Physica A 392 (2013) 2265–2277

Contents lists available at SciVerse ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

# Minimum spanning trees for community detection

Jianshe Wu [a], Xiaoxiao Li [a,*], Licheng Jiao [a], Xiaohua Wang [b], Bo Sun [c]

[a] *Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, PR China*

[b] *Aeronautical Computing Technique Research Institute, Xi'an 710068, PR China*

[c] *ZTE corporation, PR China*

ARTICLE INFO

*Article history:*
Received 17 July 2012
Received in revised form 12 December 2012
Available online 9 January 2013

*Keywords:*
Community structure
Minimum spanning tree
Distance matrix
Complex network

ABSTRACT

A simple deterministic algorithm for community detection is provided by using two rounds of minimum spanning trees. By comparing the first round minimum spanning tree (1st-MST) with the second round spanning tree (2nd-MST) of the network, communities are detected and their overlapping nodes are also identified. To generate the two MSTs, a distance matrix is defined and computed from the adjacent matrix of the network. Compared with the resistance matrix or the communicability matrix used in community detection in the literature, the proposed distance matrix is very simple in computation. The proposed algorithm is tested on real world social networks, graphs which are failed by the modularity maximization, and the LFR benchmark graphs for community detection.

## 1. Introduction

Community structure is popular in real world networks [1–8]. It is described as the edges among the nodes in the same community are denser than those among the nodes in different communities [9,10]. A community often corresponds to a functional unit in the network, thus the detection of communities has important practical significances. Tremendous efforts have been done in this direction in recent years [11]. The most commonly used algorithms are designed to maximize the value of the function known as "modularity" defined by Newman and Girvan [12–18]. A recent review of the community detection is given in Ref. [11]. However, modularity maximization is found to be affected by the resolution limit [19–22]. There is an intrinsic scale depending on the number of links of the network, communities smaller than that scale in size may not be resolved, even if they were complete graphs connected by single bridges [19].

A possible scheme for the resolution limit problem is that defining a distance matrix for the network and then detecting the communities through a clustering algorithm, such as $k$-means [23]. Alves presented a method to reveal the hierarchical structure by computing the resistance matrix of the network through using electric network theory [24]. In fact, the resistance matrix is a distance matrix. Estrada and Hatano defined the concept of communicability of complex networks [25], considering the communicability matrix as input of the $k$-means clustering algorithm, communities can be detected [26]. However, computing the resistance matrix needs to compute the eigenvalues and eigenvectors of the Laplacian matrix of the network [24]; computing the communicability matrix needs to compute the eigenvalues and eigenvectors of the adjacency matrix of the network [25]. It is well known that the time complexity of computing eigenvalues and eigenvectors is $O(N^3)$, where $N$ is the number of nodes. This made a high computational cost for large size networks in spite of the clustering algorithm.

In fact, the concept of distance between a pair of nodes is similar to that of edge betweenness defined by Girvan and Newman [1]. The edge betweenness of an edge is defined as the number of shortest paths between pairs of vertices

[*] Corresponding author. Tel.: +86 29 88202279.
*E-mail addresses:* jshwu@mail.xidian.edu.cn (J. Wu), luomosajia@gmail.com (X. Li).

0378-4371/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
doi:10.1016/j.physa.2013.01.015

that run along it. Thus, the edges among communities always have high edge betweenness. By repeatedly calculating the betweenness for all edges in the network and removing the edge with the highest betweenness, until no edge remains, the GN algorithm can obtain the hierarchical tree of a network. The GN algorithm avoids many of the shortcomings of traditional methods [27], but it is computationally expensive. The time complexity of calculating betweenness for all $M$ edges in a graph of $N$ nodes is $O(MN)$, the GN algorithm running in worst-case time is $O(M^2N)$ due to the repeated calculations for the removal of each edge. To overcome the problem of GN algorithm, Radicchi et al. presented an alternative algorithm by calculating only local quantities instead of the edge betweenness, named edge-clustering coefficient [27]. Compared with the GN algorithm, the accuracy of the obtained algorithm is similar, but the computational speed is much higher. The time complexity of Radicchi et al.'s algorithm is $O(M^2)$ [27]. Radicchi et al. give explicit definition for community in both strong and weak senses also.

Similar to the principle that Radicchi et al. improve the GN algorithm using only local information, the distance matrix can be calculated by using only local information. In fact, the distance between most pairs of nodes is not needed to be known for community detection (see Section 2).

Inspired by the seminal works by Girvan and Newman [1], and Radicchi et al. [27], in this paper, a simple distance matrix is defined firstly with time complexity $O(N)$, and then we generate the two rounds minimum spanning tree (MST) of the network. The communities are detected by using the two rounds MST. MST has been applied for data clustering [28–30]. In graph theory, the first round (1st-) MST of a graph $G(V, E)$ is a tree-like (acyclic) subgraph of $G$ that includes all the nodes in $V$ and the total distance is minimum. The second round (2nd-) MST is a tree-like subgraph without repeated edge from the 1st-MST, that the total distance is minimum. It is observed that two types of the edges in the MSTs are usually larger than others: one is those edges traversing between nodes in different communities and the other is those in the bottom (end branches) of the MST. Thus the edges in the MSTs traversing among communities can be easily identified. By removing an edge in the 1st-MST or 2nd-MST which traverses between nodes in different communities, the network is separated into two parts. From all the binary separations obtained with the two rounds MST, communities can be detected. The inconsistent nodes of two separations from the 1st-MST and the 2nd-MST are the overlapping nodes of communities.

This paper is arranged as follows. In Section 2, the distance matrix is defined. In Section 3, detailed algorithm is provided by using the two rounds MST. In Section 4, the proposed algorithm is tested on networks which are failed by modularity maximization due to the resolution limit, real world social networks, and the networks of LFR benchmark for community detection [31].

## 2. Establishment of the distance matrix

Similar as the distance matrix in clustering of data mining [23], where the distance between two data is a measure by which they should belong to the same cluster or not, in the distance matrix of a complex network the distance between two nodes should be a measure by which they belong to the same community or not. Obviously, the adjoint matrix of the network cannot be a distance matrix. If there is no path between two nodes, then the corresponding distance is infinite. In a connected graph, there is at least one path between each pair of nodes, thus the distance is finite.

### 2.1. Distance matrix from heuristics

In the MST, a node connects to its neighbors with the smallest distance [32,33]. For a pair of nodes ($v_i$ and $v_j$) in the network without a path less than or equal to 3 hops, it is almost impossible that $e_{ij}$ is an edge in the MST, thus the distance $d_{ij}$ is not needed to compute.

The distance between a pair of nodes is computed by the following two rules:

(1) For all the paths between $v_i$ and $v_j$, if the number of hops is larger than 3, then $d_{ij}$= $\eta$ and $\eta > 3$. In this case, we do not compute $d_{ij}$ and set the value of $d_{ij}$ to be $\eta$ directly. It is not needed to compute the distance of a node to all the rest nodes but only a few of its nearest neighbors. This dramatically decreases the amount of computation for the distance matrix.

(2) If there are several paths between $v_i$ and $v_j$ in which the number of hops is less than or equal to 3, then $d_{ij}$ is computed by (1).

$$d_{ij} = \frac{1}{n_a 1/1 + n_b 1/2 + n_c 1/3}. \tag{1}$$

In this case, there are three kinds of paths:

$$\begin{cases} a & \text{the path which is 1 hop.} \\ b & \text{the path which is 2 hops.} \\ c & \text{the path which is 3 hops.} \end{cases}$$

In (1), $n_a$, $n_b$, and $n_c$ are the number of paths with 1, 2, and 3 hops, respectively. In computing $n_a$, $n_b$, and $n_c$, two paths should be completely different, in other words, they do not share a common edge in the entire path. The distance of (1) is defined by heuristics also.

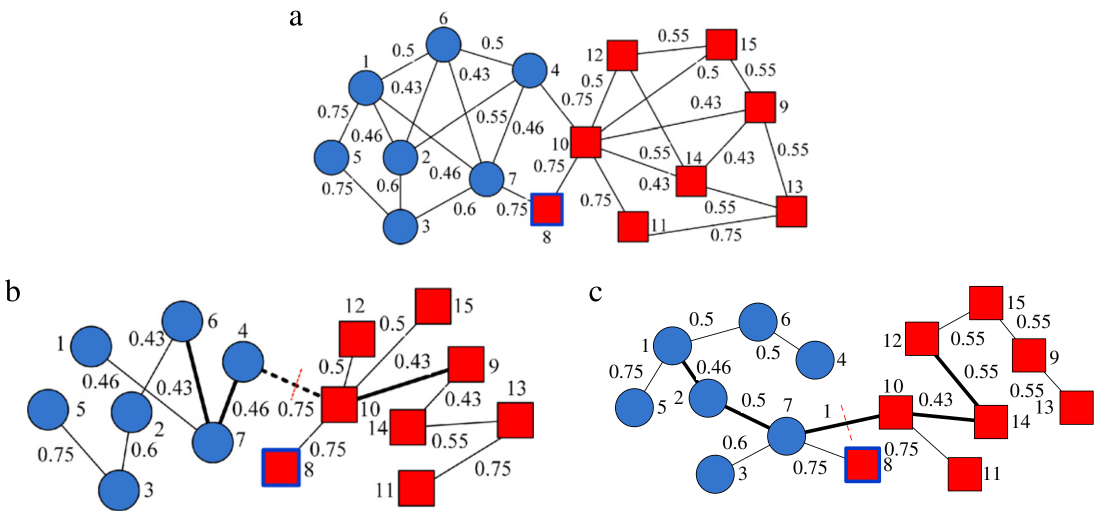

**Fig. 1.** A network and its two rounds MST. (a) The graph of the network. (b) The 1st-MST. (c) The 2nd-MST.

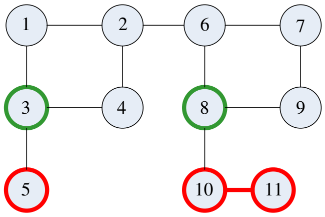

**Fig. 2.** A network with two leaves: $v_5$ and $\{v_{10}, v_{11}\}$.

**Table 1**
Paths for computing $d_{12}$.

| The kind of the paths | Paths | Number of hops |
|---|---|---|
| $a(n_a = 1)$ | $v_1$–$v_2$ | 1 |
| $b(n_b = 1)$ | $v_1$–$v_6$–$v_2$ | 2 |
| | $v_1$–$v_6$–$v_4$–$v_2$ (pick out) | 3 |
| | $v_1$–$v_5$–$v_3$–$v_2$ | 3 |
| $c(n_c = 2)$ | $v_1$–$v_7$–$v_3$–$v_2$ (pick out) | 3 |
| | $v_1$–$v_7$–$v_4$–$v_2$ | 3 |
| | $v_1$–$v_7$–$v_6$–$v_2$ (pick out) | 3 |

Here, see Fig. 1(a) for an example to illustrate the process of computing a distance matrix. The distance between $v_1$ and $v_{11}$ need not to be computed because the shortest path is $v_1$–$v_6$–$v_4$–$v_{10}$–$v_{11}$, which is a 4-hop path. There are 7 paths between $v_1$ and $v_2$ with 1, 2, or 3 hops, thus $d_{12}$ needs to be computed. The details of computing $d_{12}$ are shown in Table 1.

Three paths are picked out because they repeatedly use an edge which has been used by another path. Then from (1), $d_{12} = \frac{1}{1\times1+1\times1/2+2\times1/3} = 0.4615$. By this way, the distance matrix $D = [d_{ij}]$ can be obtained.

The definition of distance matrix seems very simple, in Section 4 we will show that it is enough to reveal the edges which traverse between communities.

### 2.2. Distance matrix without leaf nodes

Leaves structure of a graph has been introduced in Ref. [24]. A leaf is a tree-like topology in graph, removing leaves can simplify the detection of communities. Fig. 2 shows a graph with two leaves. Node $v_5$ connects to $v_3$; $v_{10}$ and $v_{11}$ connect to $v_8$. So $v_5$ can be assigned into the community of $v_3$; $v_{10}$ and $v_{11}$ assigned into the community of $v_8$. Thus the leaf nodes can be removed before computing the distance matrix $D$ and this is simple by using the adjacency matrix $A$ of the network.

The leaf nodes have two kinds: one is that there is only one edge connected with it, examples are $v_5$ and $v_{11}$ in Fig. 2; the other is that there are two or more edges, an example is $v_{10}$ in Fig. 2. The second kind will be changed to the first kind after removing the leaf nodes connected to it. With this observation, the leaf nodes can be removed by repeatedly deleting the rows and columns in $A$ corresponding to the nodes with only one edge, until all nodes in $A$ has two or more edges.

For the graph shown in Fig. 2, the adjacency matrix is

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

It can be seen that there is only one non-zero element in the 5-th row and 5-th column respectively, which indicates $v_5$ is a node with only one edge.

In the denominator of (1), the fact 1/1, 1/2, and 1/3 correspond to the contributions of the 1, 2, and 3-hop paths to the distance, which may not be suitable in some cases. In our experiments based on (1), it is found that if increasing edges exist in the original connected graph to the two rounds MST, better result may be achieved. Thus, a parameter $\delta$ is introduced to tune the distance of edges in the original graph. Then the distance matrix is modified as:

$$D_M = D_R \circ (E - A_R \delta), \quad 0 \le \delta < 1, \tag{2}$$

where $D \circ B = C$ denotes the Hadamard product of matrices $D$ and $B$ [34], which means $c_{ij} = d_{ij} b_{ij}$ for all elements of the matrix $C$. $E$ is a matrix with all elements being one, $A_R(D_R)$ is the adjacency (distance) matrix of the network after removing leaf nodes.

## 3. Algorithm for community detection

After the distance matrix of a network is established by (2), its MST can be found by using an algorithm in the literature [32,33]. In MST, a node connects with its nearest neighbor by distance. Since MST has no loop, by removing any edge in the MST, all the nodes in the network are separated into two parts. Of course, it should be separated at the edges which traverse between different communities.

### *3.1. Two rounds MST*

In mathematics, the two rounds MST are denoted as follows:

$$T_1 = f_{mst}(V, E) \tag{3}$$

$$T_2 = f_{mst}(V, E - T_1) \tag{4}$$

where $f_{mst} : (V, E) \to T$ is a function (operation or algorithm) to create a MST from a graph. In this paper, the Kruskal's algorithm [32] is used in the experiments. See the network shown in Fig. 1(a) again, after the matrix $D_M$ is obtained by (2) when $\delta = 0$, the 1st- and 2nd-MST are obtained and shown in Fig. 1(b) and Fig. 1(c) respectively.

As mentioned above, all the nodes of a graph are separated into two sub-sets by removing an edge in the MST, which can be presented as $(s_1, s_2)$. In the 1st-MST shown in Fig. 1(b), the nodes is separated into two sub-sets by removing $e_{4-10}$ drawn by dotted line, which is one of the edges with the largest distance, where $s_1 = \{1, 2, 3, 4, 5, 6, 7\}$ and $s_2 = \{8, 9, 10, 11, 12, 13, 14, 15\}$. In the 2nd-MST shown in Fig. 1(c), by removing $e_{7-10}$ drawn by dotted line, which is the edge with the largest distance, the obtained two sub-sets are: $s_1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $s_2 = \{9, 10, 11, 12, 13, 14, 15\}$. By comparing the two separations obtained from the two MSTs, it is seen that only $v_8$ is different in the two separations, in fact, $v_8$ is an overlapping node of the two communities.

In general speaking, a graph with $N$ nodes has $N - 1$ edges in its MSTs, thus $N - 1$ separations can be obtained from the 1st-MST or 2nd-MST. Two sequences of separations can be obtained from the two rounds MST respectively as follows:

$$S^{T_1} = \left\{ \left(s_{11}^{T_1}, s_{12}^{T_1}\right), \left(s_{21}^{T_1}, s_{22}^{T_1}\right), \ldots, \left(s_{(N-1)1}^{T_1}, s_{(N-1)2}^{T_1}\right) \right\}, \tag{5}$$

$$S^{T_2} = \left\{ \left(s_{11}^{T_2}, s_{12}^{T_2}\right), \left(s_{21}^{T_2}, s_{22}^{T_2}\right), \ldots, \left(s_{(N-1)1}^{T_2}, s_{(N-1)2}^{T_2}\right) \right\}, \tag{6}$$

where $\left(s_{i1}^{T_j}, s_{i2}^{T_j}\right)$ denote the separations resulted from removing the $i$-th edge in $T_j$, $s_{i1}^{T_j}$ and $s_{i2}^{T_j}$ are sub-sets of nodes, $s_{i1}^{T_j} \cup s_{i2}^{T_j} = V$, $1 \le i \le N - 1$, and $j \in \{1, 2\}$. A problem is that there are so many separations in (5) and (6), the concern is which one is coming from removing an edge traversing between communities.

It is observed that the edges with larger distance always distribute in the boundary of a community, corresponding to low level edges (end branches) in the MSTs or edges traversing between communities. By eliminating the low level edges in MSTs, the edges traversing between communities can be evaluated by their distance. In fact, if in a separation $(s_1, s_2)$, $|s_1| = 1$ and $|s_2| = N - 1$, then it is infeasible. It is found that a community usually contains more than 3 nodes in actual networks. In another word, a sub-set with less than 3 nodes cannot be a community. Thus (7) can be applied to eliminate the separations in (5) and (6), each of them comes from removing a low level edge in the MSTs.

$$\left|\left(\left|s_{ip}^{T_j}\right| + n_{ip}\right)\right| \le \varepsilon. \tag{7}$$

The value of $\varepsilon$ is assigned according to priori information of the network, which can be set in the range [3,5] in general simulation. $n_{ip}$ is the number of nodes in the leaves connected to the nodes in $s_{ip}^{T_j}$. Since the MSTs are tree-like topologies, many of the separations can be eliminated by (7). For example in both Fig. 1(b) and (c), if $\varepsilon = 2$, then 8 of the 14 separations will be eliminated.

By eliminating some of the separations by (7), suppose there are $L_1$ and $M_1$ separations in (5) and (6) respectively, they are rearranged and denoted as (8) and (9) respectively:

$$S_\varepsilon^{T_1} = \left\{\left(s_{11}^{T_1}, s_{12}^{T_1}\right), \left(s_{21}^{T_1}, s_{22}^{T_1}\right), \dots, \left(s_{L_11}^{T_1}, s_{L_12}^{T_1}\right)\right\}, \quad L_1 \le N - 1, \tag{8}$$

$$S_\varepsilon^{T_2} = \left\{\left(s_{11}^{T_2}, s_{12}^{T_2}\right), \left(s_{21}^{T_2}, s_{22}^{T_2}\right), \dots, \left(s_{M_11}^{T_2}, s_{M_12}^{T_2}\right)\right\}, \quad M_1 \le N - 1, \tag{9}$$

where $\left|s_{11}^{T_1}\right| \le \left|s_{21}^{T_1}\right| \le \dots \le \left|s_{L_11}^{T_1}\right|$ and $\left|s_{11}^{T_2}\right| \le \left|s_{21}^{T_2}\right| \le \dots \le \left|s_{M_11}^{T_2}\right|$.

As observed in the network shown in Fig. 1, only the separation by removing the edge with the largest distance in the two MSTs is used for community detection. By this observation, we can only preserve those separations by removing an edge of larger distance. A threshold has to be introduced for this purpose. Here, we use the average value of the distance of the edges corresponding to the separations in (8) and (9).

$$d_{avg}^{T_1} = \frac{1}{L_1} \sum_{e_l \in T_1} e_l, \tag{10}$$

$$d_{avg}^{T_2} = \frac{1}{M_1} \sum_{e_l \in T_2} e_l. \tag{11}$$

In the separations in (8) and (9), eliminate those come from removing an edge whose distance is less than $d_{avg}^{T_1}$ and $d_{avg}^{T_2}$ respectively, the rest separations are denoted as follows.

$$S_d^{T_1} = \left\{\left(s_{11}^{T_1}, s_{12}^{T_1}\right), \left(s_{21}^{T_1}, s_{22}^{T_1}\right), \dots, \left(s_{L_21}^{T_1}, s_{L_22}^{T_1}\right)\right\}, \quad L_2 < L_1, \tag{12}$$

$$S_d^{T_2} = \left\{\left(s_{11}^{T_2}, s_{12}^{T_2}\right), \left(s_{21}^{T_2}, s_{22}^{T_2}\right), \dots, \left(s_{M_21}^{T_2}, s_{M_22}^{T_2}\right)\right\}, \quad M_2 < M_1. \tag{13}$$

By this operation, about a half of the separations in (8) and (9) are eliminated. In the following, a separation coming from removing an edge $e_{ij}$ in the 1st-MST or 2nd-MST may be denoted as separation $e_{ij}$ in 1st-MST or 2nd-MST.

With these two sequences, the communities can be detected by comparing each separation in (12) with those in (13). First of all, let us consider an extreme network with two communities and all the separations in (12) and (13) come from removing an edge traversing between the two communities. They should satisfy

$$\left|s_{i1}^{T_1} - s_{j1}^{T_2}\right| = 0 \quad \text{or} \quad \left|s_{i1}^{T_1} - s_{j2}^{T_2}\right| = 0, \quad i = 1, \dots, L_2, j = 1, \dots, M_2, \tag{14}$$

where $\left|s_{i1}^{T_1} - s_{j1}^{T_2}\right|$ is the number of inconsistent nodes between $s_{i1}^{T_1}$ and $s_{j1}^{T_2}$. But for a general network, only a partial or none of the separations satisfy (14). Recall the example in Fig. 1, the overlapping node $v_8$ is allocated into $s_2 = \{8, 9, 10, 11, 12, 13, 14, 15\}$ from the 2nd-MST and allocated into $s_1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$ from the 1st-MST. Thus condition (14) cannot be satisfied due to the inconsistent node $v_8$. These inconsistent nodes are usually in the border of communities, in fact, they are overlapping nodes. So condition (15) is applied for community detection.

$$\left|s_{i1}^{T_1} - s_{j1}^{T_2}\right| \le \theta \quad \text{or} \quad \left|s_{i1}^{T_1} - s_{j2}^{T_2}\right| \le \theta, \quad i = 1, \dots, L_2, j = 1, \dots, M_2, \tag{15}$$

where $\theta$ is the largest number of inconsistent nodes. If a separation in (12) and a separation in (13) satisfy (15), then the network is separated into two parts and the overlapping nodes are also detected. A pair of separations which satisfies (15) is called a valid separation (pair). Two or more valid separation pairs which present the same result except overlapping nodes are called similar separations (separation pairs).

Here, two examples are given to illustrate the process.

**Table 2**
Intermediate results on the network of Example 1.

<table>
<tr><th colspan="3">Separations in the 1st-MST (A)</th><th colspan="2">Separations in the 2nd-MST (B)</th></tr>
<tr><td rowspan="5">distance ↓</td><td>$e_{4-10}$</td><td>1,2,3,4,5,6,7||8,9,10,11,12,13,14,15</td><td>$e_{7-10}$</td><td>9,10,11,12,13,14,15||1,2,3,4,5,6,7,8</td></tr>
<tr><td>$e_{47}$</td><td>1,2,3,5,6,7||4,8,9,10,11,12,13,14,15</td><td>$e_{12-14}$</td><td>9,12,13,15||1,2,3,4,5,6,7,8,10,11,14</td></tr>
<tr><td>$e_{67}$</td><td>2,3,5,6||1,4,7,8,9,10,11,12,13,14,15</td><td>$e_{27}$</td><td>1,2,4,5,6||3,7,8,9,10,11,12,13,14,15</td></tr>
<tr><td>$e_{9-10}$</td><td>9,11,13,14||1,2,3,4,5,6,7,8,10,12,15</td><td>$e_{12}$</td><td>1,4,5,6||2,3,7,8,9,10,11,12,13,14,15</td></tr>
<tr><td></td><td></td><td>$e_{10-14}$</td><td>9,12,13,14,15||1,2,3,4,5,6,7,8,10,11</td></tr>
</table>

**Example 1.** When $\delta = 0$, the 1st-MST and 2nd-MST of Fig. 1(a) are shown in Fig. 1(b) and (c) respectively. $N = 15$, both of the two MSTs have 14 separations. By removing the separations which satisfy (7) with $\varepsilon = 3$, the remaining separations are shown in Table 2.

In this example, $d_{avg}^{T1} = 0.5172$ and $d_{avg}^{T2} = 0.5871$. In (A) and (B) of Table 2, by eliminating those separations, each of them comes from removing an edge whose distance is less than $d_{avg}^{T1}$ and $d_{avg}^{T2}$ respectively, only one separation remains in the two columns of Table 2 respectively, they are $e_{4-10}$ and $e_{7-10}$. By comparing the two separations, it is seen that condition (15) is satisfied with $\theta \geq 1$. Thus the two communities are $s_1 = \{1, 2, 3, 4, 5, 6, 7\}$ and $s_2 = \{8, 9, 10, 11, 12, 13, 14, 15\}$, $v_8$ is an overlapping node, which is the separation in the first round MST.

In column (A) of Table 2, if we preserve the two separations by removing the edges of distance $d_{4-10} = 0.75$ and $d_{47} = 0.46$, then $v_4$ also can be detected as an overlapping node.

**Example 2.** Fig. 3 shows a graph of 24 nodes, which has 3 communities denoted in different shapes. The main results in the process are given in Table 3, where the parameters are $\delta = 0$, $\varepsilon = 3$, and $\theta = 1$.

Valid separation pair $\{e_{17-24}, e_{11-24}\}$ and $\{e_{12-17}, e_{11-24}|v_{17}\}$ are similar, the difference is only one overlapping node $v_{17}$. Thus, three communities are detected, they are: $s_1 = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $s_2 = \{10, 11, 12, 13, 14, 15, 16, 17\}$, and $s_2 = \{18, 19, 20, 21, 22, 23, 24\}$. Obviously, $v_9$ is an overlapping node between $s_1$ and $s_2$ and $v_{17}$ is an overlapping node between $s_2$ and $s_3$. Overlapping node $v_9$ is assigned into $s_1$ due to that it comes from $e_{5-11}$, a separation from the 1st-MST. Overlapping node $v_{17}$ is assigned into $s_2$ due to that it comes from $e_{17-24}$, which has larger distance than $e_{12-17}$.

Until now, we can summarize the above process and provide a detailed algorithm.

### 3.2. Two rounds MST based algorithm

**Input**: the adjacency matrix of a graph which describes the network.

**Output**: communities of the network and the overlapping nodes.

Parameters: $\delta$, $\varepsilon$, and $\theta$.

***Step*** 1: pick out the leaf nodes from the adjacency matrix by the method described in Section 2.2, and then compute the distance matrix $D_M$ by (2) and the method described in Section 2.1.

***Step*** 2: Compute the 1st-MST and the 2nd-MST from the distance matrix $D_M$. In the simulations of this paper, we use the Kruskal's algorithm [32].

***Step*** 3: From the two MSTs, generate two sequences of separations $S_\varepsilon^{T_1}$ and $S_\varepsilon^{T_2}$ (see (8) and (9)) respectively, which do not include those separation satisfying (7).

***Step*** 4: Compute the average distance $d_{avg}^{T_1}$ and $d_{avg}^{T_2}$ for those separations, see (10) and (11) for detail. From $S_\varepsilon^{T_1}$ and $S_\varepsilon^{T_2}$, remove those separations which come from removing an edge whose distance is less than $d_{avg}^{T_1}$ and $d_{avg}^{T_2}$, the resulted two sequences of separations are $S_d^{T_1}$ and $S_d^{T_2}$ respectively.

***Step*** 5: For a given value of $\theta$, examine each separation in $S_d^{T_1}$ with that in $S_d^{T_2}$ to see whether condition (15) holds, if hold, then the separation pair is valid and the separation in $S_d^{T_1}$ will be taken in use. Similar separations are incorporated together. Inconsistent nodes are the overlapping nodes.

***Step*** 6: Combine all the valid separations and obtain the communities and their overlapping nodes.

After the operations from step 2 to step 6, some of the edges are removed and several communities are detected. In a network with many communities, some of the above resulted communities may be further divided into sub-communities.

***Step*** 7: In each of the detected communities, repeat from step 3 again to see whether the communities can be further divided into sub-communities, until all the detected communities cannot be divided.

**Table 3**
Intermediate results on the network of Example 2.

| Separations in the first round MST ( $S_{\varepsilon}^{T_1}$ ) | | |
|---|---|---|
| distance | $e_{17-24}$ | 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17\|\|18,19,20,21,22,23,24 |
| | $e_{12-17}$ | 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16\|\|17,18,19,20,21,22,23,24 |
| | $e_{5-11}$ | 1,2,3,4,5,6,7,8,9\|\|10,11,12,13,14,15,16,17,18,19,20,21,22,23,24 |
| | $e_{12-16}$ | 1,2,3,4,5,6,7,8,9,10,11,13,14,15,16\|\|12,17,18,19,20,21,22,23,24 |
| | $e_{22-24}$ | 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17, 24\|\|18,19,20,21,22,23 |
| | $e_{56}$ | 1,2,3,4,6,7,8\|\|5,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24 |
| | $e_{21-22}$ | 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,22,24\|\|18,19,20,21,23 |
| | $e_{16}$ | 1,3,7,8\|\|2,4,5,6,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24 |
| | $e_{11-16}$ | 1,2,3,4,5,6,7,8,9,11\|\|10,12,13,14,15,16,17,18,19,20,21,22,23,24 |
| | | |
| Separations in the second round MST ( $S_{\varepsilon}^{T_2}$ ) | | |
| distance | $e_{5-16}$ | 1,2,3,4,5,6,7,8\|\|9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24 |
| | $e_{11-24}$ | 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17\|\|18,19,20,21,22,23,24 |
| | $e_{18-24}$ | 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19,21,24\|\|18,20,22,23 |
| | $e_{45}$ | 1,2,3,4,6,7,8\|\|5,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24 |
| | $e_{14}$ | 1,2,3,6,7\|\|4,5,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24 |
| | $e_{13-14}$ | 1,2,3,4,5,6,7,8,9,14,15,16,17\|\|10,11,12,13,18,19,20,21,22,23,24 |
| | $e_{11-13}$ | 1,2,3,4,5,6,7,8,9,10,12,13,14,15,16,17\|\|11,18,19,20,21,22,23,24 |
| | $e_{14-16}$ | 1,2,3,4,5,6,7,8,9,15,16,17\|\|10,11,12,13,14,18,19,20,21,22,23,24 |
| | $e_{12}$ | 1,4,5,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24\|\|2,3,6,7 |
| $d_{avg}^{T_1} = 0.6330$ , $d_{avg}^{T_2} = 0.5749$ | | |
| Separations in $S_d^{T_1}$ | | $e_{5-11}$, $e_{12-16}$, $e_{12-17}$, $e_{17-24}$ |
| Separations in $S_d^{T_2}$ | | $e_{5-16}$, $e_{18-24}$, $e_{11-24}$ |
| Consistent separation pair (θ=0)<br>{separation in 1-MST, separation in 2-MST } | | {$e_{17-24}$ , $e_{11-24}$} |
| Separation pairs with only one inconsistent node (θ=1)<br>{separation in 1-MST, separation in 2-MST\| overlapping nodes } | | { $e_{5-11}$, $e_{5-16}$ \| 9}, { $e_{12-17}$, $e_{11-24}$ \| 17} |

## 4. Simulations

The simulations are done on three types of networks: (1) several real world social networks; (2) graphs which are failed by the modularity maximization due to the resolution limit; (3) the LFR benchmark graphs for community detection.

### 4.1. Social networks

*Lusseau's network of bottlenose dolphins.* Fig. 4 shows the network of 62 bottlenose dolphins living in Doubtful Sound (New Zealand) [35,36]. An edge between a pair of dolphins is established if the associations occurred more frequently than expected by chance. The network is split naturally into two large communities [37,38], represented by squares and circles.

In this simulation, the parameters are $\delta = 0$, $\varepsilon = 3$, and $\theta = 2$. The 1st-MST and 2nd-MST are shown in Fig. 4(a) and (b) respectively, in which the leaf nodes are shaped in triangles. The distances of the edges are also shown in the figure.

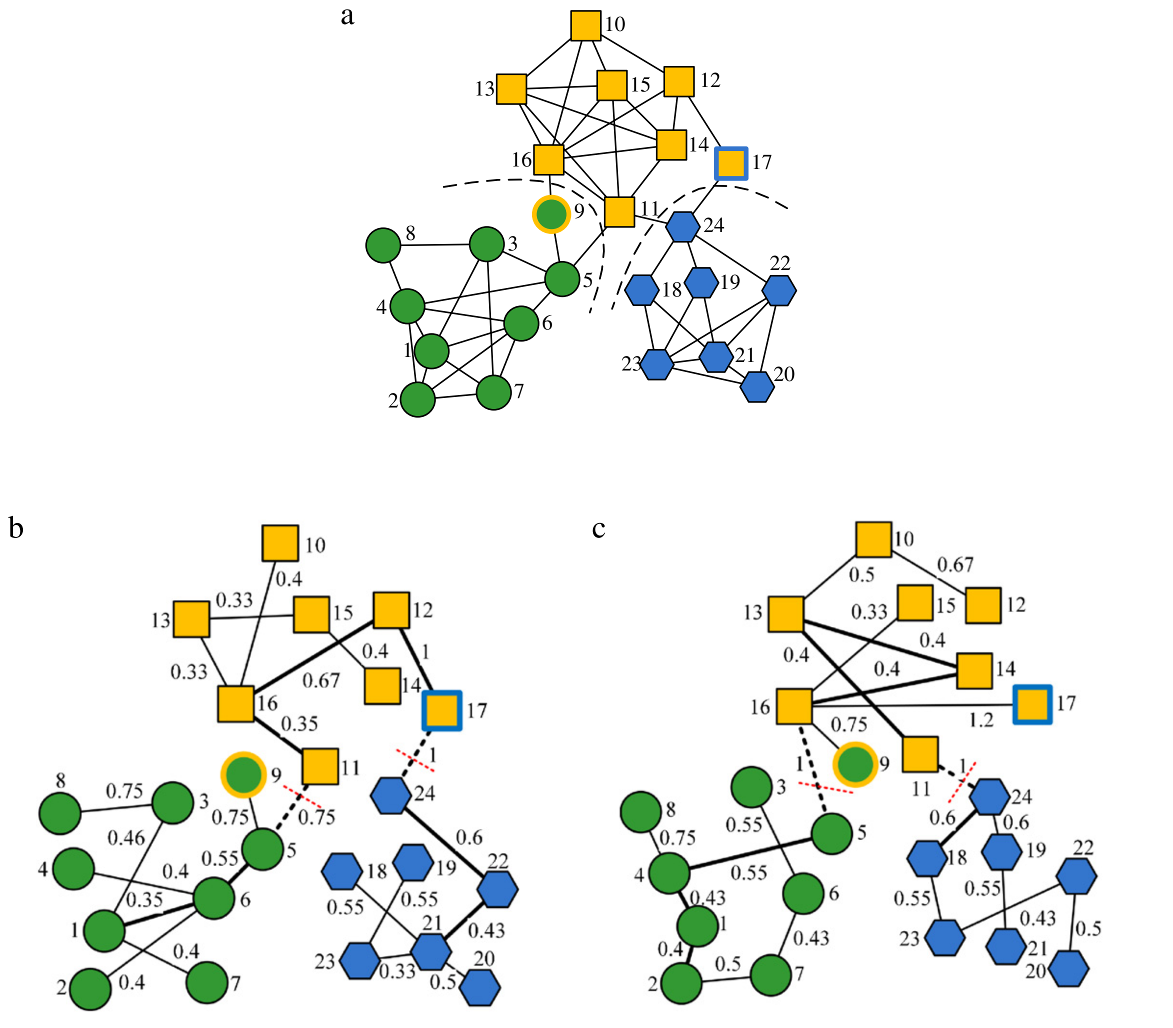

**Fig. 3.** A network and its two rounds MST. (a) The graph of the network. (b) The 1st-MST when $\delta = 0$. (c) The 2nd-MST when $\delta = 0$.

**Table 4**
Valid separation of the algorithm on the network of bottlenose dolphins.

| Valid separation pair | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Separation in $T_1$ | $L_1$ | $L_3$ | $L_2$ | $L_2$ | $L_3$ |
| Separation in $T_2$ | $L_1$ | $L_3$ | $L_1$ | $L_2$ | $L_1$ |
| Overlapping node | SN89 | SN89, Beescratch | SN89, PL | SN89, Oscar | SN89, Knit |

The average distance of the two MSTs are $d_{avg}^{T_1} = 0.3616$ and $d_{avg}^{T_2} = 0.3837$ respectively. After removing the separations (edges) in the 1st-MST whose distance is less than or equal to $d_{avg}^{T_1}$, only seven separations are preserved (see the edges with thick lines in Fig. 4(b)). Similarly, after removing the separations in the 2nd-MST, six separations are preserved (see the edges with thick lines in Fig. 4(c)). By comparing the separations in $S_d^{T_1}$ with that in $S_d^{T_2}$, five valid separation pairs are obtained (see Table 4) which satisfy (15), all of them are similar separations. Thus, the network is correctly separated into 2 communities with five overlapping nodes: SN89, Beescratch, PL, Oscar, and Knit.

*Krebs' network of books on American politics.* This network was compiled by V. Krebs [39,40] and is shown in Fig. 5. The nodes represent the books about US politics sold by the online bookseller Amazon.com and edges represent the frequent co-purchasing of books by the same buyers. Shapes of the nodes represent the political alignment of the books, specifically, circles are liberal, squares are conservative, and triangles are centrist or unaligned. Four communities are detected by our algorithm when $\delta = 0.2$, $\varepsilon = 3$, and $\theta = 2$, see the rectangle in the upper-right of Fig. 5(a). Fig. 5(b) is the 1st-MST and Fig. 5(c) is the 2nd-MST.

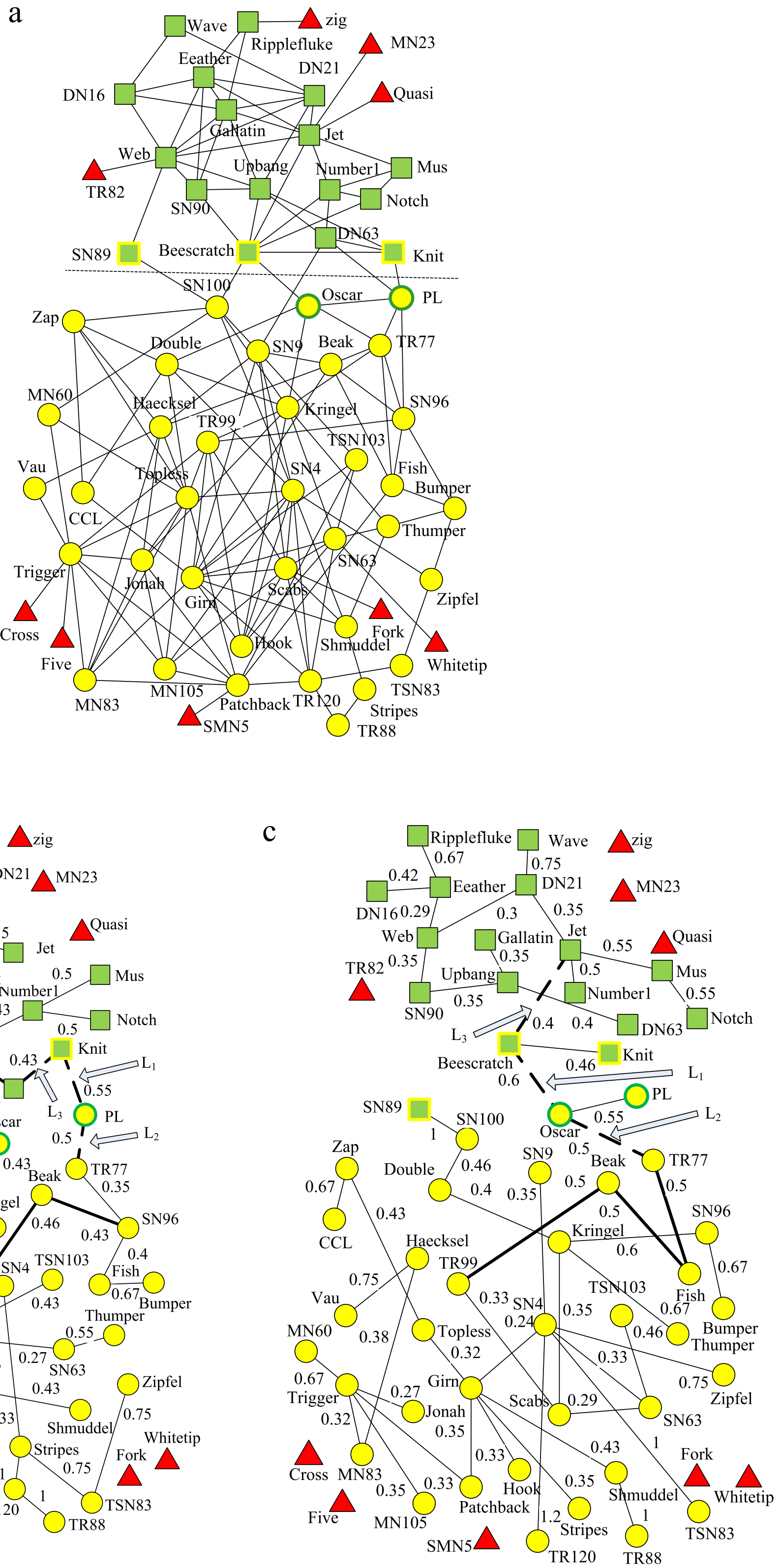

Fig. 4. Simulation results on the network of bottlenose dolphins when $\delta = 0$, $\varepsilon = 3$, and $\theta = 2$, in which the triangles indicate leaf nodes. (a) The graph of the network, dolphins in the two communities are denoted as squares and circles respectively, the result by our algorithm is indicated by the dashed line. (b) The 1st-MST. (c) The 2nd-MST.

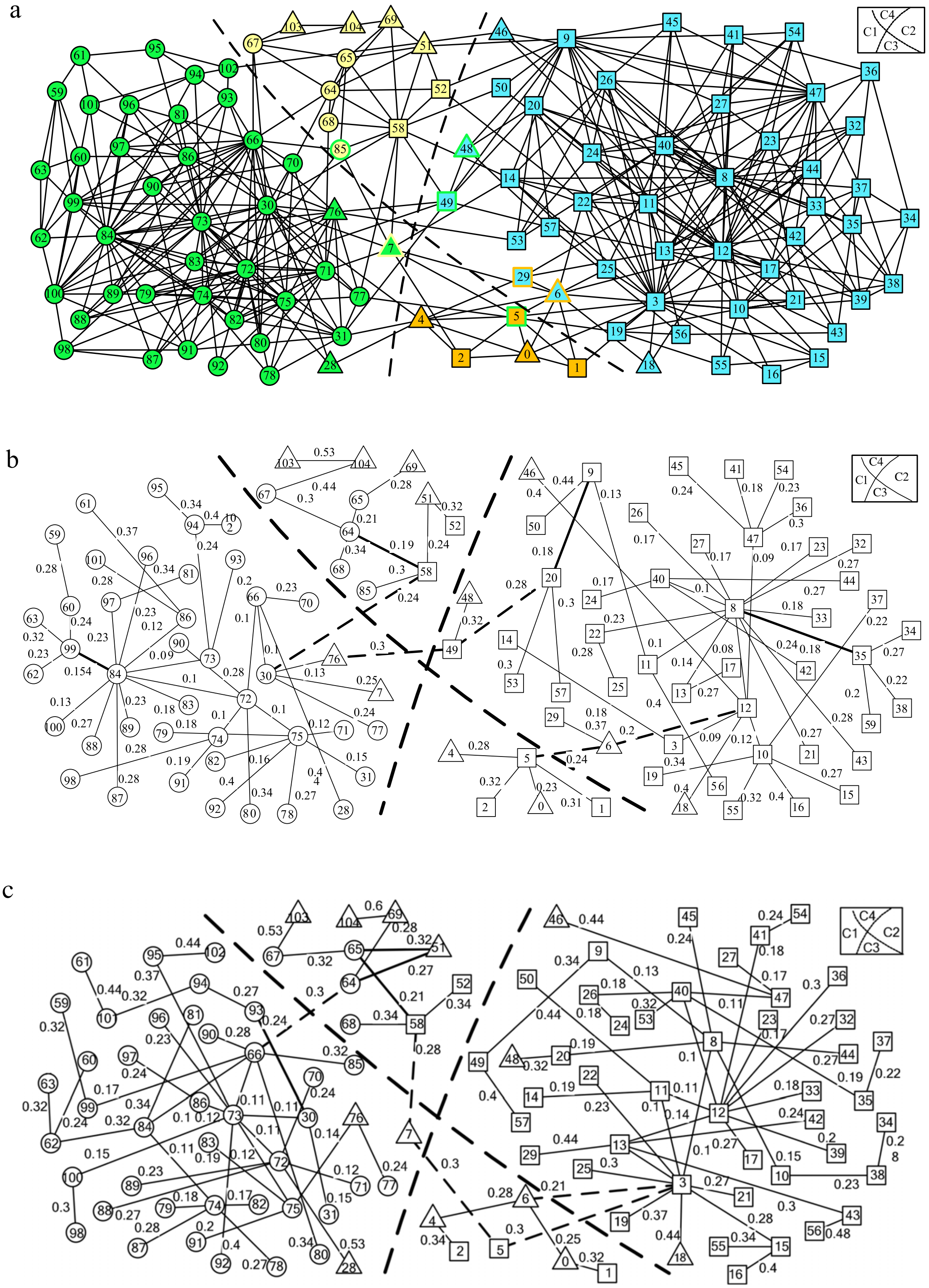

**Fig. 5.** Simulation results on the Krebs' network when $\delta = 0.2$, $\varepsilon = 3$, and $\theta = 2$. Four communities are detected; see the rectangle in the upper-right of the figures. (a) The graph of the network. (b) The 1st-MST. (c) The 2nd-MST.

The average distance of the two MSTs are $d_{avg}^{T_1} = 0.1539$ and $d_{avg}^{T_2} = 0.1831$ respectively in the first iteration of the algorithm from step 1 to step 6. After removing the separations (edges) in the 1st-MST whose distance is less than or equal to

**Table 5**
Intermediate results of the algorithm on the Krebs' network.

| | First iteration from step 2 to step 6 | | | | | | Second iteration |
|---|---|---|---|---|---|---|---|
| Resulted communities | (C1 + C4) \| (C2 + C3) | | | | (C1 + C2 + C4) \| C3 | | C1 \| C4 |
| Valid separation pair | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Separation in $T_1$ | $e_{49–76}$ | $e_{49–76}$ | $e_{49–76}$ | $e_{20–49}$ | $e_{5–6}$ | $e_{6–12}$ | $e_{30–58}$ |
| Separation in $T_2$ | $e_{5–7}$ | $e_{7–58}$ | $e_{3–5}$ | $e_{5–7}$ | $e_{3–6}$ | $e_{3–6}$ | $e_{64–66}$ |
| Overlapping nodes | None | $v_7$ | $v_5$ | $v_{48}, v_{49}$ | $v_5, v_6$ | $v_5, v_{29}$ | $v_{85}$ |

**Table 6**
Intermediate results of the algorithm on the graph of Fig. 6(a).

| | First iteration from step 2 to step 6 | | Second iteration |
|---|---|---|---|
| Resulted communities | C1 \| (C2, C3, C4) | (C1, C2, C3) \| C4 | C2 \| C3 |
| Separation in $T_1$ | $e_{1–21}$ | $e_{42–47}$ | $e_{28–41}$ |
| Separation in $T_2$ | $e_{13–21}$ | $e_{30–46}$ | $e_{30–42}$ |
| Overlapping node | None | None | None |

$d_{avg}^{T_1}$, only nine separations are preserved (see the edges with thick lines in Fig. 5(b)). Similarly, after removing the separations in the 2nd-MST, nine separations are preserved (see the edges with thick lines in Fig. 5(c)). By comparing the separations in $S_d^{T_1}$ with that in $S_d^{T_2}$, seven valid separation pairs obtained, which satisfy (15). The detailed intermediate results are given in Table 5. In the first iteration of the algorithm from step 4 to step 6, separation pair 1, 2, 3, and 4 are similar, they divide the network into two communities: (C1 + C4) and (C2 + C3). Separation pair 5 and 6 are similar, which divide the network into two communities: (C1 + C2 + C4) and C3. Thus three communities are detected: C2, C3, and (C1 + C4). After removing the correspond edges from the 1st- and 2nd-MST, in the second iteration of the algorithm in (C1 + C4) from step 4 to step 6, separation pair 7 divide the community (C1 + C4) into two sub-communities: C1 and C4. Thus four communities are detected from the algorithm, which is consistent with the result achieved by Newman [9].

For the overlapping nodes, from Table 5, it is obvious that $v_{85}$ is the overlapping node of community C1 and C4. As for $v_5$, $v_6$, $v_7$, $v_{29}$, $v_{48}$, and $v_{49}$ in Table 5, it is seen that they are nodes or nearest neighbors of the two MSTs with thick line. In the 1st-MST, $e_{76–49}$ traverse between community C1 and C2, which indicate $v_{48}$ and $v_{49}$ are their overlapping nodes; $e_{5–6}$ traverse between community C2 and C3, which indicate $v_5$, $v_6$ and $v_{29}$ are overlapping nodes of C2 and C3. In the 2nd-MST, $e_{5–7}$ traverse between community C1 and C3, which indicate $v_7$ and $v_5$ are their overlapping nodes. Thus, $v_5$ is the overlapping node of three communities: C1, C2, and C3.

### 4.2. Graphs failed by the modularity maximization

The resolution limit problem is introduced in detail in Refs. [19,20,22]. Due to the resolution limit, it is impossible to make sure whether a community (large or small), obtained through modularity maximization, is indeed a single community or a cluster of smaller communities. Fig. 6(a) shows a graph with four communities (cliques). However, it is identified only three communities by modularity maximization: two cliques of 20 nodes plus a community formed by the joining of the two cliques of 5 nodes [19,26]. Using the proposed algorithm, the four communities can be easily detected. Fig. 6 and Table 6 show the simulation results when $\delta = 0$, $\varepsilon = 3$, and $\theta = 2$. The 1st- and 2nd-MST are shown in Fig. 6(b) and (c) respectively. $d_{avg}^{T_1} = 0.3471$ and $d_{avg}^{T_2} = 0.52$, the corresponding edges in $S_d^{T_1}$ and $S_d^{T_2}$ are drawn in thick lines in Fig. 6(a) and (b) respectively. In first iteration of the algorithm C1, C4, and (C2 + C3) are detected; in second iteration community (C2 + C3) is divided into C2 and C3.

Simulation is also done on the graph shown in Fig. 3(A) in Ref. [19] with $m = 20$, which is also failed by the modularity maximization, the 10 cliques can be easily detected as communities by the proposed algorithm.

### 4.3. LFR benchmark graphs

The LFR benchmark generate graphs that both degree and community size distributions are power laws, with exponents $\gamma$ and $\beta$ respectively. The benchmark has six parameters: $N$ the number of nodes, $\langle k \rangle$ average degree, $k_{\max}$ maximum degree of nodes, $\gamma$-exponent for the degree of nodes, $\beta$-exponent for the community size, $\mu$ mixing parameter. Each node shares a fraction of $1 - \mu$ connections with the other nodes in its community and a fraction $\mu$ with the other nodes of the network. The proposed algorithm is also tested on the LFR benchmark with $N = 1000$ and $N = 5000$, respectively.

In Fig. 7, each curve shows the variation of the normalized mutual information with the mixing parameter $\mu$. The normalized mutual information is a measure of similarity between the detected communities and the true communities [31]. In the simulations, the parameters of the algorithm are changed according to the mixing parameter of the graphs, which indicates a difference between the modularity maximization and the proposed algorithm. If we have some priori information about the network, which may help us to set an optimal value for the parameters and consequently increase the performance of the algorithm.

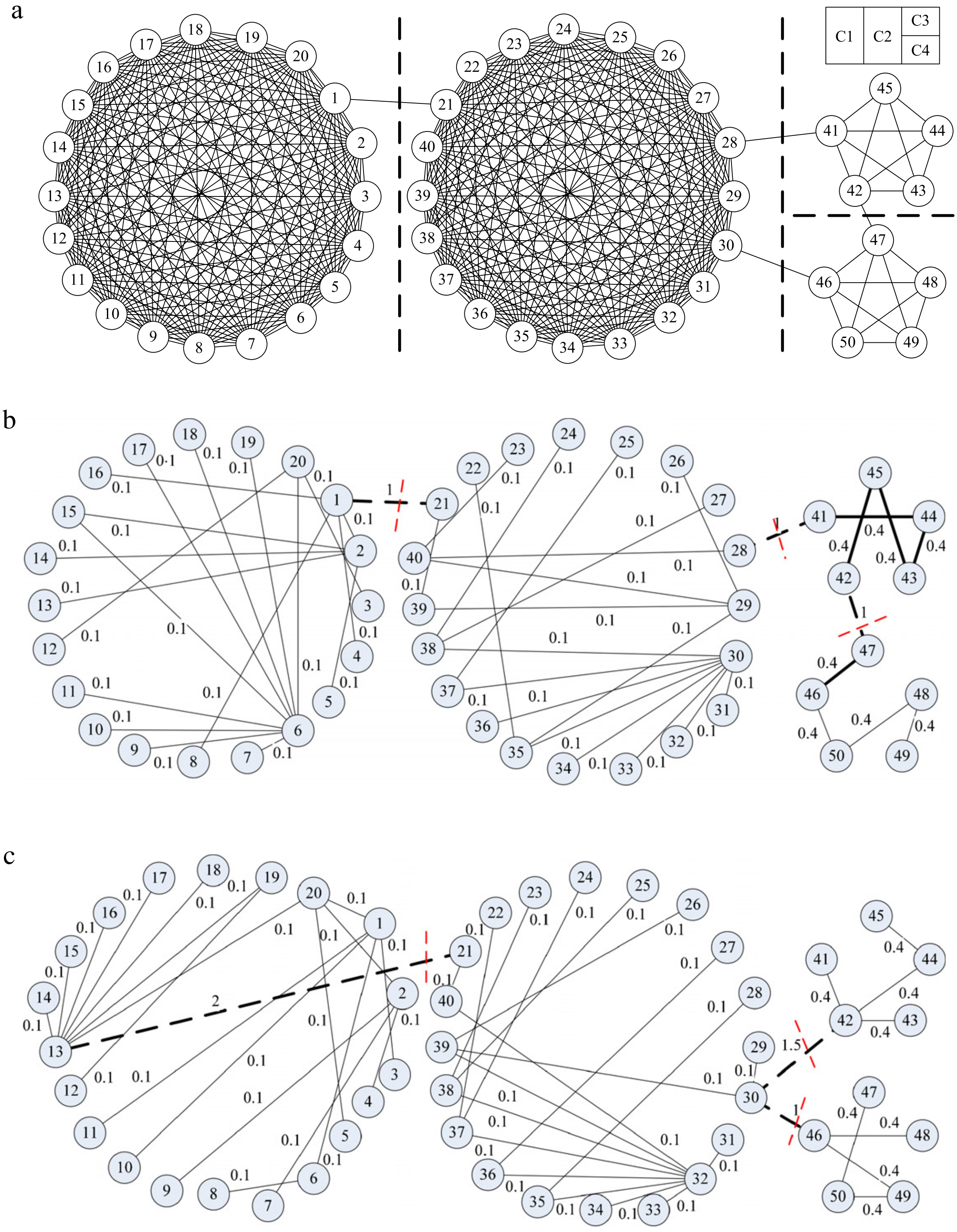

**Fig. 6.** Simulation results on a graph which is failed by modularity maximization when $\delta = 0$, $\varepsilon = 3$, and $\theta = 2$. Four communities are detected; see the rectangle in the upper-right of Fig. 4(a). (a) The graph of the network. (b) The 1st-MST. (c) The 2nd-MST.

## 5. Conclusions

A simple distance matrix is defined for a network, it is observed that the edges with large value of distance are usually distributed in the border of communities; in contrast, the edges with small value of distance are usually distributed within a community. By generating the two rounds minimum spanning tree, the community structure of the network can be revealed.

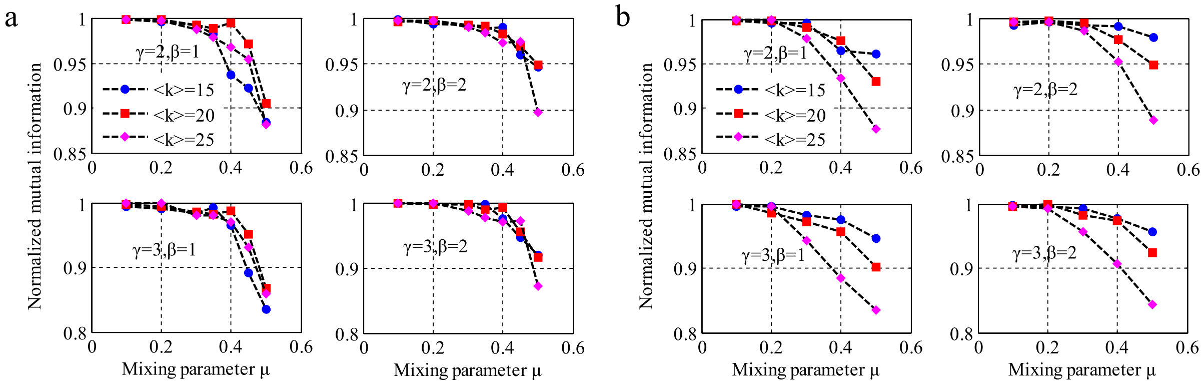

**Fig. 7.** Simulation results on the LFR benchmark with $k_{\max} = 40$. (a) $N = 1000$, the parameters are $\delta = 0$, $\varepsilon = 3$, and $\theta = 2$ when $\mu = 0.1$–$0.4$; $\delta = 0$, $\varepsilon = 5$, and $\theta = 5$ when $\mu = 0.45$; $\delta = 0.3$, $\varepsilon = 5$, $\theta = 8$ when $\mu = 0.5$. Each point corresponds to an average of 30 graphs realizations. (b) $N = 5000$, the parameters are $\delta = 0$, $\varepsilon = 3$, and $\theta = 2$ when $\mu = 0.1$–$0.4$; $\delta = 0$, $\varepsilon = 5$, $\theta = 5$ when $\mu = 0.5$. Each point corresponds to an average of 15 graphs realizations.

The proposed algorithm is deterministic, which is quite different from the modularity maximization. The algorithm is out of the problem of resolution limit, but may require some priori information to improve the performance.

## Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the National Natural Science Foundation of China (Nos. 61072139 and 61072106).

## References

[1] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99 (2002) 7821–7826.
[2] E.A. Leicht, M.E.J. Newman, Community structure in directed networks, Phys. Rev. Lett. 100 (2008) 118703.
[3] S. Gregory, Ordered community structure in networks, Physica A 391 (2012) 2752–2763.
[4] J. Huang, H. Sun, J. Han, B. Feng, Density-based shrinkage for revealing hierarchical and overlapping community structure in networks, Physica A 390 (2011) 2160–2171.
[5] C. Piccardi, L. Calatroni, F. Bertoni, Communities in Italian corporate networks, Physica A 389 (2010) 5247–5258.
[6] J. Wu, L. Jiao, C. Jin, F. Liu, M. Gong, R. Shang, W. Chen, Overlapping community detection via network dynamics, Phys. Rev. E 85 (2012) 016115.
[7] K. Steinhaeuser, N.V. Chawla, Identifying and evaluating community structure in complex networks, Pattern Recognit. Lett. 31 (2010) 413–421.
[8] M. Barigozzi, G. Fagiolo, G. Mangioni, Identifying the community structure of the international-trade multi-network, Physica A 390 (2011) 2051–2066.
[9] M.E.J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. USA 103 (2006) 8577.
[10] M.E.J. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (2004) 066133.
[11] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (2010) 75–174.
[12] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2004) 026113.
[13] M.E.J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (2006) 036104.
[14] S. Cafieri, P. Hansen, L. Liberti, Loops and multiple edges in modularity maximization of networks, Phys. Rev. E 81 (2010) 046102.
[15] S. Cafieri, P. Hansen, L. Liberti, Locally optimal heuristic for modularity maximization of networks, Phys. Rev. E 83 (2011) 056105.
[16] W. Zhan, Z. Zhang, J. Guan, S. Zhou, Evolutionary method for finding communities in bipartite networks, Phys. Rev. E 83 (2011) 066120.
[17] Z. Lü, W. Huang, Iterated tabu search for identifying community structure in complex networks, Phys. Rev. E 80 (2009) 026130.
[18] M.J. Barber, J.W. Clark, Detecting network communities by propagating labels under constraints, Phys. Rev. E 80 (2009) 026129.
[19] S. Fortunato, M. Barthélemy, Resolution limit in community detection, Proc. Natl. Acad. Sci. USA 104 (2007) 36–41.
[20] J.M. Kumpula, J. Saramäki, K. Kaski, J. Kertész, Limited resolution and multiresolution methods in complex network community detection, Proc. SPIE 6601 (2007) 660116.
[21] B.H. Good, Y.-A. de Montjoye, A. Clauset, The performance of modularity maximization in practical contexts, Phys. Rev. E 81 (2010) 046106.
[22] A. Lancichinetti, S. Fortunato, Limits of modularity maximization in community detection, Phys. Rev. E 84 (2011) 066122.
[23] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (2010) 651–666.
[24] N.A. Alves, Unveiling community structures in weighted networks, Phys. Rev. E 76 (2007) 036101.
[25] E. Estrada, N. Hatano, Communicability in complex networks, Phys. Rev. E 77 (2008) 036111.
[26] E. Estrada, Community detection based on network communicability, Chaos 21 (2011) 016103.
[27] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying communities in networks, Proc. Natl. Acad. Sci. USA 101 (2004) 2658–2663.
[28] C.T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Trans. Comput. C-20 (1971) 68–86.
[29] Y. Xu, V. Olman, D. Xu, Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning tree, Bioinformatics 18 (2002) 536–545.
[30] C. Zhong, D. Miao, R. Wang, A graph-theoretical clustering method based on two rounds of minimum spanning trees, Pattern Recognit. 43 (2010) 752–766.
[31] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Rev. E 78 (2008) 046110.
[32] J.B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem, Proc. Amer. Math. Soc. (1956) 48–50.
[33] R. Prim, Shortest connection networks and some generalizations, Bell Syst. Tech. J. 36 (1957) 1389–1401.
[34] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, England, 1986.
[35] D. Lusseau, The emergent properties of a dolphin social network, Proc. R. Soc. Lond. Ser. B 270 (2003) S186–S188.
[36] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, S.M. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Can geographic isolation explain this unique trait? Behav. Ecol. Sociobiol. 54 (4) (2003) 396–405.
[37] F. Xie, M. Ji, Y. Zhang, D. Huang, The detection of community structure in network via an improved spectral method, Physica A 388 (2009) 3268–3272.
[38] J. Wu, R. Lu, L. Jiao, F. Liu, X. Yu, D. Wang, B. Sun, Phase transition model for community detection, Physica A 392 (2013) 1287–1301.
[39] http://www.cise.ufl.edu/research/sparse/matrices/Newman/index.html.
[40] V. Krebs, http://www.orgnet.com/, (unpublished).