2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Community Detection Based on Local Topological Information in Power Grid

Zengqiang Chen, Zheng Xie
Tianjin Key Laboratory of Intelligent Robotics
College of Computer & Control Engineering, Nankai University
Tianjin, 300071, China
chenzq@nankai.edu.cn
xiezheng@mail.nanka.edu.cn

Qing Zhang
College of Science, Civil Aviation University of China
Tianjin 300300, China
qzhang@cauc.edu.cn

Abstract—**This paper proposes a novel algorithm based on local similarities to detect community structure in complex network. By analyzing the strengths and weaknesses of popular similarity indexes, a new index of node similarity is defined which can reflect closeness of local connections in networks as community does. And the similarity between a node and a community is defined by the sum of similarities between this node and all nodes within the community. Then networks can be partitioned without presetting the number of communities based on the assumption that nodes with highest similarities tend to merge together, additionally bridging nodes as byproducts. This method's effectiveness is confirmed by applying it to the IEEE 39-bus and 118-bus standard power grids. Influence of the bridging nodes in cascading failures is also discussed.**

Keywords—power grid; community detection; local similarity; bridging nodes

I. INTRODUCTION

With the rapid development of power industry, the scale of power grid is increasing and the architecture becomes more and more complex. This trend adds the difficulty of power system's analysis and control a lot. Development of complex network theory brings new perspectives on this system, such as Small-world, community structure, cascading failures and networks robustness [1].

Communities in networks, which are local structures of dense inner connections and sparse links between them, help to solve the complex problem. They partition the network of large scale into loosely coupled networks of much smaller scale easier to control. This divide-and-conquer strategy improves the robustness of the system. In power grid, community structure is often used in reactive power network partition [2], coherency-based dynamic equivalence [3], and power system restoration [4] and so on.

This work is partially supported by Natural Science Foundation of China Grants No. 61174094 and the Tianjin Nature Science Foundation Grant No.13JCYBJC17400.

There have been many methods developed by now to detect communities in networks, for example, GN [5] and MFC [6] as hierarchical clustering, CNM [7] and GA [8] as modularity optimization and many other algorithms [9]. Applying these network partition methods to power grids compensates some deficiencies of traditional method based on geographical and administrative area only and gets some available results. Ni et al.[2] pre-divide the network by voltage sensitivity matrix and merge the partition by modularity optimization. They get communities in the power grid that can ensure the control ability for generators and balance reactive power in place within the community. Lin et al.[4] apply GN algorithm by removing edges of biggest betweennesses in sequence in power grid. The hierarchical structure detected helps to restore the power subsystems quickly and parallel them in right order. Pan et al.[10] redefine the modular index introducing the reactive power balance degree of the network partitions. Then they adopt the improved Louvain hierarchical algorithm in power grids and find out pilot nodes of good observability and controllability sensitivity. Guo et al.[11] view a node's controllability sensitivity as its coordinate in space and in this way map the power grid to coordinate space. Then they apply traditional hierarchical agglomerative clustering to grids to acquire the community structure. Recently more attention is paid to power grid network partition to enhance the robustness. Pahwa et al.[12] propose a constraint programming formulation and modified Fast Greedy algorithm and the Bloom algorithm for optimal islanding in grids. Mehrjerdi et al.[13] apply spectral k-way partitioning formulation in grids and propose fuzzy secondary voltage to avoid propagation of disturbances between regions.

This paper redefines a local similarity index and each node selects the community to which it belongs without artificial hypothesis of the count of communities. Bridging nodes can be detected along the community detection process. Also bridging nodes' influence on cascading failure is examined.

The rest of the paper is organized as follows. Section II introduces the community detection process including similarity index definition, network partition and some supplements. Section III analyzes the computation cost and Section IV applies the algorithm in standard power grids and compares the influence of different indexes and clustering methods. Finally, Section V concludes this paper and puts forward a future research direction.

II. COMMUNITY DETECTION

The network partitioning method we proposed is based on the assumption that nodes with high similarities are more likely to form a community. Network partitioning, unlike hierarchical clustering including divisive and agglomerative approaches, improves the performance of partition by moving nodes among communities. The goal of exchanging nodes is to put similar nodes or closely connected nodes within one community as possible. The similarity index and the partition process are given as follows.

A. Similarity Index

Similarity is an index describing probabilities nodes belong to the some communities based on the assumption above. Given the network topological information by an adjacent matrix A, many indexes have been developed to measure the node similarity [14,15].

Common Neighbors index (CN) describes the similarity by the number of two nodes' common neighbors, namely

$$similarity(x,y)=\left|\Gamma(x)\cap\Gamma(y)\right| , \quad (1)$$

where $\Gamma(x)$ denotes the neighbor set of node x and $|X|$ the number of elements in set X. CN is simple and easy to calculate, but ignores further topology information about the network.

Katz index is developed as the ensemble of all paths between nodes, defined as

$$similarity=\beta A+\beta^2A^2+\beta^3A^3+\cdots=(I-\beta A)^{-1}-I \quad ,(2)$$

with parameter $\beta<1$ controlling different influence of paths. Note that β should be smaller than the reciprocal of the maximum of the eigenvalues of matrix A to ensure the convergence of Eq.(2) and often is set 0.01. This index considers the global information of the network. However, it's always difficult to get the global topology and consuming to calculate Katz index.

Then Local Paths index (LP) is proposed as a trade-off, namely

$$similarity=A^2+\beta A^3 . \quad (3)$$

It contains a little more network information and increases little computation cost. However, LP is proposed to predict links between disconnected nodes, which abandons direct connection as similarity between the endpoints. Take this example in Fig.1, with LP index the indirectly connected nodes D and B has similarity 1.01 bigger than that 0.03 of directly connected nodes D and E. It goes against intuition and does not fit the community detection process with similarity.

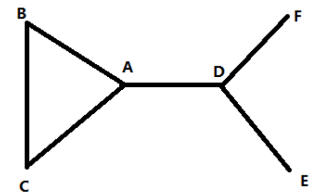

Fig.1 the example network

To solve these problems mentioned above, this paper combines advantages of LP and Katz index and proposes a new local similarity index (LS)

$$similarity=A+\beta A^2+\beta^2A^3 . \quad (4)$$

Parameter β is set a little bigger than that of the two indexes above, as the LS index should involve more local topology information which is beneficial for community detection. It can be adjusted appropriately to fit the community structure best. To fit the grids, it is set 0.4 in this paper. The highest paths order is set 3 here as LP does. It calculates the times node X can reach node Y within three steps, which indirectly reflects connections between a node's neighbors. This metric reflects local close connections well as community does.

B. Network Partition

K-means and K-medoids are popular clustering methods [16]. Both of them proceed by selecting k initial cluster centers randomly and assign each node to its nearest cluster centers to get a community structure. Then cluster centers are updated according to the communities. This process is repeated until it converges when there's no further change in each node's assignment to communities. The difference between these two methods is cluster centers' updating. K-means clustering calculates the means of nodes coordinates as cluster centers, while K-medoids clustering selects medoids, which have biggest similarities with other nodes inner clusters, as cluster centers.

However, the above methods both fails to detect community in networks. K-means is applicable for vectors in Euclidean Space but not for relational data in networks described as adjacent matrix. K-medoids clustering is suitable for data describing both attribute and relations but only considers nodes similarity with the medoids, neglecting connections between non-medoids nodes, which doesn't fit the community structure in network well.

Both of the two methods above update cluster centers and calculate similarities between nodes and centers as similarities between nodes and communities, to assign nodes to communities. This paper designs a novel clustering method improving the above process. Instead of updating cluster centers, we calculates the sum of similarities between a node and all nodes in a community as the similarity between the node and the community, namely the similarity between node i and all nodes in community C_j

$$similarity(i,C_j)=\sum_{p\in C_j} similarity(i,p) \cdot \quad (5)$$

The similarity between each node and each community can be viewed as a node's grade of membership in a community. So we assign node i to the community C^i which has the biggest similarity with it, namely

$$C^i = \arg \max_{C_j,(j=1,2\cdots k)} similarity(i, C_j) . \quad (6)$$

To illustrate this algorithm, the partition steps are given as follows:

1) Randomly partition all the nodes in network G into k initial communities $C_1, C_2, \cdots C_k$, and k is a parameter to adjust the number of initial communities.

2) For each node in network G, take following steps.

3) For node i, calculate by (5) the similarity index $similarity(i, C_j)$ between node i and each community $C_j, (j = 1, 2 \cdots k)$ and then assign node i belongs to the community C^i by (6), which has the biggest similarity with it.

4) The algorithm stops if it reaches the given number of iterations or assignments of each node don't change any more. Otherwise go to step 2).

C. Supplements of the algorithm

With the assignments of nodes changing, the number of communities may decrease as communities merge together automatically. So the initial count of communities can be set as the upper bound of possible amount of communities, then natural community structure can be acquired without human intervention.

Additionally, nodes' membership information can be analyzed to get more characteristics about node influence in the community structure. For example, nodes' membership can help to find bridging nodes which are closely connected to more than one community. To describe bridging nodes' community uncertainty, an index called bridging property is defined here, namely

$$bridging(i) = \sum_{j=1,2\cdots k} \left(max - similarity(i, C_j) \right)^2 \quad (7)$$

with intermediate variable $max = \max_{j=1,2\cdots k} similarity(i, C_j)$.

The greater this index is, the more connections the node has with other communities and the more likely it is a bridging node.

Since nodes are assigned randomly to form the initial community structure and different initial conditions may lead to different final partitions, it is reasonable to take the algorithm many times and select the community structure that fit the network best. Here the modularity index proposed by Newman et al. is used to evaluate the performance of the community structure detected by our method. Its modularity is proposed as the following equation

$$Q = \frac{1}{2M} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2M} \right) \delta\left(C_i, C_j \right) \quad (8)$$

where A_{ij} is element of adjacent matrix A, M is the counts of edges in the network, k_i is degree of node i, C_i is the community which node i belongs to and $\delta\left(C_i, C_j\right)$ equals 1 if node i and j are in the same community and 0 otherwise. This index will be calculated in the experiment to monitor the community detection process.

III. ALGORITHM COST ANALYSIS

A. Calculation of similarity matrix

According to [15], reasonable choice of the paths order partially depends on the average shortest distance of the network. The features of small world and six degrees of complex network taken into consideration, high paths order just increases computation much but influences similarity index little. This paper sets 3 as the highest paths order and the corresponding time complexity to calculate the similarity matrix is $O(N < k >^3)$. On the other hand, as the computing process of similarities between a node and the others is independent, parallel computing may be taken to decrease the time cost.

B. Cost of network partition

For node similarity matrix S, the ith row denotes the similarity between node i and the other nodes in the network, and element $S_{ij} = 0$ means node i and j can't be reachable within three steps. For description convenience, nodes reachable within three steps from node x are called near neighbors of x below. As node's neighbors are likely to be linked, the count of each node's near neighbors gets an average magnitude $< k >^3$. It is efficient to record the near neighbors of a node using a linked list since complex networks are often sparse. Each node's similarities with different communities can be calculated when traversing its linked list and then used to partition the network. As a result, this partition process's time complexity is $O(tN < k >^3)$ with iteration times t needed. Similar to the node similarity calculation, each node's assignment is also independent and can be parallel computed.

To sum up, the whole time complexity of this algorithm is $O((t+1)N < k >^3)$ and use of parallel computing could help decrease the time cost.

IV. EXPERIMENTS AND ANALYSIS

Here we apply the algorithm in IEEE 39-bus and 118-bus standard power grids. Comparison of the influence of different indexes and clustering method is also discussed. Neglecting complex electrical properties, the generators and loads are viewed as nodes and lines as links between nodes indiscriminately. Then the power grid is simplified as an undirected and unweighted network. The links between nodes can be viewed as similarity, as they reflect the interplay of nodes' voltage fluctuation.

A. Experiment on 39-bus standard power grid

In IEEE 39-bus standard power grid, there are 10 generators and 29 load nodes. To ensure that each community has at least one generator, the upper bound of the count of communities is 10. Then pre-divide the grid into 10 initial communities with one generator in each community, each load node assigned into the same community as the generator which has the most similarity with it is in. In this way, the initial condition is fixed and so will the final community structure detected be. The change of modularity in community detection process is shown in Fig.2.

As can be seen, the initial community structure we pre-divide has modularity 0.5151. Then the communities automatically exchange nodes and sometimes merge together according to the local similarity index. The modularity shows an upward tendency which means the partition tends to make an obvious community structure. The iteration stops in only 3 steps and the algorithm finally partitions the 34 nodes in this power grid into 5 communities with modularity 0.6092. The community structure is shown in Fig.3.

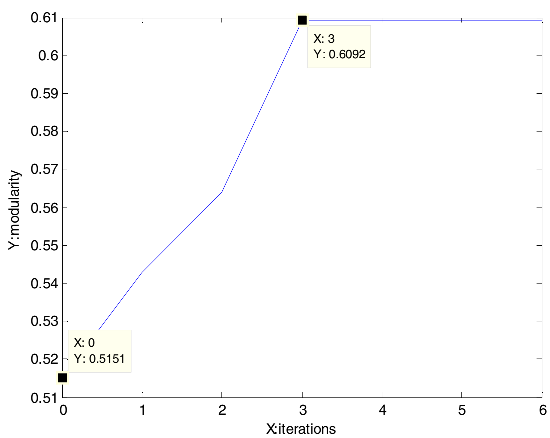

Fig.2. Change of modularity along with iterations of community detection

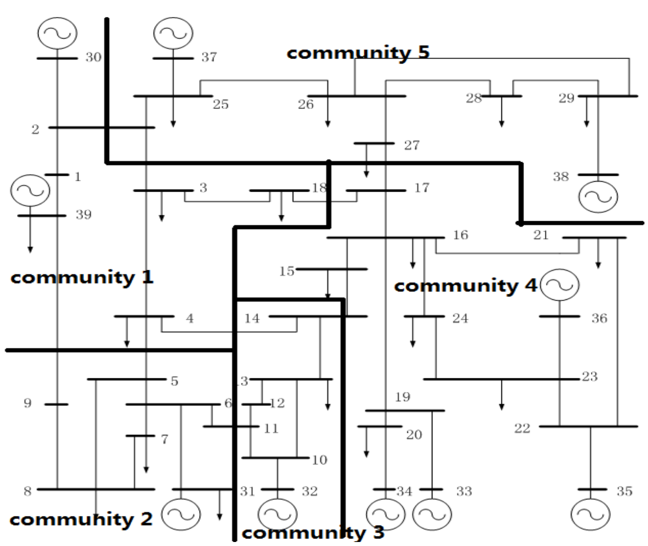

Fig.3.Five communities discovered in IEEE 39-bus standard power grid. Nodes are separated by bold lines.

Furthermore, to estimate the influence of bridging nodes discovered in this power grid community detection, a simplified cascade failure model is applied here. As loads transmit through shortest electrical paths, here suggest each node's initial load $L(0)$ is given as its betweenness in the initial network and capacity C is given as $C=(1+\alpha)L(0)$ in [17]. Constant α is called the tolerance parameter and set 0.2 in this simulation. After removing nodes, loads are redistributed that a node's load changes to its betweenness in the changed network.

Nodes chosen randomly, with biggest degrees, with biggest betweennesses or with biggest bridging properties are removed respectively to trigger the cascading failure. The damage caused by the cascade is quantified by the size of the largest connected component in the network left. Result of the simulation is showed (see Fig.4).

As Fig.4.(a) presents, when intentionally attacking the power grid, removing nodes with top degrees at a time may cause the most serious cascading failure among the four

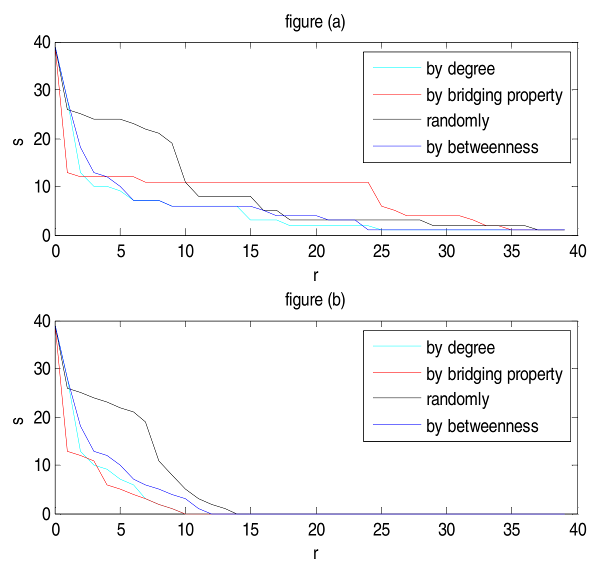

Fig.4. Cascading failure in IEEE 39-bus standard power grid, as triggered by removing nodes in different ways. Parameter r is the count of removing nodes and s is the size of the largest connected component in the left network. Nodes of corresponding counts are removed all at once in figure (a) and one by one in figure (b).

strategies and randomly may be the least. However, when a node of top index is removed, nodes of near top indexes may be overloaded and removed passively as cascading failure. At next step, if these nodes are removed once together, the left network won't change any more and the subsequent nodes' damage effect are covered by the previous ones'. That's why in Fig.4.(a) the size may keep stable when the count of removing top nodes increases. If the nodes left in the cascading are removed in order one by one, as the below Fig.4.(b) shows, removing the nodes sorted by bridging property will cause the most serious cascading failure. Degree is also a good index of nodes' influence in the network, as it causes serious cascading failure in both cases and requires less computation cost.

To see more details, each node is removed to check other failure nodes triggered, listed in table 1. The rest nodes removed not on the list mean that they cause no cascading failure. The top five bridging nodes are of indexes 4, 14, 18, 17 and 27. Each of them removed can cause cascading failure of many other nodes. On the other hand, there are overlaps between the damage effects of these nodes, which is consistent with above analysis.

TABLE 1. FAILURE TRIGGERED BY EACH NODE'S REMOVAL

Node removed	Failure nodes triggered
1	5,8,9,10,11,12,13
2	5,8,9,12,15,17,18,27,39
3	1,7,8,9,11,12,15,17,25,26,27,39
4	1,6,7,8,9,10,11,12,13,17,18,27,39
5	1,7,9,10,11,12,13,39
6	12,13
8	1,39
10	12
13	4,5,6,11,1,9,17,18,39
14	3,4,5,6,7,11,12,17,18,39
15	3,4,17,18,39,11
17	2,3,4,9,14,15,25
18	15,25,27
21	23,24
24	21,22
25	17,18,27,15
26	18,4,14,15,1,7,8,9,11,39
27	2,3,18,25,8,9

B. Experiment on 118-bus standard power grid

This community detection process is also applied in the IEEE 118-bus standard power grid, in which there are 54 generators and 64 load nodes. The change of modularity in community detection process is shown in Fig.5.

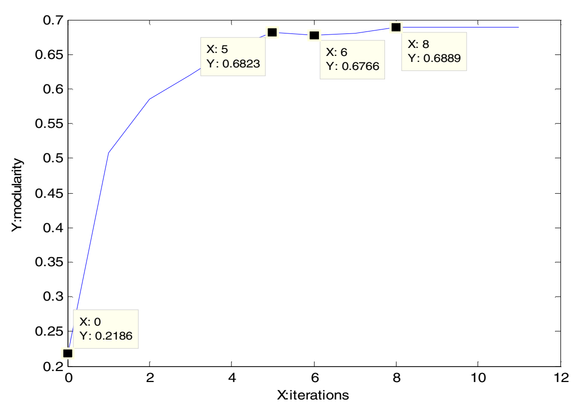

Fig.5. Change of modularity along with iterations of community detection

We can see that the initial community structure has modularity 0.2186. Then the modularity is generally in an upward trend but has a little fluctuation during the iterations. As Fig.5 shows, the modularity is 0.6823 in step 5, 0.6766 in step 6 and 0.6889 in step 8. This flexibility is a big advantage combining with other modularity optimization methods. As many other methods are based on greedy algorithm, the modularity is monotone increasing and easy to trap into the local optimum. The fluctuation above means that the algorithm here may avoid this defect and get better community structure.

As communities exchange nodes and merge together, this network is naturally partitioned into 8 communities in 8 iterations. The community structure detected is shown in Fig.6.

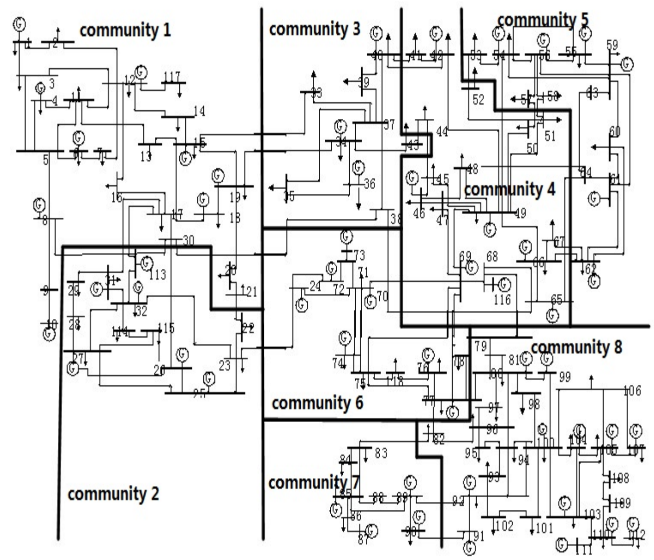

Fig.6. Four communities discovered in IEEE 118-bus standard power grid. Nodes are separated by bold lines.

Intuitively, nodes in the same community distribute near each other and have close connections. On the other hand, the community structure has modularity 0.6889 a little bigger than that of corresponding unweighted network in [18], which means the algorithm in this paper can decouple this power grid better. Details are showed in the following part C.

C. Comparison of different indexes and clustering methods

To compare different influence of similarity indexes on community detection, experiments are done in above grids. Also modularity is calculated here to evaluate the performance of clustering with different similarity indexes. Results are showed in table 2 as following. Pan's clustering method [18] in corresponding unweighted grids is called Pan for short.

TABLE 2. SIMILARITY INDEXESES' AND CLUSTERING METHODS' INFLUENCE ON COMMUNITY DETECTION

	39-bus grid	118-bus grid	300-bus grid
CN	-0.0666	0.0805	0.2400
Katz	0.5801	0.6939	0.7626
LP	0.2968	0.2231	0.4554
LS	0.6092	0.6889	0.7831
Pan	0.6226	0.6875	0.7442

As can be seen from table 2, communities detected with CN and LP indexes have smaller modularities in both 39 and 118-bus grids. It verifies the previous theoretical analysis that CN and LP indexes, neglecting direct connection as similarity, don't fit the community detection well. Katz and LS indexes help to find more suitable community structure here. LS performs better than Katz index in 39-bus grid and reverses in 118-bus grid. Katz index with global topology considers more comprehensive information than LS does, which may be useful to describe similarities between nodes. On the other hand, Katz involves much irrelevant information contributing little to detect local community structure. LS needs less information than Katz does, which makes it available and easy to calculate and help to discover good community structure as Katz does, even better. Compared with Pan's clustering method in corresponding unweighted grids in [18], the method proposed in this paper partitions 118 and 300 bus grids with higher modularity.

V. CONCLUSION AND FUTURE WORK

Based on study of previous method on community detection, this paper proposes a new similarity index reflecting local density in networks. With this index, partition of the networks can be acquired as each node is assigned to the community which has biggest similarity with the node. Along the process of community detection, the bridging nodes can also be discovered and their influence on cascading failure is examined. However, the algorithm here is used in highly simplified network and may have problems when applying in real power grid. More electrical characteristics should be taken into consideration in future work.

REFERENCE

[1] Wang Xiaofan, Li Xiang, Chen Guanrong. Network Science: An Introduction [M]. Beijing, Higher Education Press. 2012.

[2] Ni Xiangping, Ruan Qiantu, Mei Shengwei, He Guangyu. A new network partitioning algorithm based on complex network theory and its application in Shanghai power grid [J]. Power System Technology, 2007, 31(9): 6-12.

[3] Ni Xiangping, Mei Shengwei. Coherency-based Dynamic Equivalence Algorithm with Community Structure of Complex Network Theory [J]. Automation of Electric Power Systems, 2008, 32(7): 10-13.

[4] Lin Zhenzhi, Wen Fushuan, Zhou Hao. A new algorithm for restoration subsystem division based on community structure of complex network theory [J]. Automation of Electric Power Systems, 2009, 12: 004

[5] Girvan Michelle, Newman Mark E.J. Community structure in social and biological networks [J]. Proceedings of the National Academy of Sciences. 2002, 99(12): 7821-7826

[6] Gary William Flake, Steve Lawrence, C.Lee Giles, Frans M.Coetzee. Self-organization and identification of web communities [J]. Computer. 2002, 35(3): 66-70.

[7] Clauset Aaron, Newman Mark E.J , Moore Cristopher . Finding community structure in very large networks[J]. Physical review E. 2004 70(6) : 06611

[8] Guimera Roger, Luis A. Nunes Amaral. Functional cartography of complex metabolic networks[J]. Nature .2005,433(7028) : 895-900.

[9] Fortunato Santo. Community detection in graphs[J]. Physics Reports. 2010, 486(3) : 75-174.

[10] Pan Gaofeng, Wang Xinghua, Peng Xiangang, Wu Xiaomei. Improved network partition and pilot node selection method for reactive power/voltage control based on community detection algorithm [J]. Power System Protection and Control. 2013,41 (22):32-37.

[11] Guo Qinglai, Sun Hongbin, Zhang Boming, Wu Wenchuan. Power network partitioning based on clustering analysis in mvar control space [J]. Automation of Electric Power Systems.2005, 29(10): 36-40.

[12] Pahwa S, Youssef M, Schumm P, C.Scoglio, N.Schulz. Optimal intentional islanding to enhance the robustness of power grid networks[J]. Physica A: Statistical Mechanics and its Applications, 2013, 392(17): 3741-3754.

[13] Mehrjerdi H, Lefebvre S, Saad M, et al. A decentralized control of partitioned power networks for voltage regulation and prevention against disturbance propagation[J]. IEEE Transactions on Power Systems, 2013, 28(2): 1461-1469.

[14] Lü Linyuan, Zhou Tao. Link prediction in complex networks: A survey[J]. Physica A: Statistical Mechanics and its Applications. 2011,390(6):1150-1170.

[15] Lü Linyuan, Jin Ci-Hang , Zhou Tao .Similarity index based on local paths for link prediction of complex networks[J]. Physical Review E .2009,80(4): 046122.

[16] Han J, Kamber M, Pei J. Data mining: concepts and techniques[M]. Morgan kaufmann, 2006.

[17] Motter A E, Lai Y C. Cascade-based attacks on complex networks[J]. Physical Review E, 2002, 66(6): 065102

[18] Pan Gaofeng, Wang Xinghua, Peng Xiangang, Wu Xiaomei. Study of power grid partition identification method based on community structure detection [J]. Power System Protection and Control. 2013,41 (13):116-121