Multi-leader Selection in Complex Networks

Dan Wang, Wei Chen, and Li Qiu

***Abstract*— In this paper, we study the multi-leader selection problem in complex networks. While selecting a single leader can be done via various centrality measures, selecting multiple leaders is much more involved than a simple order of the nodes in terms of centrality measures. In many situations, it is often desirable to see that the multiple leaders selected are as representative as possible. Motivated by this, a clustering based two-step approach is proposed in this paper. Specifically, in order to select k leaders in a complex network, we first partition the network into k clusters and then find a leader within each cluster. For network partitioning, we propose a hierarchical algorithm by exploiting the properties of the Fiedler vector. For the single leader selection in each cluster, we resort to the eigenvector centrality, the closeness centrality and the effective resistance as useful tools. Examples on several real-world networks are worked out to illustrate the effectiveness of our method.**

I. Introduction

Large-scale networked systems are becoming ubiquitous. Examples include social networks [1], biological networks [2], data networks [3], power networks [4], [5], distributed control systems [6], and transportation networks [7]. Efficient manipulation and control of such networked systems is of central importance in various research fields. A fundamental issue arising in different applications is the leader selection in such networks. In social networks, selecting leaders facilitates the study of behavior influence in the network. In biological networks, recognising vital proteins in protein-protein interaction networks helps biologists to study the cellular modules or even find ways to cure diseases. In power networks, identifying the most vulnerable nodes enhances the security of the grid and placing the phasor measurement units (PMUs) in proper positions improves the estimation performance. In distributed control systems, deploying the controllers in a desired way enhances the control performance.

Many studies on single leader selection have been reported during past decades, e.g., [8], [9]. A natural method for single leader selection in a network is to order the nodes via a certain centrality measure and then choose the top-ranked node as the leader. Now if one wishes to select k leaders from the network, a simple analogy would be to order the nodes and then choose the top k forerunners. However, this may not be a wise strategy in many applications since the top-ranked nodes may contain similar or even identical information. For instance, in social networks, the most influential members may come from a small community and, thus, do not serve as a good representation of the whole network. For the PMU placement in power networks, the top-ranked nodes may produce highly correlated measurements and, thus, may not be a good choice for estimation purposes.

As noted above, multi-leader selection is not a simple inheritance of single leader selection; more factors need to be taken into consideration. In particular, we are driven by the idea that in many applications, it is desirable to have the selected leaders spread across the network such that they can contain complementary information. For instance, in social networks, one may wish to see leaders elected from different communities to have more diverse voices. In power networks, one may wish to place the PMUs widely spread over the network so as to have better estimation performance. In Internet networks, one may wish to select webpages that contain important information on different topics rather than those on a single hot topic.

Inspired by this idea, we propose a clustering based two-step approach for multi-leader selection. To be specific, we first partition the network into different clusters and then select a leader within each cluster. For network partitioning, we propose a hierarchical algorithm by exploiting the properties of the Fiedler vector. For the single leader selection in each cluster, we resort to the eigenvector centrality, the closeness centrality and the effective resistance as useful tools. We apply our approach to several real-world networks to verify its effectiveness.

The rest of this paper is organized as follows. Section II gives some preliminary knowledge on spectral graph theory and effective resistance. Section III presents in detail the proposed clustering-based multi-leader selection. The proposed approach is then applied to several real-world networks in Section IV. Some concluding remarks follow in Section V.

II. Mathematical Preliminaries

A. Spectral graph theory

Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ denotes the node set, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the edge set, and $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ denotes the nonnegative weighted *adjacency matrix*. An edge $(i, j) \in \mathcal{E}$ if and only if $a_{ij} > 0$. Self-loops and multiple edges are not considered. A sequence of edges $(i_1, i_2), (i_2, i_3), \ldots, (i_{k-1}, i_k)$ with $(i_j, i_{j+1}) \in \mathcal{E}$ for all $j \in \{1, \ldots, k-1\}$ is called a *path* from node i_1 to node i_k. Let $\mathcal{M}$ be a subset of $\mathcal{V}$. The *induced subgraph* $\mathcal{G}(\mathcal{M})$ is a subgraph whose node set is

This work was partially supported by Research Grants Council of Hong Kong Special Administrative Region, China under the Theme-Based Research Scheme T23-701/14-N.

Dan Wang and Li Qiu are with the Department of Electronic & Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China (dwangah@connect.ust.hk, eeqiu@ust.hk). Wei Chen is with the Department of Electrical Engineering & Computer Sciences, University of California at Berkeley, USA (wchenust@gmail.com).

$\mathcal{M}$ and whose edge set consists of all the edges in $\mathcal{E}$ with both endpoints in $\mathcal{M}$. Define the *degree* of node i as

$$d_i = \sum_{j=1}^{n} a_{ij}.$$

Denote the *degree matrix* of $\mathcal{G}$ by

$$D = \text{diag}\{d_1, d_2, \ldots, d_n\}.$$

The *Laplacian matrix* L of $\mathcal{G}$ is defined as

$$L = D - A.$$

The Laplacian L is positive semidefinite and has a zero eigenvalue with a corresponding eigenvector being $\mathbf{1} \in \mathbb{R}^n$, where $\mathbf{1}$ denotes the vector with all elements equal to 1 [10]. Suppose $n \geq 2$. Denote the eigenvalues of L by $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and the chosen orthogonal eigenvectors by $x_1, x_2, \ldots, x_n$, respectively. The second smallest eigenvalue λ_2 is called the *algebraic connectivity* [11] and any corresponding eigenvector x_2 is called a *Fiedler vector* of the graph [13].

The following lemma shows that the algebraic connectivity is monotonically increasing with the edge weights.

Lemma 1 ([12]): Given two graphs $\mathcal{G}_1 = (\mathcal{V}, \mathcal{E}_1, A_1)$ and $\mathcal{G}_2 = (\mathcal{V}, \mathcal{E}_2, A_2)$. Denote by L_1 and L_2 the Laplacians of $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively. If $A_1 \leq A_2$ holds elementwise, then $\lambda_2(L_1) \leq \lambda_2(L_2)$.

We also state a useful property of the Fiedler vector x_2. Define the following subsets:

$$\begin{aligned}
\mathcal{M}_{\geq 0} &= \{i \in \mathcal{V} \,|x_{2i} \geq 0\}, \\
\mathcal{M}_{\leq 0} &= \{i \in \mathcal{V} \,|x_{2i} \leq 0\}, \\
\mathcal{M}_{> 0} &= \{i \in \mathcal{V} \,|x_{2i} > 0\}, \\
\mathcal{M}_{< 0} &= \{i \in \mathcal{V} \,|x_{2i} < 0\},
\end{aligned}$$

where x_{2i} means the ith element of x_2.

Lemma 2 ([13]): If $\mathcal{G}$ is connected, then both induced subgraphs $\mathcal{G}(\mathcal{M}_{\geq 0})$ and $\mathcal{G}(\mathcal{M}_{\leq 0})$ are connected. Furthermore, if $\{i \in \mathcal{V} \,|x_{2i} = 0\} = \emptyset$, then the induced subgraphs $\mathcal{G}(\mathcal{M}_{> 0})$ and $\mathcal{G}(\mathcal{M}_{< 0})$ are connected.

B. Effective resistance

Consider a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$. Place a resistor to each edge with resistance equal to the reciprocal of the edge weight. Then a resistive electrical network associated with the graph $\mathcal{G}$ is obtained. Suppose a unit of current is injected into one node i and extracted from another node j. Then the voltage difference between node i and node j can be calculated by using basic circuit theory [15]. This quantity is called the *effective resistance* between node i and node j [16]. The notion of effective resistance has many applications over a wide variety of fields [17], [18], and has recently attracted much attention from the system and control community [19].

The effective resistance can be computed via the Moore-Penrose pseudoinverse $L^\dagger = [l^\dagger_{ij}]$ of the Laplacian L. Let $u_k \in \mathbb{R}^n$ denote the vector whose kth element is 1 and all the other elements are 0. Then the effective resistance between nodes i and j can be calculated by [16]

$$r_{ij} = (u_i - u_j)^T L^\dagger (u_i - u_j) = l^\dagger_{ii} - 2l^\dagger_{ij} + l^\dagger_{jj}.$$

Since $L^\dagger$ is symmetric, we have $r_{ij} = r_{ji}$. A property of the effective resistance is stated as below, which will play an important role in our algorithm for leader selection.

Lemma 3 ([16]): The effective resistance gives a distance function in $\mathcal{V}$ of $\mathcal{G}$.

III. Identification of Leaders in a Network

We study the multi-leader selection problem in this paper. One main motivation of this work comes from applications aiming to maximize the influence on a network with limited resources. For example, deploying limited number of controllers in a distributed control system. Therefore, we assume the number of leaders is given. We are aware that in some literature, the number of leaders is also left as a decision variable, due to the nature of the applications. However, that kind of setup is out of the scope of this study.

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, we wish to find k leaders as a good representation of the whole graph. As reasoned before, multi-leader selection is much more involved than a simple inheritance of single leader selection. Ranking the nodes via a certain centrality measure and selecting the first k forerunners may not be a good choice in many contexts, especially when diversity is of essential importance. Take the ten-node path shown in Fig. 1 as an example. The nodes 5 and 6 are top-ranked nodes by the closeness centrality, which will be formally introduced later. If controllers are deployed in these two nodes in a distributed control system, their coverage is limited and thus the operation on node 1 and node 10 might be very weak, which is not desirable for control purposes. In power networks, if two PMUs are placed on nodes 5 and 6, we may lose the measurements of nodes 1 and 10. In both scenarios, node 3 and 8 serve as better choices of leaders, as shown in Fig. 2. In fact, these two nodes are picked out by our algorithm, which will be introduced bellow.

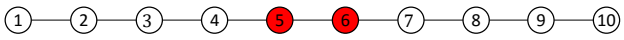

Fig. 1. Two leaders by ranking

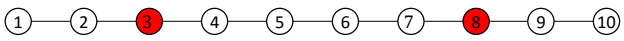

Fig. 2. Two leaders by our algorithm

Before formally introducing the algorithm, let us first look into an extreme case for some intuitions. Suppose the graph $\mathcal{G}$ is disconnected and can be decomposed into k connected subgraphs. In this case, to select k leaders in the network, it appears quite reasonable to choose a leader from each subgraph. In this way, each leader selected contains the information of the subgraph to which it belongs. Hence, these leaders serve as a good representation of the whole graph.

Inspired by this intuition, for a given graph $\mathcal{G}$, connected or not, we propose a clustering based two-step approach to find k leaders in the graph, i.e., partitioning the network into k clusters and then selecting a leader within each cluster.

A. Network partitioning

We propose a hierarchical algorithm to partition the graph $\mathcal{G}$ into k clusters, namely, to partition the node set $\mathcal{V}$ into k distinct subsets.

Firstly, bipartition $\mathcal{G}$ into two clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ according to the sign pattern of the Fiedler vector of $\mathcal{G}$. Specifically, the nodes corresponding to the nonnegative entries in the Fiedler vector are put in one cluster and the others are put in another cluster. The underlying rationale is due to the nice property of Fiedler vector stated in Lemma 2. Since the nodes corresponding to the zero entries in the Fiedler vector are boundaries of two clusters, they will not be chosen as leaders. So they can be put together with the nodes corresponding to positive or negative entries in the Fiedler vector. In fact, it has been pointed out in the literature that such bipartition via the Fiedler vector has the effect of maximizing the sum of weights of intra-cluster edges and minimizing the sum of weights of inter-cluster edges. Notice that such bipartition relies on the simplicity of λ_2 such that the Fiedler vector is unique (up to multiplication by scalars). If unfavorably, λ_2 is not simple, one can slightly perturb the Laplacian to avoid the trouble from the nonuniqueness of the Fiedler vector.

To go from two clusters to k clusters, we use a hierarchical approach by recursively implementing the above bipartition method. Suppose at some point in the process, l clusters are already in place. We then choose one cluster and bipartition it into two subclusters. Keep repeating this process until k clusters are formed. The key issue in this process is to decide which cluster to choose for further bipartition. To this end, we appeal to the algebraic connectivity as a useful tool. From Lemma 1, we know that the algebraic connectivity λ_2 reflects the degree of connectivity of a graph. The smaller λ_2 is, the sparser a graph is and the more likely it has a clustering structure. In order to compare two graphs with possible different sizes, the normalized algebraic connectivity λ_2/n is used instead of λ_2. Hence, in every step of the recursive bipartition, we choose the cluster whose induced subgraph has the smallest normalized algebraic connectivity for further bipartition.

The hierarchical partition introduced above is summarized in Algorithm 1.

We are aware that there are various algorithms for network clustering in the literature, including the well known spectral clustering algorithms [20]. See Algorithm 2 for the details of the spectral clustering method proposed in [22]. An issue worth attention is that the spectral clustering algorithm is sensitive to the choice of initial conditions due to the use of k-means [21] algorithm. We illustrate this point by a simple example of partitioning a twenty-node cycle graph to two clusters. As shown in Fig. 3, spectral clustering may produce two clusters with 9 nodes and 11 nodes, respectively, if the initial condition is not chosen carefully. This contradicts the intuition that the highly symmetrical cycle graph should be cut along any diameter so that the resultant two clusters have equal number of nodes. Nevertheless, our algorithm adopts a recursive bipartition approach based on the sign patten of the Fiedler vector and, thus, does not have the concern about the initial conditions. With our algorithm, the cycle graph is indeed cut along a diameter.

Algorithm 1 Hierarchical clustering algorithm

Input: A: the adjacency matrix, k: the number of clusters.

1: Compute the Laplacian matrix L and a corresponding Fiedler vector x_2.
2: Partition the nodes of the graph into two clusters: $\mathcal{C}_1 = \mathcal{M}_{\geq 0}$ and $\mathcal{C}_2 = \mathcal{M}_{<0}$.
3: Calculate λ_2/n of the present clusters. Find the cluster whose induced subgraph has the smallest normalized algebraic connectivity for continuing partitioning.
4: Repeat from step 1 until the number of clusters increases to k.

Output:

$\mathcal{C}_1, \ldots, \mathcal{C}_k$: k clusters.

Algorithm 2 Normalized spectral clustering [22]

Input: A: the adjacency matrix, k: the number of clusters.

1: Compute the degree matrix D and the normalized Laplacian matrix $L_n = D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}}$.
2: Let $U = \begin{bmatrix} x_1 & x_2 & \cdots & x_k \end{bmatrix}$, where $x_1, \ldots, x_k$ are the first k eigenvectors of L_n. Normalize each row of U (with the Hölder 2-norm) to get U_n.
3: Let $y_j \in \mathbb{R}^k$ $(j = 1, \ldots, n)$ be the jth row of U_n.
4: Cluster y_j in $\mathbb{R}^k$ into k clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$ using the k-means algorithm.

Output:

$\mathcal{C}_1, \ldots, \mathcal{C}_k$: k clusters.

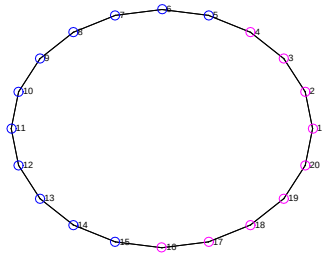

Fig. 3. Partition of a cycle graph by spectral clustering.

B. Single leader selection in a network

After partitioning a network into k clusters, we shall find a leader within each cluster. We provide three methods to select the leader.

1) Leader selection by the eigenvector centrality: Eigenvector centrality is a measure of the influence of the nodes in a network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ [23]. The basic idea comes from the perception that a node connected to important nodes is also important. In this regard, the centrality of a node can be

evaluated as a weighted sum of the centralities of the nodes connected to it. Formally, if we denote the centrality of node i by p_i, then

$$p_i = \frac{1}{\lambda} \sum_{j=1}^{n} a_{ij} p_j,$$

where λ is a constant. Denote the centrality vector by $p = \begin{bmatrix} p_1 & p_2 & \dots & p_n \end{bmatrix}'$. The above equation can be rewritten in matrix form as

$$\lambda p = Ap.$$

Clearly, λ is an eigenvalue of A and p is a corresponding eigenvector. It is customary to choose λ to be the eigenvalue of A with the largest magnitude. By the Perron-Frobenius theorem [14], there exists a corresponding eigenvector p having all nonnegative entries, which is referred to as Perron vector. We rank all nodes in a graph according to the entries of the Perron vector of the adjacency matrix A and then select the top-ranked node as the leader. See Algorithm 3.

Algorithm 3 Single leader selection via PerronRank

Input: A: the adjacency matrix.

1: Compute the Perron vector p of A.
2: Find the leader corresponding to the largest entry of p.

Output:

The leader.

2) Leader selection by the closeness centrality: The closeness centrality gives another useful tool for leader selection, especially in applications where quick spread of information is crucial. Define the length of a path to be the sum of the reciprocals of the weights of the traversed edges[1]. Among all paths between two nodes, the shortest path is the one with the minimum length. The distance between node i and node j, denoted by d_{ij}, is defined to be the length of the shortest path between them. The closeness centrality [24] of a node is then defined as the reciprocal of the sum of its distances to all other nodes.

The node with highest closeness centrality is selected to be the leader. This means that the leader has the smallest sum of distances to the other nodes. Mathematically, it boils down to the following optimization problem:

$$\min_{i=1,2,\dots,n} \sum_{j} d_{ij}. \tag{1}$$

Algorithm 4 Single leader selection via closeness centrality

Input: $\mathcal{G}$: a connected graph.

1: Compute the distance matrix $D = [d_{ij}]$.
2: Solve the optimization problem (1).

Output:

The leader.

[1]This is different from the conventional definition that simply takes the sum of the weights of the traversed edges to be the length of a path. The rationale for taking the reciprocals of the weights comes from the intuition that large weight indicates strong bonding in many applications.

3) Leader selection by the effective resistance: As stated in Lemma 3, the effective resistance gives a distance function in the node set of a graph. It quantitatively measures the bonding between two nodes by taking all the paths between them into account. The smaller the effective resistance is, the more bonding two nodes are. In this regard, a leader can be selected via the effective resistance.

Specifically, denote by $R = [r_{ij}]$ the effective resistance matrix of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where r_{ij} is the effective resistance between node i and j. Denote R_i as the ith row of R. Then finding a leader in the network can be formulated as the following optimization problem:

$$\min_{i=1,2,\dots,n} \|R_i\|, \tag{2}$$

where $\|\cdot\|$ takes Hölder p-norm. The choice of norm depends on the specific context. For instance, if $\|\cdot\|_1$ is used, the node having the smallest sum of effective resistance with other nodes is selected. If $\|\cdot\|_\infty$ is used instead, then the node having the smallest maximum effective resistance with other nodes is selected.

The detailed algorithm is presented below.

Algorithm 5 Single leader selection via effective resistance

Input: $\mathcal{G}$: a connected graph.

1: Compute the effective resistance matrix $R = [r_{ij}]$.
2: Solve the optimization problem (2).

Output:

The leader.

C. Computational complexity

In the two-step approach, for the network partitioning, we resort to the algebraic connectivity and Fiedler vector of the network Laplacian. Only λ_2 and x_2 need to be calculated, which can be efficiently done by distributed methods such as the Jacobi-Davidson method [25]. It is often the case that each cluster is of small size after network partitioning. The centrality measure can then be computed easily. In the case when the cluster size is relatively big, distributed approaches can be used to calculate the centralities as in the literature, for instance, [26]. Therefore, our method works efficiently for large-scale networks. Examples are depicted in next section.

IV. Experimental Results

We apply our algorithm to several real-world networks. In each case, our algorithm reliably identifies the leaders.

A. Zachary's karate club network

We first illustrate our method by identifying the influential persons in a benchmark social network: Zachary's Karate Club [27]. It consists of 34 nodes, the members of a karate club, who were observed over a period of three years. The edges in the network represent the interactions among the members outside the activities of the club. During the course of the study, a disagreement developed between the president and the instructor of the club, which ultimately resulted in the fission of the club into two separate groups, one in support

of the instructor and one in support of the president. Here we use a weighted version of Zachary's network and apply our algorithm to it.

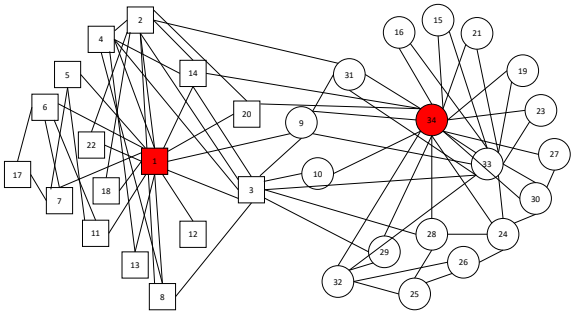

Fig. 4. Two leaders in Zachary's karate club

As shown in Fig. 4, the nodes depicted by squares belong to one cluster and the nodes depicted by circles belong to another cluster. Node 1 and node 34 are the leaders we found (by all three methods we mentioned). The partition consists with the actual division of the club members after the break-up, as revealed by which club they attended afterwards. The two leaders are indeed the key members of the club as they correspond to the president and the instructor.

In this experiment as well as the later ones, when using the effective resistance to identify the leader within each cluster, we adopt Hölder 1-norm in the optimization problem (2).

B. IEEE 30-bus system

We now turn to applying our algorithm to the placement of PMUs in the IEEE 30-bus system [4]. Fig. 5 shows the simple transmission grid model of the system. Each bus denoted by a horizontal or vertical bar represents a power plant, a substation, or a load. The buses are connected by transmission lines, forming a power network. We express the topology of the grid as an undirected graph. Each node in the graph corresponds to a bus in the grid, and edges connecting the nodes correspond to the transmission lines. We would like to install 9 PMUs in the grid, which ensure observability of the full system [28]. The graph partitioning result is shown in Table I and the positions for PMUs to be placed are shown in Fig. 5.

It is interesting to see that the single leader selection within each cluster by all three methods coincide. This is partially because each cluster contains no more than five nodes and the induced graph is simple, i.e., a cycle or a tree. In general, we do not expect the results yielded by the three methods to coincide.

The positions we find to place PMUs are highly consistent with the results in [28]. Only two positions, 6 and 23, are selected differently.

TABLE I

CLUSTERS OF THE IEEE 30-BUS SYSTEM

Clusters of the network	{1 2 3 4}, {5 7}, {6 10 21}, {8 28}, {12 13 14 15 23}, {9 11}, {16 17}, {18 19 20}, {22 24 25 26}, {27 29 30}.

C. Co-publishing Network (DBLP)

We also apply our algorithm to real-world large-scale networks to illustrate the effectiveness. We choose the DBLP collaboration network [29] from the Stanford Large Network Dataset Collection, which has ground-truth communities.

The DBLP provides a comprehensive list of research papers in computer science. A co-authorship network is constructed, where nodes represent authors and adjacent nodes represent authors with at least one shared publication. Ground-truth communities are defined as sets of authors who have published in the same journal or conference publication.

We choose a subgraph which contains the top 1000 ground-truth communities from the network for evaluation. This subgraph contains 9715 nodes and 31860 edges. Our algorithm is used to find 1000 leaders. We regard each connected component in the network as a cluster, and implement our hierarchical algorithm to get 1000 clusters. Then we find the single leader in each cluster with the three approaches presented in last section.

A local perspective of the results is shown in Fig. 6. In the figure, an empty circle represents a node of the network, nodes within each large red circle are in the same cluster, and red, yellow and blue nodes in each cluster are leaders given by eigenvector centrality, closeness centrality and effective resistance, respectively. Only a red node is shown if they coincide. The results for the whole network are left out because of page limitations, and can be found from authors.

We choose one measure method in the field of community detection to compare our partitioning results with the ground-truth communities. We calculate the correlation of the similarity matrices of the ground-truth community and the resultant cluster to measure the cluster validity [30]. The correlation is 0.6251, which means that our results are highly consistent with the real community structures.

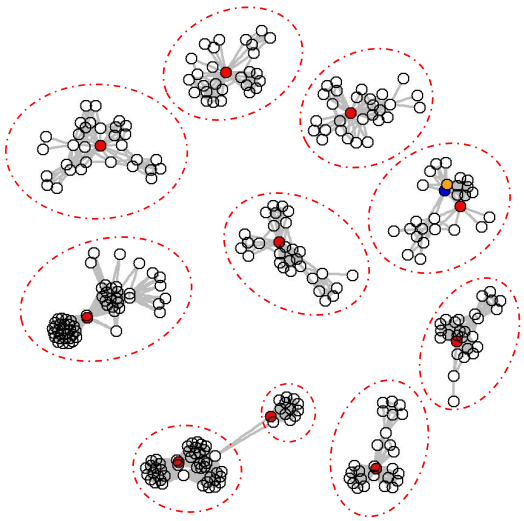

Fig. 6. A local sample of leaders in DBLP

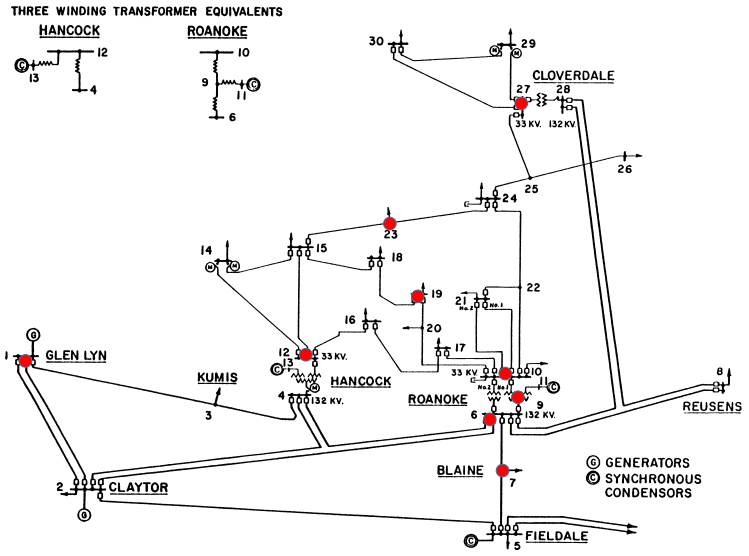

Fig. 5. IEEE 30-bus system with 9 PMUs placed

V. Conclusions

In this paper, we propose a method for multi-leader selection in networks of various kinds via a clustering based two-step approach. Specifically, to find k leaders in a network, we first partition the network into k clusters and then find a leader within each cluster. For the network partitioning, a hierarchical algorithm is proposed via a recursive implementation of bipartition. As for the leader selection in each cluster, three methods based on eigenvector centrality, closeness centrality, and effective resistance are used respectively. The proposed algorithm is applied to several real-world networks to illustrate its effectiveness.

References

[1] M. E. J. Newman, "The structure of scientific collaboration networks," *Proc. Natl. Acad. Sci.*, vol. 98, no. 2, pp. 404-409, 2001.

[2] P. F. Jonsson, T. Cavanna, D. Zicha, and P. A. Bates, "Cluster analysis of networks generated through homology: Automatic identification of important protein communities involved in cancer metastasis," *BMC Bioinformatics*, vol. 7, no. 2, pp. 1-13, 2006.

[3] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, and J. Wiener, "Graph structure in the web," *Computer Networks*, vol. 33, no. 1, pp. 309-320, 2000.

[4] R. D. Christie, "Power systems test case archive," [Online]. Available: http://www.ee.washington.edu/research/pstca/.

[5] P. Frasca, H. Ishii, C. Ravazzi, and R. Tempo, "Distributed randomized algorithms for opinion formation, centrality computation and power systems estimation: a tutorial overview," *Eur. J. Control*, vol. 24, pp. 2-13, 2015.

[6] P. Massioni and M. Verhaegen, "Distributed control for identical dynamically coupled systems: a decomposition approach," *IEEE Trans. Automat. Contr.*, vol. 54, no. 1, pp. 124-135, Jan. 2009.

[7] P. Varaiya, "Smart cars on smart roads: problems of control," *IEEE Trans. Automat. Contr.*, vol. 38, no. 2, pp. 195-207, Feb. 1993.

[8] S. Patterson and B. Bamieh, "Leader selection for optimal network coherence," in *Proc. 49th IEEE Conf. Decision Contr.*, pp. 2692-2697, 2010.

[9] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press, 2010.

[10] F. R. K. Chung, *Spectral Graph Theory*, American Mathematical Society, 1997.

[11] M. Fiedler, "Algebraic connectivity of graphs," *Czech. Math. J.*, vol. 23, no. 2, pp. 298-305, 1973.

[12] J. J. Molitierno, *Applications of Combinatorial Matrix Theory to Laplacian Matrices of Graphs*, CRC Press, 2012.

[13] M. Fiedler, "A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory," *Czech. Math. J.*, vol. 25, no. 4, pp. 619-633, 1975.

[14] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[15] E. S. Kuh and C. A. Desoer, *Basic Circuit Theory*, McGraw-Hill Book Company, 1969.

[16] D. Klein and M. Randić, "Resistance distance," *J. Math. Chem.*, vol. 12, pp. 81-95, 1993.

[17] J. Palacios, "Resistance distance in graphs and random walks," *Int. J. Quantum Chem.*, vol. 81, no. 1, pp. 29-33, 2001.

[18] I. Gutman and B. Mohar, "The quasi-Wiener and the Kirchhoff indices coincide," *J. Chem. Inf. Comput. Sci.*, vol. 36, no. 5, pp. 982-985, 1996.

[19] G. Young, L. Scardovi, and N. Leonard, "A new notion of effective resistance for directed graphs–Part I: Definition and properties," *IEEE Trans. Automat. Contr.*, vol. 61, no. 7, pp. 1727-1736, Jul. 2016.

[20] U. V. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, pp. 395-416, 2007.

[21] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14, pp. 281-297, 1967.

[22] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," *Advances in Neural Information Processing Systems*, vol. 2, pp. 849-856, 2002.

[23] P. Bonacich, "Power and centrality: A family of measures," *Am. J. Sociol.*, vol. 92, no. 5, pp. 1170-1182, 1987.

[24] A. Bavelas, "Communication patterns in task-oriented groups," *The Journal of the Acoustical Society of America*, pp. 725-730, 1950.

[25] G. L. G. Sleijpen and H. A. Van der Vorst, "A Jacobi–Davidson iteration method for linear eigenvalue problems," *SIAM Review*, vol. 42, no. 2, pp. 267-293, 2000.

[26] K. You, R. Tempo, and L. Qiu, "Distributed algorithms for computation of centrality measures in complex networks," 2016. arXiv:1507.01694v3 [cs.SY].

[27] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropol. Res.*, vol. 33, pp. 452-473, 1977.

[28] M. H. Wen, J. Xu, and V. O. Li, "Optimal multistage PMU placement for wide-area monitoring," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4134-4143, 2013.

[29] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowledge and Information Systems* vol. 42, no. 1, pp. 181-213, 2015.

[30] P-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Pearson Education India, 2006.