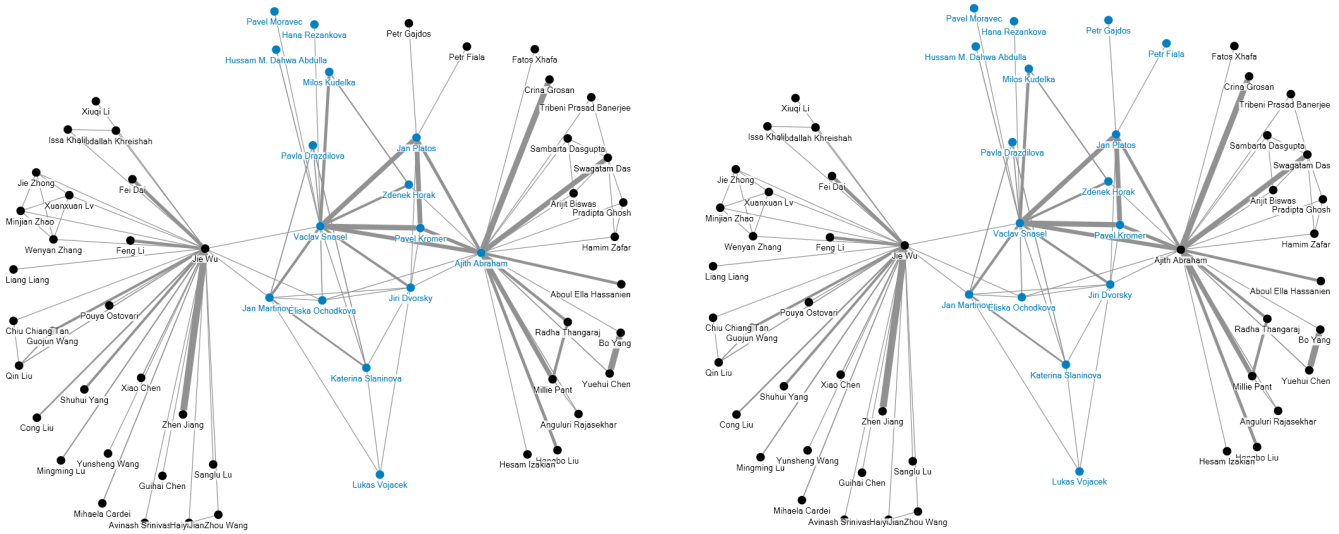

Fig. 4. Detected community with Algorithm 1 (left) and Algorithm 2 (right) for 'Vaclav Snasel'.

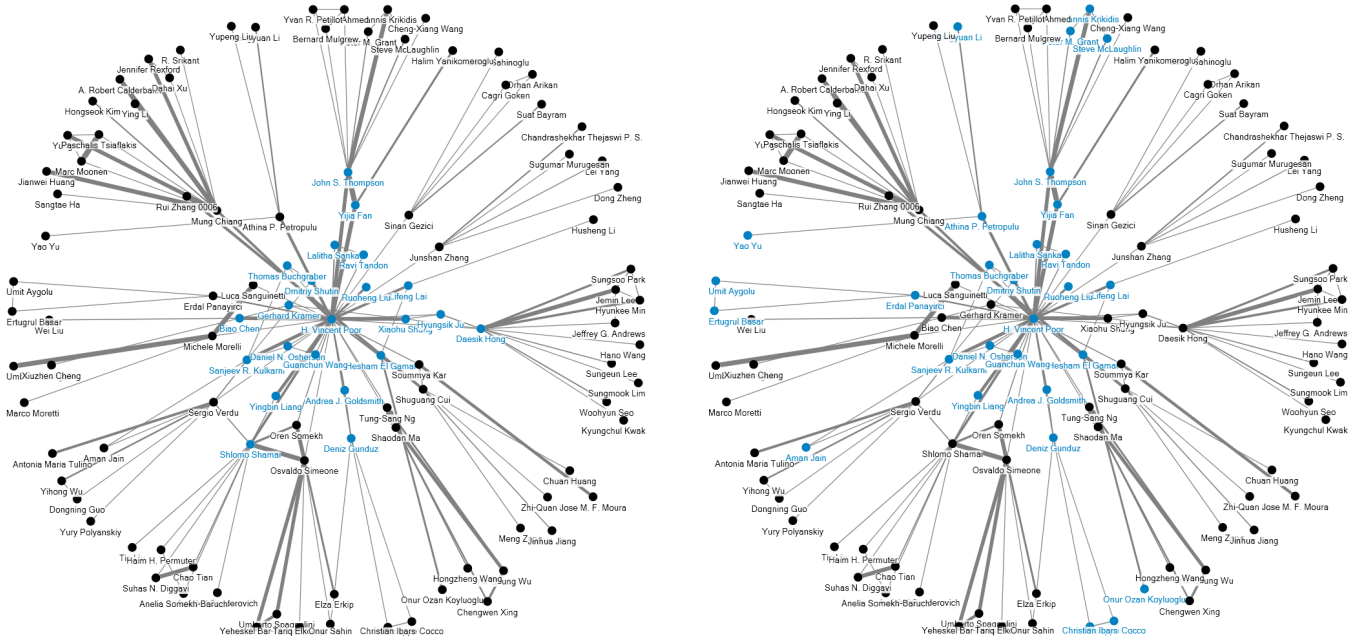

Fig. 5. Detected community with Algorithm 1 (left) and Algorithm 2 (right) for 'H. Vincent Poor'.

and development of innovative softcomputing methods and their practical use, reg. no.CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic; by SGS, VSB-Technical University of Ostrava, under the grant no. SP2012/58.

REFERENCES

[1] A. Barabasi, "Linked. the new science of networks. how everything is connected to everything else and what it means for science, business and everyday life," *Perseus, Cambridge*, 2002.

[2] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical review E*, vol. 72, no. 2, p. 027104, 2005.

[3] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, p. 2658, 2004.

[4] J. Leskovec, K. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 631–640.

[5] M. Girvan and M. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, p. 7821, 2002.

[6] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[7] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[8] J. Bagrow and E. Bollt, "Local method for detecting communities,"

TABLE I
SUB-COMMUNITIES DETECTED AROUND 'VACLAV SNASEL'.

Sub-community	Vertices
L_1	Vaclav Snasel, Pavel Kromer, Jiri Dvorsky, Katerina Slaninova, Pavla Drazdilova, Lukas Vojacek, Milos Kudelka, Zdenek Horak, Jan Martinovic, Ajith Abraham, Jan Platos, Eliska Ochodkova
L_2	Vaclav Snasel, Pavel Kromer, Jiri Dvorsky, Katerina Slaninova, Pavla Drazdilova, Lukas Vojacek, Jan Martinovic, Ajith Abraham, Jan Platos, Eliska Ochodkova
L_3	Vaclav Snasel, Jiri Dvorsky, Katerina Slaninova, Pavla Drazdilova, Lukas Vojacek, Jan Martinovic, Ajith Abraham, Eliska Ochodkova
L_4	Vaclav Snasel, Jiri Dvorsky, Jan Martinovic, Ajith Abraham, Eliska Ochodkova
L_5	Vaclav Snasel, Pavel Kromer, Ajith Abraham, Jan Platos
L_6	Vaclav Snasel, Milos Kudelka, Zdenek Horak
L_7, L_8, L_9	Vaclav Snasel, Hana Rezankova | Vaclav Snasel, Pavel Moravec | Vaclav Snasel, Hussam M. Dahwa Abdulla
Set relations	$L_4 \subset L_3 \subset L_2 \subset L_1$, $L_5 \subset L_2, L_6 \subset L_1$, $L_4 \cap L_5 = L_3 \cap L_5 =$ {Vaclav Snasel, Ajith Abraham}

TABLE II
SUB-COMMUNITIES DETECTED AROUND 'H. VINCENT POOR'.

Sub-community	Vertices
L_1	H. Vincent Poor, Daniel N. Osherson, Guanchun Wang, Dmitriy Shutin, Thomas Buchgraber, Sanjeev R. Kulkarni
L_2	H. Vincent Poor, Gerhard Kramer, Xiaohu Shang, Biao Chen, Hyungsik Ju, Daesik Hong
L_3	H. Vincent Poor, Daniel N. Osherson, Sanjeev R. Kulkarni, Guanchun Wang
L_4	H. Vincent Poor, Thomas Buchgraber, Dmitriy Shutin, Sanjeev R. Kulkarni
L_5	H. Vincent Poor, Xiaohu Shang, Gerhard Kramer, Biao Chen
L_6	H. Vincent Poor, Xiaohu Shang, Hyungsik Ju, Daesik Hong
L_7	H. Vincent Poor, Lalitha Sankar, Ravi Tandon
L_8	H. Vincent Poor, Andrea J. Goldsmith, Deniz Gunduz
L_9	H. Vincent Poor, Yingbin Liang, Shlomo Shamai
L_{10}	H. Vincent Poor, Yijia Fan, John S. Thompson
L_{11}	H. Vincent Poor, Lifeng Lai, Hesham El Gamal
L_{12}	H. Vincent Poor, Ruoheng Liu
Set relations	$L_3 \subset L_1, L_4 \subset L_1$, $L_4 \cap L_4 =$ {H. V. Poor, Sanjeev R. Kulkarni} $L_5 \subset L_2, L_6 \subset L_2$, $L_5 \cap L_6 =$ {H. V. Poor, Xiaohu Shang}

Physical Review E, vol. 72, no. 4, p. 046108, 2005.

[9] Z. Horak, M. Kudelka, V. Snasel, A. Abraham, and H. Rezankova, "Forcoa. net: An interactive tool for exploring the significance of authorship networks in dblp data," in *Computational Aspects of Social Networks (CASoN), 2011 International Conference on*. IEEE, 2011, pp. 261–266.

[10] M. Newman, "Detecting community structure in networks," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 38, no. 2, pp. 321–330, 2004.

[11] S. Schaeffer, "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27–64, 2007.

[12] A. Clauset, M. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

[13] K. Wakita and T. Tsurumi, "Finding community structure in mega-scale social networks," *Arxiv preprint cs/0702048*, 2007.

[14] S. Gregory, "A fast algorithm to find overlapping communities in networks," *Machine Learning and Knowledge Discovery in Databases*, pp. 408–423, 2008.

[15] J. Bagrow, "Evaluating local community methods in networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, p. P05001, 2008.

[16] J. Chen, O. Zaïane, and R. Goebel, "Local community identification in social networks," in *Social Network Analysis and Mining, 2009. ASONAM'09. International Conference on Advances in*. IEEE, 2009, pp. 237–242.

[17] J. Chen, O. Zaiane, and R. Goebel, "Detecting communities in large networks by iterative local expansion," in *Computational Aspects of Social Networks, 2009. CASON'09. International Conference on*. IEEE, 2009, pp. 105–112.

[18] A. Clauset, "Finding local community structure in networks," *Physical Review E*, vol. 72, no. 2, p. 026132, 2005.

[19] F. Luo, J. Wang, and E. Promislow, "Exploring local community structures in large networks," *Web Intelligence and Agent Systems*, vol. 6, no. 4, pp. 387–400, 2008.

[20] P. Orponen and S. Schaeffer, "Efficient algorithms for sampling and clustering of large nonuniform networks," *Arxiv preprint cond-mat/0406048*, 2004.

[21] M. Kudělka, P. Dráždilová, E. Ochodková, K. Slaninová, and Z. Horák, "Local community detection and visualization: Experiment based on student data," in *Proceedings of the Third International Conference on Intelligent Human Computer Interaction (IHCI 2011), Prague, Czech Republic, August, 2011*. Springer, pp. 291–303.

[22] T. Opsahl, F. Agneessens, and J. Skvoretz, "Node centrality in weighted networks: Generalizing degree and shortest paths," *Social Networks*, vol. 32, no. 3, pp. 245–251, 2010.

[23] M. Kudelka, Z. Horak, V. Snasel, and A. Abraham, "Social network reduction based on stability," in *Computational Aspects of Social Networks (CASoN), 2010 International Conference on*. IEEE, 2010, pp. 509–514.

[24] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Physical Review E*, vol. 80, no. 5, p. 056117, 2009.

[25] T. Opsahl and P. Panzarasa, "Clustering in weighted networks," *Social networks*, vol. 31, no. 2, pp. 155–163, 2009.