Discovering Global Network Communities Based on Local Centralities

BO YANG
Jilin University
and
JIMING LIU
Hong Kong Baptist University

One of the central problems in studying and understanding complex networks, such as online social networks or World Wide Web, is to discover hidden, either physically (e.g., interactions or hyperlinks) or logically (e.g., profiles or semantics) well-defined topological structures. From a practical point of view, a good example of such structures would be so-called network communities. Earlier studies have introduced various formulations as well as methods for the problem of identifying or extracting communities. While each of them has pros and cons as far as the effectiveness and efficiency are concerned, almost none of them has explicitly dealt with the potential relationship between the global topological property of a network and the local property of individual nodes. In order to study this problem, this paper presents a new algorithm, called ICS, which aims to discover natural network communities by inferring from the local information of nodes inherently hidden in networks based on a new centrality, that is, clustering centrality, which is a generalization of eigenvector centrality. As compared with existing methods, our method runs efficiently with a good clustering performance. Additionally, it is insensitive to its built-in parameters and prior knowledge.

Categories and Subject Descriptors: G.2 [**Discrete Mathematics**]: Graph Theory—*Network problems*

General Terms: Algorithm, Design, Theory

Additional Key Words and Phrases: Complex network, community mining, graph theory, centrality, World Wide Web

The research work reported in this article has been carried out in the Center for e-Transformation Research at Hong Kong Baptist University, and supported in part by a Hong Kong RGC Central Allocation Grant (HKBU 2/03/C), the Major State Basic Research Development Program of China (973 Program) (2003CB317001), and a National Natural Science Foundation of China grant (No. 60503016).

Authors' addresses: B. Yang, College of Computer Science and Technology, Jilin University, Changchun, China 130012; email: ybo@jlu.edu.cn; J. Liu, Department of Computer Science, Hong Kong Baptist University, Kowloong Tong, Hong Kong; email: jiming@comp.hkbu.edu.hk.

ACM Reference Format:

Yang, B. and Liu, J. 2008. Discovering global network communities based on local centralities. ACM Trans. Web 2, 1, Article 9 (February 2008), 32 pages. DOI = 10.1145/1326561.1326570 http://doi.acm.org/10.1145/1326561.1326570

1. INTRODUCTION

Network communities are groups of network nodes, within which links are dense, but between which links are sparse [Girvan and Newman 2002, Newman 2004a]. The ability to discover communities from different kinds of networks can help us better understand and visualize their topological structure, and hence exploit them more effectively from a practical point of view. For instance, the ability to discover Web communities is useful or even crucial for search engines to improve both the efficiency and the accuracy of their search [Flake et al. 2002; Kleinberg 1999].

Although algorithms addressing this problem have been previously developed, such as spectral methods [Fiedler 1973; Pothen et al. 1990; Shi and Malik 2000], Kernighan-Lin algorithm [Kernighan and Lin 1970], Girvan-Newman algorithm [Girvan and Newman 2002; Newman and Girvan 2004], Newman algorithm [Newman 2004b], MFC algorithm [Flake et al. 2002], HITS algorithms [Kleinberg 1999], as well as others [Scott 2000; Wu and Huberman 2004; Tyler et al. 2003; Radicchi et al. 2004; Pirolli et al. 1996; Kumar et al. 1999; Chakrabarti et al. 1999; Reichardt et al. 2004], the issue of how to efficiently and effectively discover network communities remains an open challenge, since it is nontrivial to get a good trade-off among the following three important requirements: speed, accuracy, and insensitivity to parameters. As a result, many of the existing methods with good accuracy tend to be computationally expensive (with a time complexity of at least O(n^2)), sensitive to their build-in parameters, and dependent on prior knowledge.

For examples, bisection methods, such as spectral methods [Fiedler 1973; Pothen et al. 1990; Shi and Malik 2000] and Kernighan-Lin algorithm [Kernighan and Lin 1970], need to know the expected number or the approximate sizes of communities, in order to detect and extract all communities. Hierarchical methods, such as Girvan-Newman algorithm [Girvan and Newman 2002; Newman and Girvan 2004], Newman algorithm [Newman 2004b], and their improved variants [Tyler et al. 2003; Radicchi et al. 2004], on the other hand, output dendrograms representing the hierarchical structures of communities; it is not easy to determine which layer of the dendrogram corresponds to a meaningful partition of a given network. Newman has earlier provided a solution to this issue based on the notion of modularity [Newman and Girvan 2004]. He suggests that one should cut the dendrogram to produce a partition with the maximum modularity. Nevertheless, in many real-world applications, the expected partitions often correspond to local optima rather than global ones.

The methods for identifying Web communities are also sensitive to their parameters. The MFC algorithm [Flake et al. 2002] requires some source pages as sinks, in order to call the Max flow-Min cut procedure, which may potentially influence the clustering accuracy. The HITS algorithm [Kleinberg 1999] relies on the quality of the initial graph obtained by the query result of a search engine. The SAE algorithm [Pirolli et al. 1996] is sensitive to its parameters, such as the relaxing rate and the spreading rate.

In our previous work, we have studied how to discover community structures from signed social networks, which contain both positive and negative links, using an agent-based heuristic method [Yang et al. 2007]. Also, we have developed an AOC (autonomy oriented computing [Liu et al. 2004; Liu et al. 2005]) based self-organization method for discovering communities from distributed and decentralized networks, such as sensor networks and communication networks [Yang and Liu 2007]. These studies have inspired us to rethink this problem from a different angle, that is, how to develop an effective and efficient method to discover communities based on the relationship between the global network structure and the local property of individual nodes. According to this idea, our present work has developed a new algorithm, called ICS, based on a proposed node centrality. As will be described later in this paper, the ICS algorithm demonstrates good performance in the above-mentioned three aspects.

The remainder of the article is organized as follows: Section 2 describes our definition of community structure and the basic idea behind the ICS. Section 3 presents the ICS algorithm. Section 4 tests its performance against different networks. Section 5 compares it with existing methods and discusses its distinct features. Finally, Section 6 concludes the paper by highlighting the major contributions of our work and discusses some future extensions.

2. DEFINITIONS AND THE BASIC IDEA

2.1 Network Community

Definition 2.1. Graph $G = (V, E)$ is a network where V is the set of nodes (or vertices), E is the set of links (or edges), $|V| = n$ and $|E| = m\,.\,\pi = (G_1, G_2)$ is a bipartition of G. $G_1 = (V_1, E_1)$, $|V_1| = n_1$, $|E_1| = m_1$. $G_2 = (V_2, E_2)$, $|V_2| = n_2$, $|E_2| = m_2$. A, A_1, A_2, respectively, are the adjacency matrices of G, G_1, G_2. G is said to be *dividable* if A, A_1 and A_2 satisfy the constraint condition $C(A, A_1, A_2)$. Otherwise, G is not dividable and is a *community*. $C(A, A_1, A_2)$ is defined as follows:

$$C(A, A_1, A_2) = \left(\sum_{j=1}^{n_1} A_{ij} \geq \sum_{j=n_1+1}^{n} A_{ij},\, 1 \leq i \leq n_1\right) \wedge \left(\sum_{j=1}^{n_1} A_{ij} \leq \sum_{j=n_1+1}^{n} A_{ij},\, n_1 + 1 \leq i \leq n\right). \tag{2.1}$$

For each node of G_1 ($1 \leq i \leq n_1$), the number of its incident links inside G_1 should be no less than that outside G_1, that is, $\sum_{j=1}^{n_1} A_{ij} \geq \sum_{j=n_1+1}^{n} A_{ij}$. Similarly, for each node of G_2 ($n_1 + 1 \leq i \leq n$), we have $\sum_{j=1}^{n_1} A_{ij} \leq \sum_{j=n_1+1}^{n} A_{ij}$.

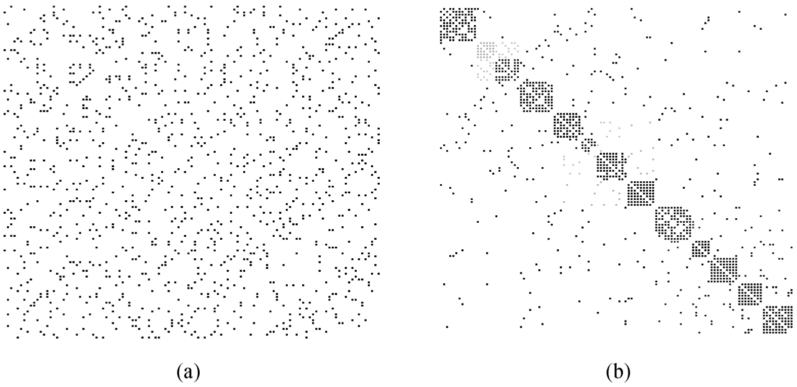


Fig. 1. A schematic example to illustrate the idea behind our algorithm. (a) The adjacency matrix of the football association network; (b) the output of our algorithm.

2.2 The Basic Idea behind Our Approach

The adjacency matrix of a given network consists of zeros and positive entries. If a network can be clearly divided into two communities, its adjacency matrix A will be transformed into an approximate diagonal matrix with two blocks, A_1 and A_2, in which A_1 and A_2 are two dense submatrices with more nonzero entries and the remainder parts of A are very sparse with more zero entries. Based on this observation, we can use the following procedure to identify all communities hidden in a given network:

First, we transform the initial matrix of a given network into an approximately diagonal one. Then, we find the partition position in the transformed matrix, such that it can bipartition the matrix into two submatrices satisfying the constraint condition C. Finally, we refine the two submatrices in a recursive way until each of the existing submatrices cannot be further divided, that is, each existing subnetwork has turned into a community. At the end of the procedure, the initial adjacency matrix of the network will be transformed to a highly regular diagonal one, in which each diagonal block denotes one network community.

Figure 1 illustrates the result of discovering all communities of a football association network [Girvan and Newman 2002] using the above mentioned method. Figure 1(a) shows its initial adjacency matrix of the network, whereas Figure 1(b) presents the output matrix in which the diagonal blocks with different grey degrees denote different communities, that is, football associations.

3. ALGORITHMS

3.1 Main Algorithm

The main steps of the ICS algorithm for discovering the communities of a network are given as follows, where the input A_0 denotes the adjacency matrix of the input network to be clustered, and the output A denotes the output matrix of the clustered network:

Algorithm 3.1. $A = \text{ICS}(A_0)$

```
1. Transform $A_0$ into a diagonal matrix $A$ by calling TAM ($A_0$) to be introduced in
   Section 3.2;
2. Find a bipartition of $A$ satisfying $C(A, A_1, A_2)$ by calling BP ($A$) to be introduced in
   Section 3.3;
3. If such a bipartition does not exist then return $A$; otherwise, bipartition $A$ into $A_1$
   and $A_2$;
4. $A'_1 = \text{ICS}\ (A_1)$;
5. $A'_2 = \text{ICS}\ (A_2)$;
6. Return the diagonal matrix consisting of $A'_1$ and $A'_2$.
```

3.2 Matrix Transformation Algorithm

Traditional optimization methods such as genetic algorithms, simulated annealing, and local search are time-consuming because they usually require much time to converge. Their performance is also extremely sensitive to the choice of parameters. In our work, we provide a new algorithm for quickly transforming an irregular adjacency matrix into an approximately diagonal one. The basic idea behind this algorithm is based on the concept of *clustering centrality*.

Definition 3.1. Let A be the adjacency matrix of network N. The *clustering centrality* of node i at time t is defined as:

$$c_i^{(t)} = ac_i^{(t-1)} + (1-a)\sum_{j=1}^{n} A_{ij}c_j^{(t-1)} \Bigg/ \sum_{j=1}^{n} A_{ij}, \tag{3.1}$$

where constant a is called *temporal coefficient* and $0 < a < 1$.

In essence, the concept of clustering centrality is a generalization of eigenvector centrality presented by Bonacih [Bonacich 1972, 1987]. Like other centralities such as degree, closeness, and betweenness [Freeman 1977], eigenvector centrality is also used to measure the importance of a node in a given network such as prestige, prominence, and power. Eigenvector centrality is defined as an n-dimension vector $C = (c_i)_n^T$, in which c_i denotes the centrality of node i and is defined as:

$$c_i = \lambda^{-1}\sum_{j=1}^{n} A_{ij}c_j, \tag{3.2}$$

where A is the adjacency matrix of a network, and λ is a constant. Equation (3.2) means an individual will become more powerful when he/she is associated to some powerful people. In the form of matrix, Equation (3.2) becomes:

$$\lambda C = AC. \tag{3.3}$$

From Equation (3.3) we find that vector C is an eigenvector of matrix A corresponding to the eigenvalue λ. This is the origin of the name *eigenvector centrality*. Actually, λ is the principal eigenvalue of A, and C is the corresponding eigenvector, both of which can be computed by an accelerated power method given in Hotelling [1936].

Now we extend the constant λ to a diagonal matrix D

$$D = diag(\lambda_1, \dots, \lambda_n), \tag{3.4}$$

where $\lambda_i = k_i = \sum_{j=1}^{n} A_{ij}$, the degree centrality of node i.

Therefore, Equation (3.2) becomes

$$c_i = \lambda_i^{-1} \sum_{j=1}^{n} A_{ij} c_j. \tag{3.5}$$

Correspondingly, Equation (3.3) becomes

$$C = D^{-1}AC = NC. \tag{3.6}$$

The normal matrix N always has the largest eigenvalue equal to 1 associated to a trivial eigenvector due to the sum of each row is equal to 1. So, the clustering centrality vector C is actually the dominant eigenvector of the normal matrix N, which can be iteratively computed using the following power method:

$$C^{(t)} = aC^{(t-1)} + (1-a)\,NC^{(t-1)}. \tag{3.7}$$

Note that, Equation (3.7) is actually the matrix version of Equation (3.1).

As shown in Equation (3.7), the clustering centrality vector at time$t$ is determined by two terms: the previous centrality vector $C^{(t-1)}$ and the new centrality vector $NC^{(t-1)}$. The coefficient, a, is used to improve the convergence speed of the power method by regulating the tradeoff between the old and new information. For this reason, we refer to a as a temporal coefficient.

Equation (3.1) can be understood in depth from the viewpoint of a random walk process. Suppose that, at each node, there is an agent whose objective is to move to a virtual destination. Without any heuristic information, each agent wanders from one node to another along the links until it hits the destination. At each step, the agent has two choices: 1) stay at current node with probability a, or 2) leave current node and go to one of its neighbors with probability 1–a. Actually, the clustering centrality of node i at time t, defined by Equation (3.1), is the probability that the agent starting from node i hits the destination after t steps. Due to the dense reciprocal linkage within communities, it is much easier to hit the destination after a few steps if the starting points of agents are within the community containing the destination (destination community). Otherwise, it will be very hard because it is very difficult to enter the destination community through sparse intercommunity “bridges.” After enough walking steps, the clustering centralities of nodes within destination community will be greater than those of outside destination community. Then we can extract destination community from entire network by appropriately cutting the sorted centrality distribution.

Based on the above discussion, the main steps of the proposed TAM algorithm for transforming a matrix are given as follows, where A_0 denotes the input adjacency matrix, and A denotes the transformed matrix.

Algorithm 3.2. $A = \mathrm{TAM}(A_0)$

```
1. Initialize $C^{(0)}$ with $n$ random numbers between 0 and 1;
2. $t = 1$;
3. for $i = 1$:n
4.     update $c_i^{(t)}$according to Equation 3.1;
5. endfor
6. $t = t + 1$;
7. Repeat Steps 3 to 6 until the following condition is true:
       $\max_k\{c_k^{(t)}\} - \min_k\{c_k^{(t)}\} < \varepsilon$
8. Sort all nodes into a non-increasing permutation according to their clustering
   centralities;
9. Rebuild $A_0$ into $A$ according to the permutation generated by Step 8.
```

The convergence speed of the power method is governed by the term of $(\lambda_2/\lambda_1)^t$, where λ_1 and λ_2 are, respectively, the largest and second largest eigenvalues. Thus, the speed of TAM is not decided by the dimension of the input matrix, but by the value of λ_2. Practically speaking, we have observed that TAM converges quite fast even for very large networks. In fact, the clustering centrality vector that is far from the convergence has provided enough information for TAM to rearrange all nodes into a permutation in which all nodes in a virtual destination community are aggregated together. From the point of random walk, a small walking step is already enough to distinguish the destination community from others. Therefore, TAM does not need to calculate a "real" eigenvector close to convergence using quite many iterative steps. In order to test the convergence speed of TAM for practical networks, we have run the algorithm against several networks of various scales. In this experiment, we set $\varepsilon = 10^{-4}$ and each network is tested with different temporal coefficients. n and m in the legend denote the numbers of nodes and links, respectively.

Figure 2 tells us some interesting properties of TAM. First, its convergence speed is fairly fast; usually no more than 100 iterations, even for a very large network with 10^5 nodes and 10^6 links. Therefore, it is reasonable to consider the number of required iterations to be insensitive to the network scale. Second, its convergence speed can be regulated by the temporal coefficient. Usually, TAM will converge faster with a greater value of the temporal coefficient.

In each iteration (Steps 3 to 6), each component of $C^{(t)}$ is updated using Equation (3.1) and will take $\mathrm{O}(k_i)$ time, where k_i is the degree of node i. The time taken in each iteration is bounded by $\sum_{i=1}^{n} \mathrm{O}(k_i + 1) = \mathrm{O}(\sum_{i=1}^{n} k_i + \sum_{i=1}^{n} 1) = \mathrm{O}(m + n)$. Thus, the time of Steps 3 to 7 is $\mathrm{O}(t(m + n))$. Step 8 takes $\mathrm{O}(n\log n)$ time to sort all nodes. Step 9 rebuilds the matrix by a matrix scanning that will take $\mathrm{O}(m + n)$ time. Therefore, the total time complexity of the TAM algorithm is $\mathrm{O}(t(m+n)+n\log n)$. From above experiments, we know that t is insensitive to the network scale and is quite small compared with n and m.

The TAM algorithm can transform an irregular adjacency matrix into an approximately diagonal one by rearranging rows and columns according to the sorted clustering centralities. We will illustrate this using four practical

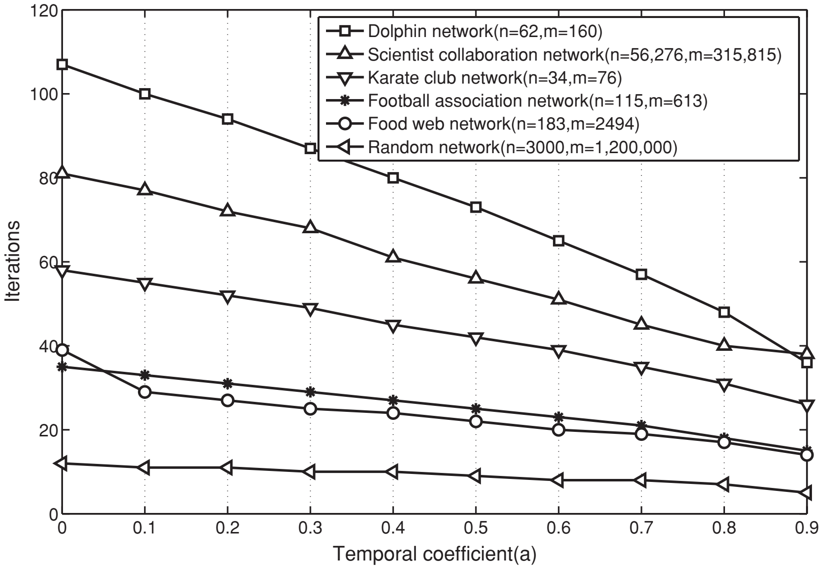


Fig. 2. Testing the convergence speed of TAM against networks of various scales.

social networks including the karate club network [Zachary 1977], the football association network [Girvan and Newman 2002], the dolphin network [Lusseau 2003], and the food web network (http://www.cosin.org/extra/data/foodwebs/WEB.html), as shown in Figure 3. We set $a = 0.5$ for all experiments. Compared with their respective initial adjacency matrices, we can see that all of transformed adjacency matrices become approximately diagonal.

3.3 Matrix Bipartition Algorithm

In this section, we will discuss how to find a bipartitions of a transformed matrix satisfying with conditions C defined in Definition 2.1. We first define two (n-1)-dimension vectors, Num$_1$ and Num$_2$. The *pos*-th component of Num$_1$ is defined as follows:

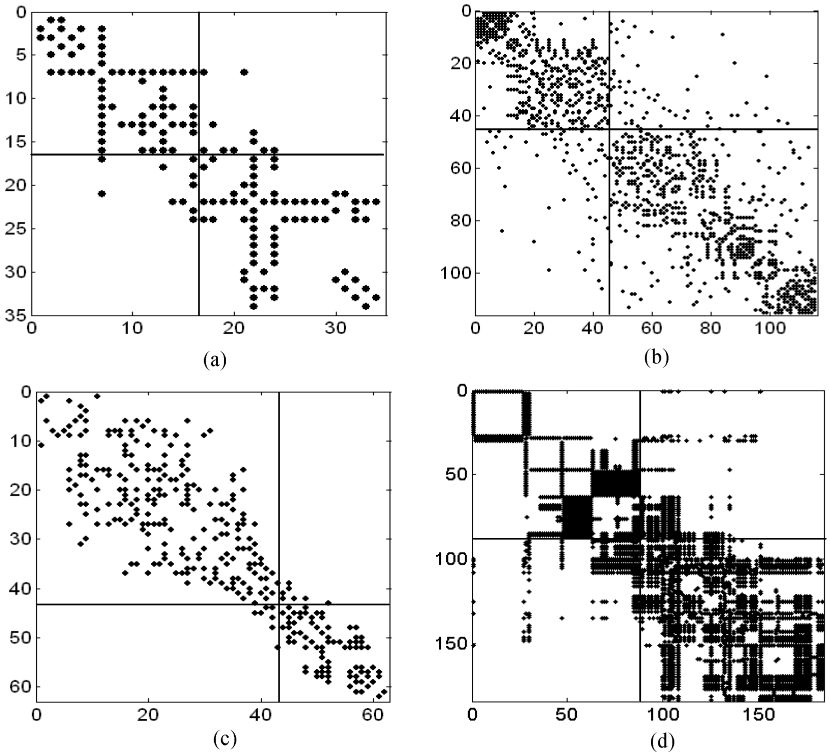
$$Num_1(pos) = \left\| \left\{ i, 1 \le i \le pos \;\middle|\; \sum_{1 \le j \le pos} A_{ij} \ge \sum_{pos < j \le n} A_{ij} \right\} \right\|, \quad \text{for } 1 \le pos < n. \tag{3.8}$$

The *pos*-th component of Num$_2$ is defined as:

$$Num_2(pos) = \left\| \left\{ i, pos < i \le n \;\middle|\; \sum_{1 \le j \le pos} A_{ij} \le \sum_{pos < j \le n} A_{ij} \right\} \right\|, \quad \text{for } 1 \le pos < n. \tag{3.9}$$

where $||S||$ denotes the size of set S.

Num$_1$(pos) and Num$_2$(pos), respectively, denote the numbers of cut positions satisfying the first part and the second part of constraint C. Num$_1$ and Num$_2$ can be quickly computed through at most three scans of the complete matrix.



(a) (b) (c) (d)

Fig. 3. The outputs of the TAM algorithm. The shown matrices are the transformed matrices after rearranging the rows and columns of their initial adjacency matrices according to the sorted clustering centralities. Dots in panes denote the non-zero entries of matrices. (a) The output matrix of the karate club network; (b) the output matrix of the football association network; (c) the output matrix of the dolphin network; (d) the output matrix of the food web network.

The first top-down matrix scanning is to compute the degree vector K, such that K(i) is the degree of node i. The second top-down matrix scanning is to compute the vector Num_1, as listed in Algorithm 3.3. In a similar manner, we can compute the vector Num_2 through a bottom-up matrix scanning.

Algorithm 3.3. $\mathrm{Num}_1 = \mathrm{Comp\_Num}_1(A)$

```
$\mathrm{Num}_1$ = zeros(1,n); /* initialize a zero vector */
for r = 1:n /* r-row index */
    sum = 0;
    for c = 1:n /* c-row index */
        sum = sum + A(r,c);
        if Sum >= K(r)/2, break; endif;
    endfor
    mid (r) = c;
endfor
for r = 1:n
    if mid(r) <= r
```

```
            $\mathrm{Num}_1$(r) = $\mathrm{Num}_1$(r) +1;
        else
            $\mathrm{Num}_1$(mid (r)) = $\mathrm{Num}_1$(mid (r)) +1;
        endif
endfor
for r = 2:n
        $\mathrm{Num}_1$(r) = $\mathrm{Num}_1$ (r) + $\mathrm{Num}_1$ (r-1);
endfor
```

Based on Num_1 and Num_2, a vector S that measures how well different positions can fit the constraint C is defined as:

$$S(pos) = \frac{Num_1(pos)}{pos} + \frac{Num_2(pos)}{n - pos}. \tag{3.10}$$

It is easy to show that $0 \leq S(pos) \leq 2$ for $1 \leq pos < n$. The candidate positions that satisfy the condition C are those with S-values equal to 2. For the cases with multiple candidates, instead of selecting one at random, one can choose the middlemost one in order to obtain communities with a roughly equal size. For the purpose of demonstrating the effectiveness of this bipartition method, Figure 3 shows the computed cut positions, which are represented by the solid crosses. All of them are identical with, or close to, real splits.

If such a position does not exist in the transformed matrix by the TAM algorithm, it indicates that the current network is already cohesive enough and does not need to be further divided. This corresponds to Step 3 of ICS. In this way, prior knowledge, such as the number of communities, is not required to decide how to stop the recursive bisections.

One can soften the constraint C to obtain a more flexible criterion, that is, a fuzzy constraint, defined as follows:

$$\begin{aligned} FC(A) : &\left(\sum_{j=1}^{n_1} A_{ij} \gg \sum_{j=n_1+1}^{n} A_{ij}, 1 \leq i \leq n_1 \right) \\ &\wedge \left(\sum_{j=1}^{n_1} A_{ij} \ll \sum_{j=n_1+1}^{n} A_{ij}, n_1 + 1 \leq i \leq n \right), \end{aligned} \tag{3.11}$$

where $\gg$ and $\ll$, respectively, denote “is much greater than” and “is much less than.” Such a fuzzy cut can be directly computed with the aid of the vector S as defined in Equation (3.10). That is, an optimal cut position in terms of the FC can be determined as follows:

$$pos = \arg\max(S). \tag{3.12}$$

In order to decide whether or not a sub-network has already been a community under the fuzzy constraint, we will present a new stopping criterion.

Suppose that A is the adjacency matrix of a given network with n nodes. Its utility function is defined as:

$$u(A) = C_n^2 - \sum_{i,j} A_{ij}/2. \tag{3.13}$$

In fact, the utility of a network corresponds to the number of links newly added in order to turn it into a clique. Let $\pi = (A_1, A_2)$ be a bipartition of A and the cost of this bipartition be defined as:

$$cost(\pi) = cut(\pi) + u(A_1) + u(A_2), \tag{3.14}$$

where $cut(\pi)$ is defined as:

$$cut(\pi) = \sum_{i \in A_1, j \in A_2} A_{ij}. \tag{3.15}$$

For a network that is already coherent enough to be a community, the cost of splitting it into two parts will be extremely costly. Based on this idea, we introduce the following new stopping criterion:

$$cost(\pi) \geq u(A). \tag{3.16}$$

That is, a subnetwork will be considered as a compact community that cannot be further divided if the cost of splitting it is greater than that without doing so. Note that the utility function and the cost function can be simultaneously computed through one matrix scanning process. Therefore, the time complexity of testing the stopping condition is also $\mathrm{O}(n+m)$.

The key steps for finding an optimal bipartition are summarized as follows, where A denotes the output of TAM, and *cut* is the optimal bipartition.

Algorithm 3.4. $cut = \mathrm{BP}(A)$

```
1. Compute the degree vector $K$ through a top-down matrix scanning;
2. Compute the vector $\mathrm{Num}_1$ through a top-down matrix scanning;
3. Compute the vector $\mathrm{Num}_2$ through a bottom-up matrix scanning;
4. Compute the vector $S$ based on $\mathrm{Num}_1$ and $\mathrm{Num}_2$;
5. Set the cut with the middlemost position with the maximum $S$-value;
6. Test the stopping condition: If true, return "cannot divide"; else return cut.
```

The matrix scanning processes in the BP algorithm are the most computationally costly steps. In terms of the adjacency list in which only nonzeros elements are stored, each scanning will take only $\mathrm{O}(n+m)$ time. Thus, the overall time required by the BP algorithm is bounded by $\mathrm{O}(n+m)$.

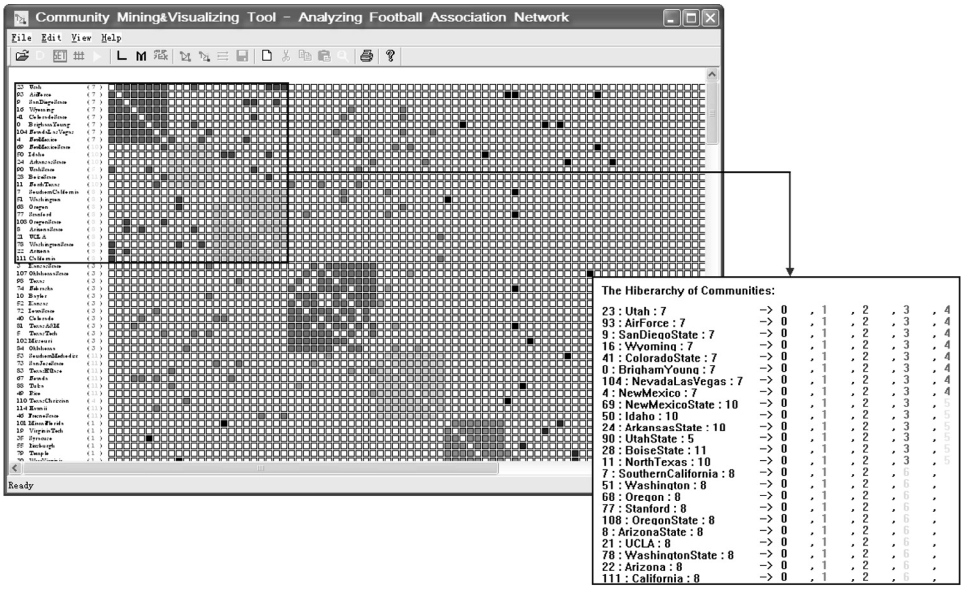
3.4 The Time Complexity of the ICS Algorithm

Let $T(A_0)$ be the time complexity of ICS and A_0 be the adjacency matrix of the network to be clustered, we have:

$$T(A_0) = \begin{cases} T_1 + T_2 + T_3 + T(A_1) + T(A_2) + T_6, & \textit{bipartition exists} \\ T_1 + T_2, & \textit{else}, \end{cases}$$

where T_1, T_2 ,T_3, and T_6 are the time required by Steps 1, 2, 3, and 6 of ICS. We have $T_3 + T_6 = \mathrm{O}(n+m)$ and $T_1 + T_2 = \mathrm{O}(t(n+m) + n\ \mathrm{long})$. Thus, the worst-case time complexity of ICS is:

$$T(A_0) = \begin{cases} \mathrm{O}(t(n\log n + m)) + T(A_1) + T(A_2), & \textit{bipartition exists} \\ \mathrm{O}t(n\log n + m), & \textit{else}. \end{cases}$$


Community Mining&Visualizing Tool - Analyzing Football Association Network
File Edit View Help
Ready

```
The Hiberarchy of Communities:
23 : Utah : 7                 -> 0   , 1   , 2   , 3   , 4
93 : AirForce : 7             -> 0   , 1   , 2   , 3   , 4
9 : SanDiegoState : 7         -> 0   , 1   , 2   , 3   , 4
16 : Wyoming : 7              -> 0   , 1   , 2   , 3   , 4
41 : ColoradoState : 7        -> 0   , 1   , 2   , 3   , 4
0 : BrighamYoung : 7          -> 0   , 1   , 2   , 3   , 4
104 : NevadaLasVegas : 7      -> 0   , 1   , 2   , 3   , 4
4 : NewMexico : 7             -> 0   , 1   , 2   , 3   , 4
69 : NewMexicoState : 10      -> 0   , 1   , 2   , 3   , 5
50 : Idaho : 10               -> 0   , 1   , 2   , 3   , 5
24 : ArkansasState : 10       -> 0   , 1   , 2   , 3   , 5
90 : UtahState : 5            -> 0   , 1   , 2   , 3   , 5
28 : BoiseState : 11          -> 0   , 1   , 2   , 3   , 5
11 : NorthTexas : 10          -> 0   , 1   , 2   , 3   , 5
7 : SouthernCalifornia : 8    -> 0   , 1   , 2   , 6   ,
51 : Washington : 8           -> 0   , 1   , 2   , 6   ,
68 : Oregon : 8               -> 0   , 1   , 2   , 6   ,
77 : Stanford : 8             -> 0   , 1   , 2   , 6   ,
108 : OregonState : 8         -> 0   , 1   , 2   , 6   ,
8 : ArizonaState : 8          -> 0   , 1   , 2   , 6   ,
21 : UCLA : 8                 -> 0   , 1   , 2   , 6   ,
78 : WashingtonState : 8      -> 0   , 1   , 2   , 6   ,
22 : Arizona : 8              -> 0   , 1   , 2   , 6   ,
111 : California : 8          -> 0   , 1   , 2   , 6   ,
```

Fig. 4. The interface of Community Mining & Visualizing Tool, in which the network that is being analyzed is the football association network. In the displayed output matrix, each dot corresponds to a "1" entry and each of the diagonal blocks with different gray degrees denotes a detected football association. This tool can also compute and visualize the hierarchical structure of all detected communities in terms of the sequence of bipartitions. For example, the label of 23:Utah:7 respectively denotes team number, team name, and association number containing this team. We can see from this example, team 23 respectively belongs to hierarchies 4, 3, 2, 1, and 0 from right to left layers, and a total of 24 teams from three different communities form a hierarchy with label 2.

In a recursive manner, one can show that:

$$T(A_0) < \mathrm{O}(Rt(nlogn + m)),$$

where R is the total number of times recursively calling ICS by Steps 4 and 5 during the course of finding all K communities. We have $R = 2K - 1$. So, the worst-case time complexity of the ICS is $\mathrm{O}(Kt(nlogn + m))$.

4. VALIDATION OF THE ICS ALGORITHM

4.1 Testing the Effectiveness of the ICS Algorithm

In this section, we will test the effectiveness of the ICS algorithm against some benchmark networks that have been commonly used in related studies. For all experiments, we set $a = 0.5$ and $\varepsilon = 10^{-4}$. All hierarchical community structures are visualized by our Community Mining & Visualizing Tool as shown in Figure 4.

4.1.1 *An Illustrative Network Example.* First of all, we use a simple network shown in Figure 5(a) to illustrate the output format of the Community Mining & Visualizing Tool. In Figures 5(b) and 5(c), the labels on the left are the node indices, followed by their corresponding community IDs. For example,

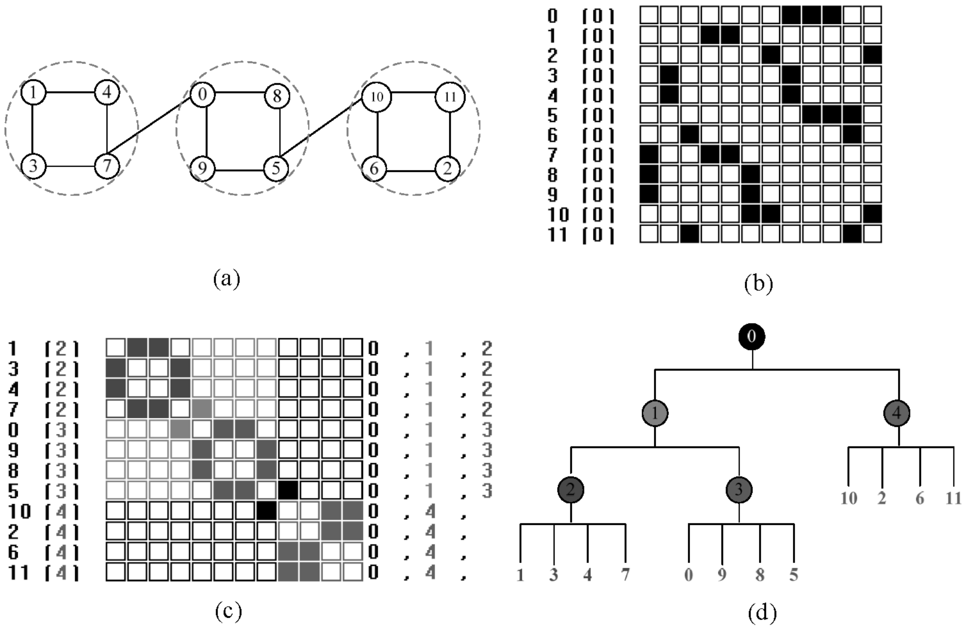


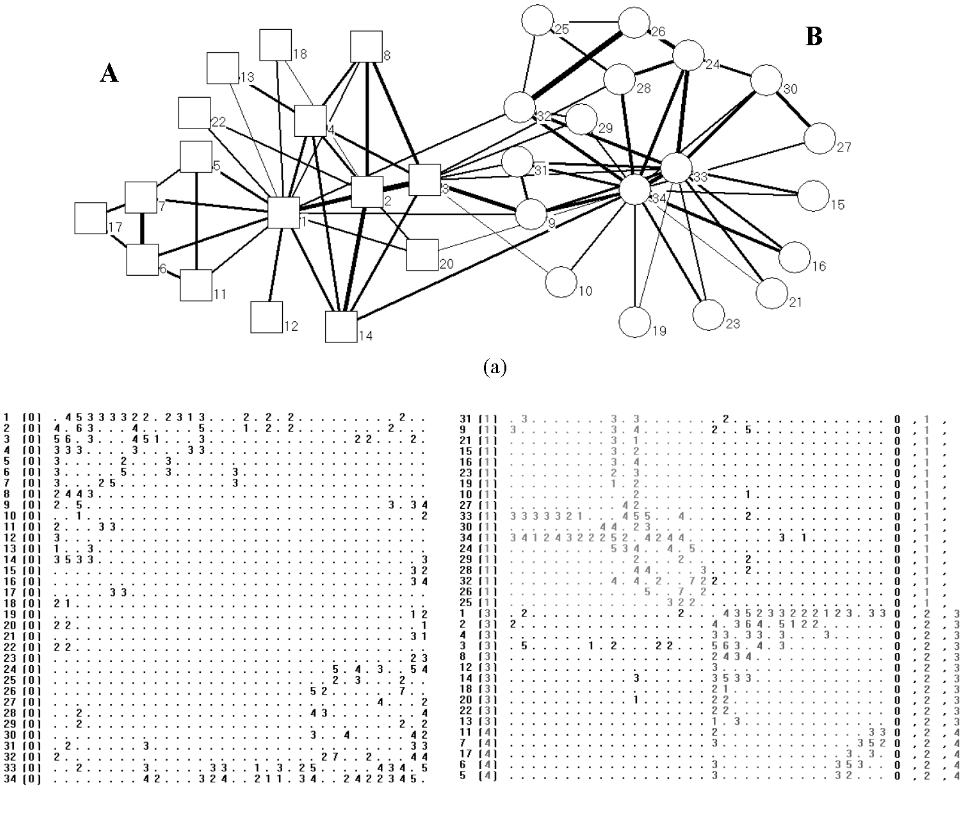
Fig. 5. The community structure of a network identified by the ICS algorithm. (a) An example network; (b) its initial adjacency matrix; (c) the output matrix; (d) the output hierarchical community structure.

in the first row of Figure 5(c), “1 (2)” denotes “node 1” belonging to “community 2.” As the ICS algorithm is, in essence, a depth-first search algorithm, the numeric labels on the right hand side of Figure 5(c) indicate the sequence of communities extracted, which also correspond to a hierarchical structure of the detected communities, as shown in Figure 5(d).

4.1.2 *The Karate Club Network.* The karate club network, as shown in Figure 6(a), describes the social interactions among the members of a karate club at an American university, which was originally constructed by Wayne Zachary in the 1970s [Zachary 1977]. The different widths of the links correspond to different interaction strengths. As reported in Zachary [1977], the club eventually split into two communities: Community A was led by its administrator (node 1) denoted by squares, and Community B by its teacher (node 33) denoted by circles.

Figure 6(c) presents the output adjacency matrix and the hierarchical community structure obtained by the ICS algorithm. We can see that the two largest groups detected are exactly identical with the real division shown in Figure 6(a).

4.1.3 *The Football Association Network.* The US college football association network [Girvan and Newman 2002] contains 115 nodes and 613 links, which correspond to football teams and games played among teams, respectively. Figure 1(a) shows its initial adjacency matrix. All teams are divided into 12 conferences. Each conference is considered as one network community since



(a)

(b)

(c)

Fig. 6. (a) The karate club network; (b) the initial adjacency matrix; (c) the output matrix and hierarchical community structure.

the number of games played within the same conference is much more than those between conferences.

Figure 7 shows the output adjacency matrix and the hierarchical community structure obtained by the ICS algorithm. In the output matrix, the string on the left of each row contains the team number, team name, and association number. Most communities are exactly identical with the real associations except for 8 teams from IA Independents (association No. 5), Western Athletic (association No. 11), and Texas Christian (association No. 4).

4.1.4 *The Dolphin Network.* The network shown in Figure 8(a) describes the social relationship of 62 bottlenose dolphins living in Doubtful Sound of New Zealand, which was first established by Lusseau based on his experimental observations of the dolphins for seven years [Lusseau 2003]. During his research studies, he found these dolphins were separated into two groups for some reasons, as shown in Figure 8(a).

Figure 8(b) shows the output adjacency matrix and the hierarchical community structure found by the ICS algorithm, in which the two biggest groups

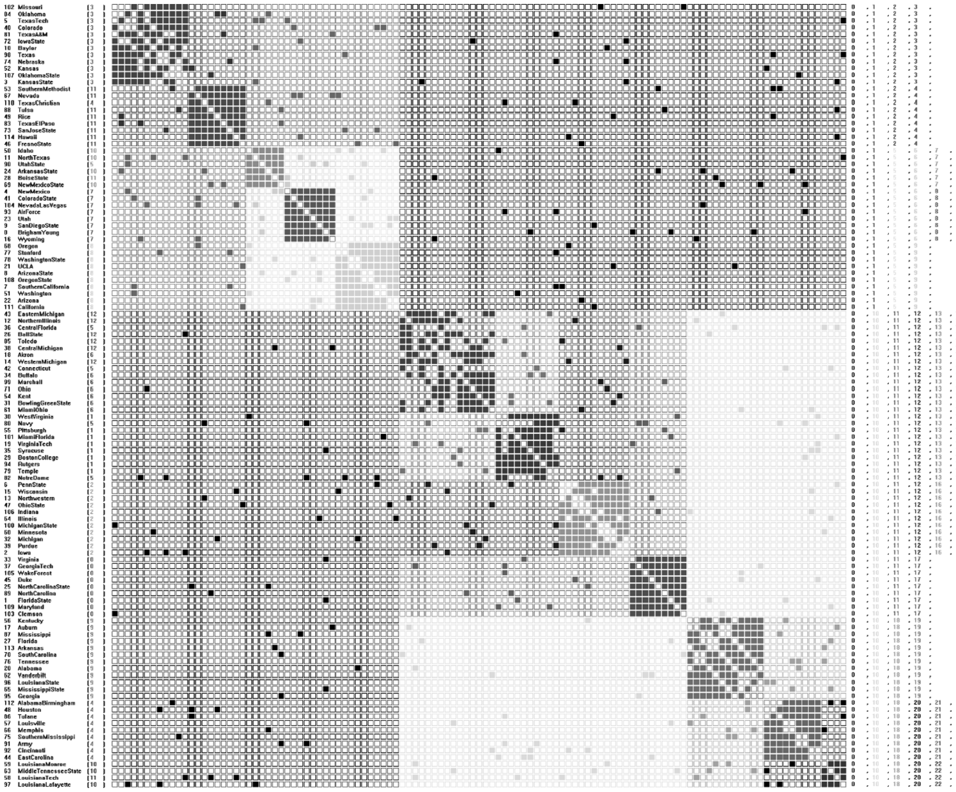


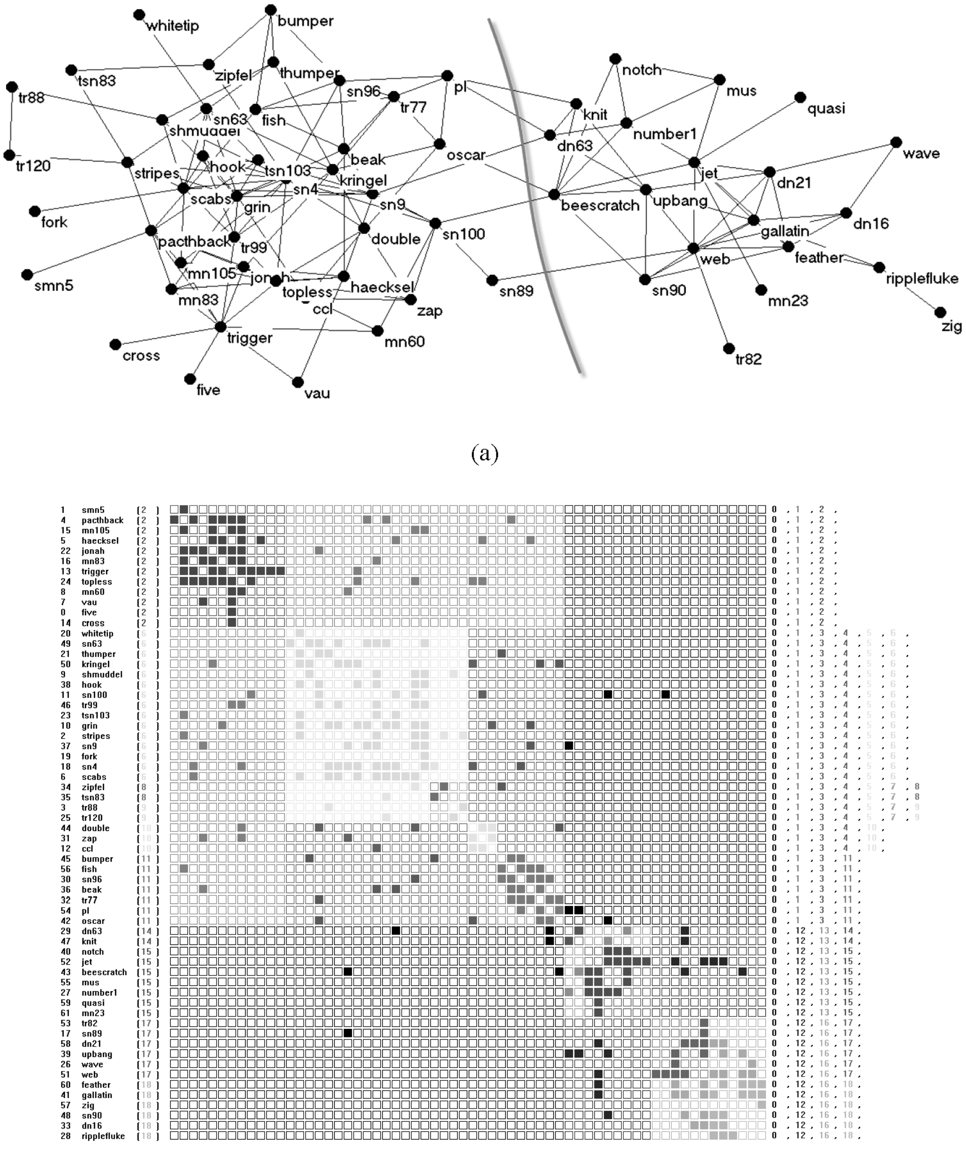
Fig. 7. The output matrix and hierarchical community structure of the football association network.

are very close to the real division except the node "sn89." In addition, the ICS algorithm predicts some potential divisions for the respective two groups, as shown in Figure 8(b).

4.2 Applying the ICS to Reduce a Complex Network

Network reduction is a useful technique for analyzing complex networks. In this section, we illustrate how to reduce a complex network into a dendrogram using the ICS. The network discussed here is obtained from the bibliography of the book *Graph Products: Structure and Recognition* [Imrich et al. 2000] (http://vlado.fmf.uni-lj.si/pub/networks/pajek/). The bibliography contains 360 papers written by 314 authors. Its corresponding network is a 2-mode graph, in which each node denotes either one person or one paper, and link (i, j) represents person i as the author of paper j, as shown in Figure 9.

Figure 10 provides the output of the ICS algorithm, in which 147 communities are detected from the above bibliography network. As we have expected, each community contains some papers and their collaborating authors. For example, community 1, as shown in Figure 10, contains 2 papers and 6 authors, corresponding to the following references:



(a)

(b)

Fig. 8. (a) The dolphin network; (b) the output matrix and hierarchical community structure.

McEliece et al. [1978] R. J. McEliece, E. Rodemich, and H. C. Rumsey, “The Lov’asz Bound and Some Generalizations,” *J. Combinatorics, Information and System Sciences*, Vol. 3, 1978, pp. 134–152.

Baumert et al. [1971] L. D. Baumert, R. J. McEliece, E. Rodemich, H. C. Rumsey, R. Stanley, and H. Taylor, “A Combinatorial Packing Problem, Computers in Algebra and Number Theory,” *Proc. SIAM-AMS Symp. Appl. Math.*,1977, pp. 97–108, American Mathematical Society, Providence, RI.

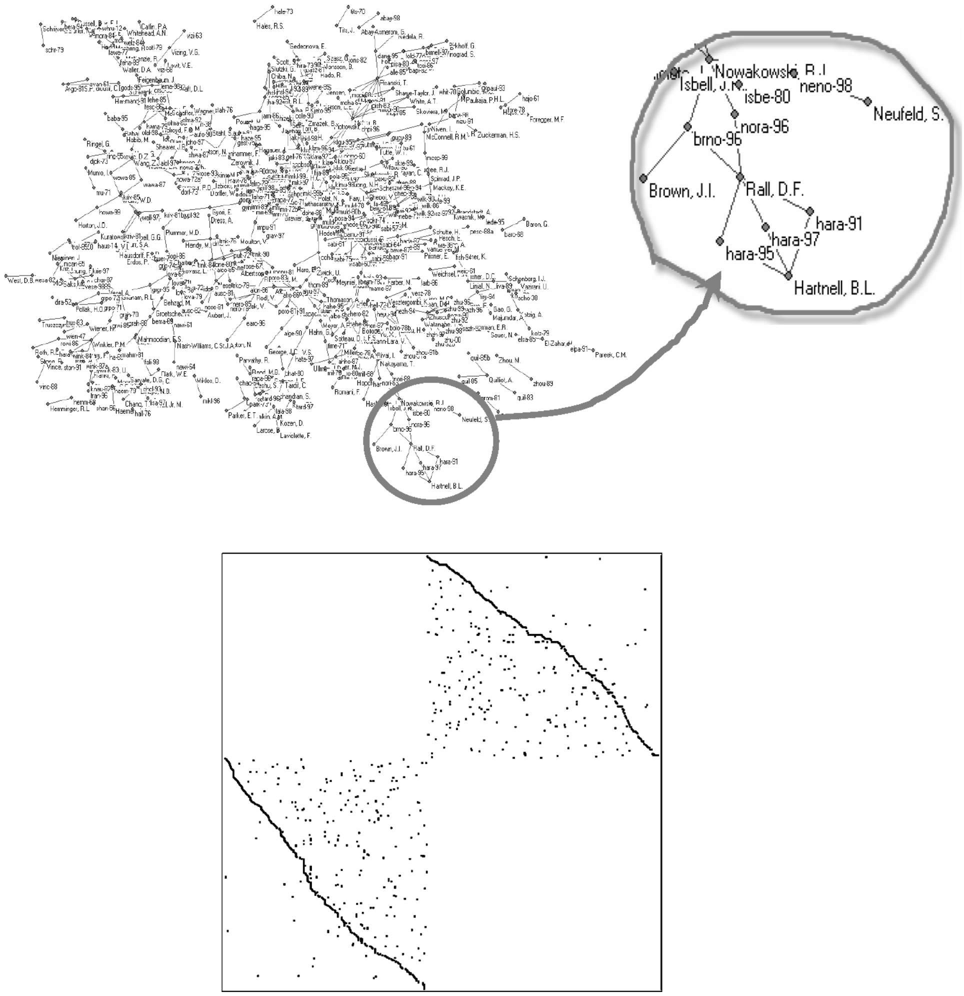


Fig. 9. The upper panel presents the bibliography network of the book "*Graph Products: Structure and Recognition* [Imrich et al. 2000]." The bottom panel shows its adjacency matrix.

Most of the detected communities are self-connected components, and component A is the biggest one, containing 13 communities, 158 papers, and 86 authors. Next, we will analyze component A in detail.

First, we transform component A into a weighted network, as shown in Figure 11(a). This network indicates the collaborations among 86 coauthors, in which link (i, j) with weight w denotes authors i and j coauthored w papers.

Then, we apply the ICS algorithm to the coauthors network and find out 14 communities as shown in Figure 11(b), in which different gray degrees indicate different clusters.

Furthermore, we reduce the clustered coauthors network into a much smaller weighted network by condensing each community as one node, as shown in Figure 11(c).

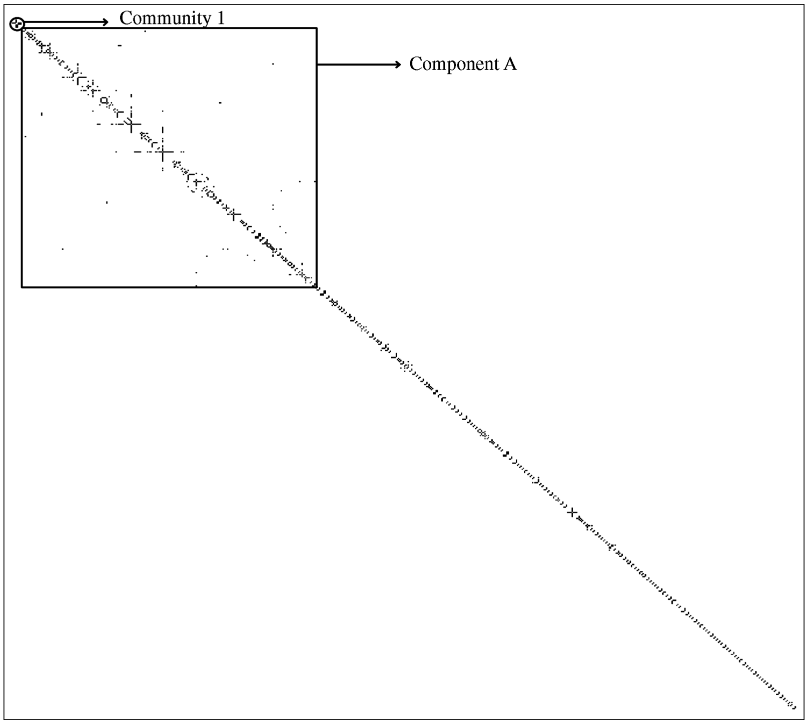


Fig. 10. The outputs of the ICS algorithm, when applied to the bibliography network.

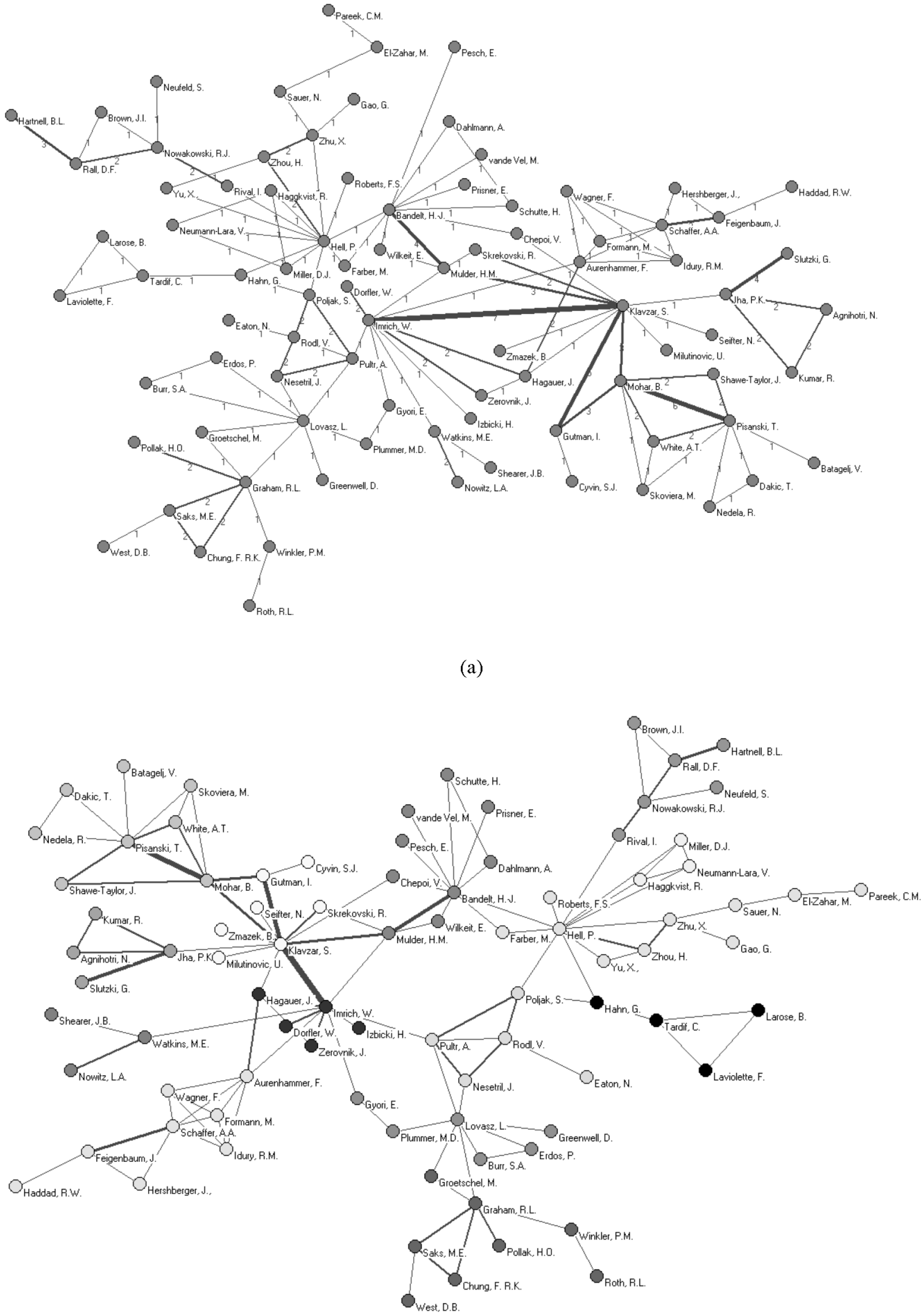
Again, the ICS algorithm is applied to obtain the top-level network using the same way, as shown in Figure 11(d).

Finally, a dendrogram corresponding to component A, as shown in Figure 11(e), is built based on the results obtained in the above steps.

4.3 Mining Web Communities

Web communities are collections of highly topic-related Web pages. The ability to automatically cluster Web communities based on the link structure is significant for improving the efficiency of search engines because clustering Web pages in terms of their reciprocally referenced relationships is computationally much cheaper than clustering them in terms of their semantic contents. In this section, we will show how to identify Web communities from a given sub-Web network using the ICS algorithm.

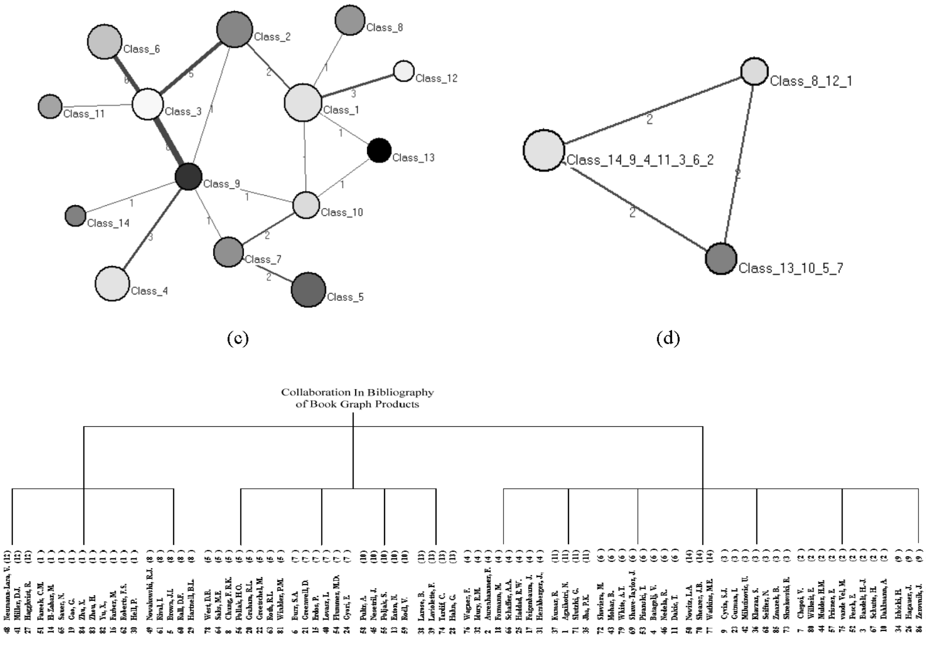
We have adopted and tested the experimental dataset as obtained from http://www.cs.toronto.edu/∼tsap/experiments/download/download.html. This dataset was originally constructed according to the following steps: (1) query search engine AltaVista with the keyword Jordan and form the root set using the first 200 returned pages; (2) enlarge the root set into the "base set" by taking in all out-links and first 50 in-links of those pages in the root set; (3) based on the base set, construct the underlying network by specifying pages as nodes and in-links and out-links between pages as directed links.



(a)

(b)

Fig. 11. (a) The coauthors network of the biggest component; (b) the clustered coauthors network; (c) the condensed coauthors network; (d) the top-level coauthors network; (e) the dendrogram of the coauthors network (*continues*).



(c)

(d)

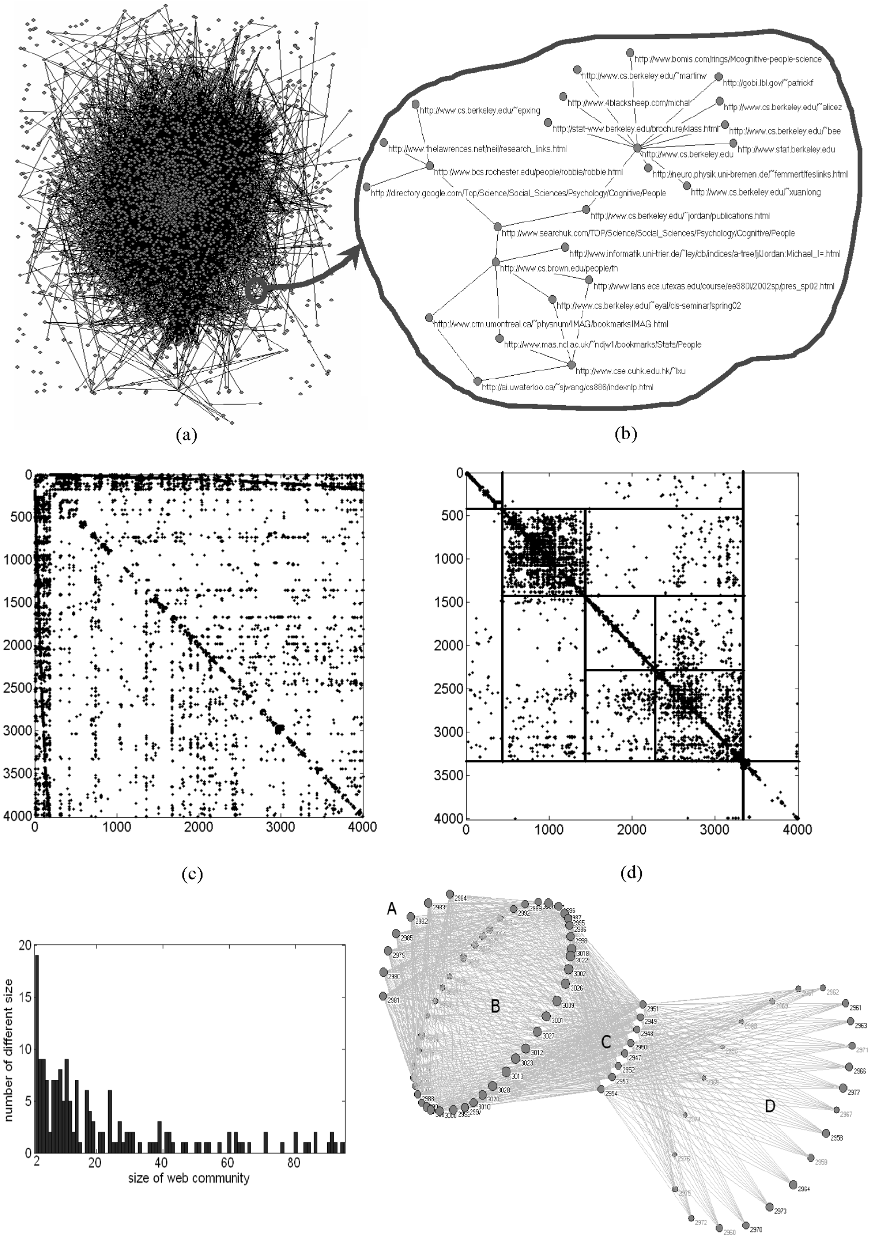
(e)

Fig. 11. (*Continued*).

Figure 12(a) shows the network structure of the sub-Web network constructed by the above steps, and Figure 12(b) shows the enlarged view of the circled area in this large network. Figure 12(c) presents its original adjacency matrix in which each black dot denotes a link. Note that the matrix is symmetric since we have ignored the directions of the links and have focused only on their density.

The ICS algorithm takes 490 seconds to find all Web communities in this network, with a personal computer of 1.8Hz CPU and 512MB memory. Figure 12(d) presents the output of ICS. We can see that five biggest groups are detected and the links between the groups are much fewer than those within the groups. In total, 173 communities have been detected and the average size of communities is 23. Figure 12(e) provides the community size distribution in terms of histogram charts which indicate how many communities in the network have a certain number of nodes. Approximately, we can see a power-law distribution emerging; that is, most communities have a small size, while a small number of communities contain quit many members.

As an example, we have selected and looked into a compound community that contains 80 Web pages. This community is composed of four groups, denoted as A, B, C, and D in Figure 12(f). It looks like a multiple-hub organization, in which groups B and D are associated together by two groups of hubs. Group B links to both hub-groups A and C, while group D only links to hub-group C. Interestingly, most pages in groups B and D are ESPN Web sites, as related to the NBA events, news, stories or people, while most of the pages in groups



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 12. The experimental results of the ICS algorithm, as applied to a sub-Web with 4,009 pages. (a) The sub-Web network; (b) the enlarged view of a circled area in (a); (c) initial adjacency matrix; (d) output of the ICS algorithm; (e) statistic data regarding the identified communities; (f) one example community.

Table I. Time Complexity of Some Related Algorithms

Algorithms	Time complexity	
	Two-way partition	K-way partition
ICS algorithm	$O(t(n\log n + m))$	$O(Kt(n\log n + m))$
Wu-Huberman algorithm [Wu et al. 2004]	$O(r(n + m))$	$O(Kr(n + m))$
Newman algorithm [Newman 2004b]	$O(n - 2)(n + m)$	$O((n - K)(n + m))$
Kernighan-Lin algorithm [Kernighan and Lin 1970]	$O(n^2)$	$O(Kn^2)$
Radicchi algorithm [Radicchi et al. 2004]	$O(m^3/n^2)$	$O(Km^3/n^2)$
GN algorithm [Girvan and Newman 2002]	$O(m^2 n)$	$O(Km^2 n)$
Spectral method [Fiedler 1973; Pothen et al. 1990; Shi and Malik 2000]	$O(m/(\lambda_3 - \lambda_2))$	$O(Km/(\lambda_3 - \lambda_2))$

A and C are related to MSN portals, such as MSN welcome, MSN hotmail, MSN money, MSN search, MSN shopping, or MSN people and chat. Detailed information of each page can be found in Appendix 1. That means quite a few Web pages belonging to the ESPN Web sites are organized together through the portals of another company. Based on this observation, it is reasonable to infer the partnership or intensive collaborations between these two involved companies.

5. DISCUSSIONS

5.1 The Time Complexity of Different Algorithms

The time complexity of some existing algorithms is presented in Table I, where m and n denote the numbers of links and nodes, λ_2 and λ_3 denote the second and third smallest eigenvalues of Laplacian matrix of a given network, t and r denote the iterations required by different algorithms. In essence, both the ICS and the GN algorithms are inspired from the concept of node centrality. As discussed in Section 3.2, the TAM algorithm is based on *clustering centrality,* which is an extension of *eigenvector centrality* [Bonacich 1972, 1987]. The GN algorithm is based on *link betweenness centrality* [Girvan and Newman 2002], which is an extension of *node betweenness centrality* [Freeman 1977]. However, calculating link betweenness is very time-consuming, and so far the fastest algorithms take $O(nm)$ time to calculate all link betweenness for a network [Newman 2001; Brandes 2001]. In contrast, calculating clustering centralities for all nodes is quite efficient. As discussed, the centrality distribution of a network can be computed within an approximate time of $O(n\log n + m)$. This is the major reason why ICS is more efficient.

5.2 Insensitivity to Built-in Parameters and Prior Knowledge

In essence, the ICS algorithm is a kind of bisection method that finds all network communities through a series of bipartitions. As discussed in Section 1, most of the existing bisection methods heavily depend on prior knowledge, such as the number of communities, the appreciate size of each community, and so on. Different from them, ICS can discover all natural communities by means of some predefined stopping conditions rather than such prior knowledge.

Table II. Parameters or Prior Knowledge as Required by Different Algorithms

Categories	Algorithms	Parameters or prior knowledge
Bisection methods	ICS algorithm	temporal coefficient, controlling error
	Kernighan-Lin algorithm [Kernighan and Lin 1970]	number of communities, approximate size of each community
	Spectral method [Fiedler 1973, Pothen et al. 1990; Shi and Malik 2000]	number of communities, or threshold of cut score
	Wu-Huberman algorithm [Wu et al. 2004]	number of communities, two nodes belonging to different communities, approximate size of each community
Hierarchical methods	GN algorithm [Girvan and Newman 2002]	knowledge of the dendrogram it produces
	Newman algorithm [Newman 2004b]	knowledge of the dendrogram it produces
	Radicchi algorithm [Radicchi et al. 2004]	cycle of order, knowledge of the dendrogram it produces
Web communities mining methods	MFC algorithm [Flake et al. 2002]	sink pages, number of communities
	HITS algorithm [Kleinberg 1999]	query results of a search engine, number of top pages
	SAE algorithm [Pirolli et al. 1996]	query results of a search engine, relaxing rate, spreading rate

Table II sums up the built-in parameters or prior knowledge as required by different algorithms. Compared to the hierarchical methods, the bisection methods and the methods for detecting Web communities involve more parameters and prior knowledge.

On the other hand, the ICS algorithm involves only two parameters: temporal coefficient a and controlling error ε. The result of the TAM algorithm is insensitive to ε, if its value is set to be small enough. Theoretically, we can set ε as small as we can. But, as restricted by the computing precision of practical software, a moderate value between 10^{-5} and 10^{-3} is suitable. As discussed before, the performance, such as the speed and accuracy of ICS, is related, but insensitive, to the temporal coefficient. As Figure 2 shows, the larger a value we set, the quicker the TAM runs. As Figure 13 shows, the smaller the a value we set, the higher the clustering accuracy the ICS can achieve. In practical applications, the temporal coefficient is set as 0.5, in order to obtain a good tradeoff between speed and accuracy.

5.3 Clustering Accuracy of the ICS Algorithm

Figure 13 presents the clustering accuracy comparison of three algorithms. This experimental method has been widely adopted by other related studies [Girvan and Newman 2002; Newman 2004b; Radicchi et al. 2004]. The networks used here are the computer-generated, random networks. In each random network, there are 4 communities with size 32. Each node in a community, on average, emits 16 links. z_{out} is the number of inter-community links. Obviously, as z_{out} increases, the community structure of a random network becomes more and

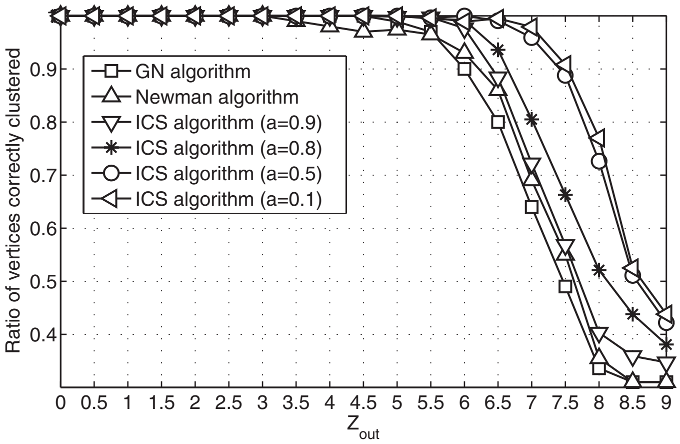


Fig. 13. Testing the clustering accuracy of three different algorithms.

more ambiguous. A clustering is correct, if it precisely detects the original four communities. In Figure 13, y-axis denotes the ratio of vertices correctly clustered by different algorithms, and each data point in the curves is obtained by running a specified algorithm over 100 such random networks.

We note that all algorithms work well when $z_{\text{out}} \leq 5.5$, correctly identifying more than 95% of nodes. In the case of $6 \leq z_{\text{out}} \leq 9$, the clustering accuracy of the ICS algorithm is better than the GN algorithm and Newman algorithm. We also note that the accuracy of ICS is related to the temporal coefficient, that is, the accuracy slightly decreases as it increases.

5.4 A Comparison with Spectral Methods

Similar to spectral methods, the ICS algorithm can be considered as one kind of recursive bisection approach that partitions a network by a series of recursive bipartitions. Generally speaking, a recursive bisection approach has three distinct features; namely, a global objective function, a strategy for performing the bisection operations, and a criterion for terminating the recursion. However, it should be pointed out that the ICS and spectral methods differ in their implementations of such features.

In spectral methods, the objective functions are based on various 'cut' scores, such as average cut or normalize cut [Fiedler 1973; Pothen et al. 1990; Shi and Malik 2000]. They bisection a network by minimizing a constraint quadratic function defined as X^TMX/X^TX. The optimal X is the second smallest eigenvector of M. M is equal to D–A in the case of average cut or $D^{-1/2}(D-A)D^{-1/2}$ in the case of normalized cut. A recursive spectral bisection will stop if no component has a cut whose score is below a predefined threshold.

On the other hand, in the ICS algorithm, the objective function is based on the community criterion as defined in Equation (2.1) or Equation (3.11). It divides a network into two by computing, sorting, and splitting the clustering centrality distribution of the network. Its recursive stopping condition is based on Equation (3.16).

Like the ICS, in practice, spectral methods can run quite fast when we use a heuristic method, such as Lanczos algorithm, to calculate an approximate second smallest eigenvector of M.

5.5 A Comparison with Cross-Association Methods

As Chakrabarti et al. [2004] defined, a cross-association is "a joint decomposition of a binary matrix into disjoint row and column groups such that the rectangular intersections of groups are homogeneous." A rectangular intersection is considered to be homogeneous if most of its entries are the same. An intersection will be called "0-homogeneous" if most of its entries are "0." Otherwise, it will be called "1-homogeneous." In some applications, such as market data analysis, the ability to mine the cross-association of a given matrix helps us reveal the hidden patterns of the relationship between objects, such as the frequent item sets of a market basket data. Several studies are related to this interesting topic, such as the ITCC algorithm [Dhillon et al. 2003]. Recently, a novel method, the CA algorithm, based on the information theory is presented by Chakrabarti et al. [2004], which uses the Shannon entropy as a new criterion to evaluate the homogeneous degree of a given matrix. The CA algorithm is an efficient method to transform a given matrix into a new one with an approximately optimal homogeneous layout in terms of a predefined cost function based on the Shannon entropy.

The CA algorithm and the ICS algorithm are similar because both cases involve matrix transforming operations. However, they are essentially distinct from each other due to the following aspects.

First, the theoretical foundations behind them are different. The criterion adopted by the CA algorithm is based on the Shannon entropy, and it attempts to find an approximately optimal solution using a local search method. However, the basic idea behind the ICS algorithm is to find a global community structure based on local centrality information.

Second, the CA algorithm is limited to dealing with the binary matrix containing only 0 and 1 because we cannot use the Shannon entropy to measure the homogeneous degree for a weighted matrix.

Finally and most importantly, cross-association and community are two different concepts in essence. Especially, we can note that the community structure is a kind of cross-association with stricter constraints. For the CA algorithm, what it concerns is to identify such a cross-association in which each intersection is homogeneous in spite of 0-homogeneous or 1-homogeneous. However, for the ICS algorithm, it tries to find such a cross-association in which each rectangular intersection is homogeneous, and at the same time, the intersections corresponding to communities are 1-homogeneous and the rest are 0-homogeneous as shown in Figure 14(a). Without considering those constrains, the CA algorithm might identify the cross-association as shown in Figure 14(b).

Figure 14(a) shows the cross-association layout obtained by the ICS algorithm, in which three communities, denoted by C_1, C_2, and C_3, are detected. Note that, among the 9 cross-intersections, the rectangles corresponding to communities are all 1-homogeneous, and the rest are all 0-homogeneous. This

	C_1	C_2	C_3
C_1	1	0	0
C_2	0	1	0
C_3	0	0	1

(a)

	C_1	C_2	C_3
C_1	0	0	1
C_2	1	0	0
C_3	0	1	1

(b)

Fig. 14. An example to show the distinction between the problems of finding cross-associations and discovering communities. In the two matrices, the rectangle with label "1" denotes a 1-homogeneous intersection and the rectangle with label "0" denotes a 0-homogeneous intersection.

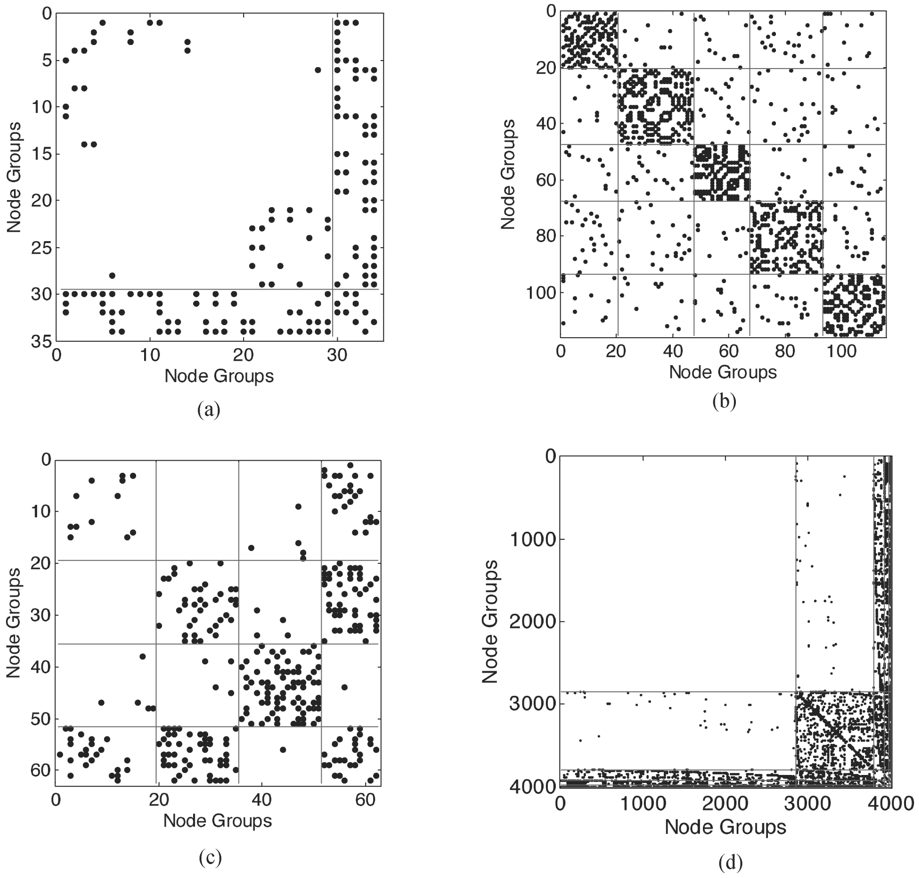
kind of cross-association implies dense links within communities and sparse links between communities. Figure 14(b) shows one possible cross-association layout obtained by the CA algorithm, in which also 9 cross-intersections are detected. But this time, the rectangles corresponding to communities are mostly 0-homogeneous, and three noncommunities are 1-homogeneous. The kind of cross-association indicates sparse links within communities and dense links between communities, which obviously is not what we really want according to the definition of community.

This difference can be clearly observed through real networks as shown in Figure 15. We can see the CA algorithm generates quite nice partitions for all networks with respect to the cross-association criterion. However, all these partitions are far from real community structures, because of the reason as mentioned above. In the case of the football association network, the cross-association layout found by the CA algorithm looks like a community structure, but it is not yet accurate enough as compared to the real network partition with 12 communities.

6. CONCLUSIONS

In this article, we have presented a new algorithm, called ICS, for discovering global network community structures based on local centralities of network nodes. The key concept proposed in our work is clustering centrality, based on which we have developed the TAM algorithm that can quickly rearrange all nodes into a new permutation according to their clustering centrality distribution. This permutation output by TAM shows a very useful feature, that is, the nodes within the same partition will be aggregated together. Therefore, one can nicely bipartition a network using an exact or a fuzzy constraint. The two subnetworks will be further refined by means of recursions until all hidden communities are found.

We have tested ICS against different kinds of networks, such as social networks, random networks, and Web page networks. Our experimental results have shown its good performance with respect to both speed and accuracy. Its major features include: (1) it is efficient with a running time scaleable to network sizes; (2) it is effective and, for most real networks, it can find the community structures similar or identical to real divisions; (3) it is insensitive to its built-in parameters and requires no prior knowledge; and (4) it can output a hierarchical structure of all communities.



(a)

(b)

(c)

(d)

Fig. 15. Partition results of the CA algorithm against four real networks discussed in Section 4. (a) The karate club network; (b) the football association network; (c) the dolphin network; (d) a subweb obtained by querying Alta Vista with the keyword "Jordan."

In our future work, we will focus our attention on studying how to extend the ICS algorithm into a decentralized algorithm being able to process distributed and dynamic networks.

APPENDIX

The node information in terms of (Node ID, web site, topic) of the Web community shown in Figure 12(f).

2947 http://g.msn.com/0nwenus0/AK/01 Welcome to MSN.com
2948 http://g.msn.com/0nwenus0/AK/02 MSN Hotmail
2949 http://g.msn.com/0nwenus0/AK/03 MSN Search—More Useful Everyday
2950 http://g.msn.com/0nwenus0/AK/04 Welcome to MSN Shopping
2951 http://g.msn.com/0nwenus0/AK/05 MSN Money—More Useful Everyday
2952 http://g.msn.com/0nwenus0/AK/06 MSN People and Chat—More Useful Everyday

2953 http://g.msn.com/0nwenus0/AK/14 Welcome to MSN.com

2954 http://espn.go.com ESPN.com

2955 http://proxy.espn.go.com/keyword/searchResults?search=Michael+Jordan&searchType=2&site=espn&CMP=IL20 ESPN.com Search Results for: Michael Jordan

2956 http://proxy.espn.go.com/keyword/searchResults?search=Michael+Jordan&searchType=1&CMP=IL20269 ESPN.com Search Results for: Michael Jordan

2957 http://insider.espn.go.com/insider/story?id=1554609 ESPN Insider: ESPN Insider: Benefits

2958 http://insider.espn.go.com/insider/story?id=1554627 ESPN Insider: ESPN Insider: Benefits

2959 http://insider.espn.go.com/insider/story?id=1554622 ESPN Insider: ESPN Insider: Benefits

2960 http://sports.espn.go.com/nba/clubhouse?team=bos ESPN.com: NBA Boston Celtics Clubhouse

2961 http://sports.espn.go.com/nba/boxscore?gameId=230409027 ESPN.com—NBA—Boston Celtics at Washington Wizards Live NBA Box Score on ESPN.com

2962 http://sports.espn.go.com/nba/clubhouse?team=mia ESPN.com: NBA Miami Heat Clubhouse

2963 http://sports.espn.go.com/nba/boxscore?gameId=230411014 ESPN.com—NBA—Washington Wizards at Miami Heat Live NBA Box Score on ESPN.com

2964 http://sports.espn.go.com/nba/clubhouse?team=atl ESPN.com: NBA Atlanta Hawks Clubhouse

2965 http://sports.espn.go.com/nba/boxscore?gameId=230412027 ESPN.com—NBA—Atlanta Hawks at Washington Wizards Live NBA Box Score on ESPN.com

2966 http://sports.espn.go.com/nba/clubhouse?team=nyk ESPN.com: NBA New York Knicks Clubhouse

2967 http://sports.espn.go.com/nba/boxscore?gameId=230414027 ESPN.com—NBA—New York Knicks at Washington Wizards Live NBA Box Score on ESPN.com

2968 http://sports.espn.go.com/nba/clubhouse?team=phi ESPN.com: NBA Philadelphia 76ers Clubhouse

2969 http://sports.espn.go.com/nba/boxscore?gameId=230416020 ESPN.com—NBA—Washington Wizards at Philadelphia 76ers Live NBA Box Score on ESPN.com

2970 http://espn.go.com/sitetools/s/help ESPN.com: SITETOOLS—ESPN.com Help

2971 http://espn.go.com/mediakit ESPN.com: MEDIAKIT—Media Kit Home

2972 http://espn.go.com/sitetools/s/contact ESPN.com: SITETOOLS—Contact ESPN

2973 http://espn.go.com/sitetools/s/tools ESPN.com: SITETOOLS—Tools

2974 http://espn.go.com/sitetools/s/sitemap ESPN.com: SITETOOLS—Site Map

2975 http://espn.go.com/sitetools/s/terms.html ESPN.com: SITETOOLS—Terms of Service

2976 http://espn.go.com/sitetools/s/privacy.html ESPN.com: SITETOOLS—Privacy

2977 http://espn.go.com/sitetools/s/help/jobs.html ESPN.com Job Opportunities

2979 http://g.msn.com/0nwenus0/AK/08 Welcome to MSN.com

2980 http://g.msn.com/0nwenus0/AK/09 MSN Hotmail

2981 http://g.msn.com/0nwenus0/AK/10 MSN Search—More Useful Everyday

2982 http://g.msn.com/0nwenus0/AK/11 Welcome to MSN Shopping

2983 http://g.msn.com/0nwenus0/AK/12 MSN Money—More Useful Everyday

2984 http://g.msn.com/0nwenus0/AK/13 MSN People and Chat—More Useful Everyday

2985 http://espn.go.com/nba/s/2003/0226/1514649.html ESPN.com: NBA—Iverson gives NBA new sort of credibility

2986 http://espn.go.com/nba/playoffs2002/s/frozenmoment4.html ESPN.com—2002 NBA Finals - Bryant's 3 sparked Lakers to three-peat

2987 http://espn.go.com/nba/columns/lawrence/1313028.html ESPN.com: NBA—Big game? Jordan in Chicago is meaningless

2988 http://espn.go.com/nba/columns/ratto_ray/1421868.html ESPN.com: NBA—This is not what Dream Teams are made of

2989 http://espn.go.com/page2/s/wiley/011004.html ESPN.com—Page2—The 'Skins are Schott

2990 http://espn.go.com/nba/columns/stein/1319557.html ESPN.com: NBA—Brand among those dissed for wrong reason

2991 http://espn.go.com/nba/columns/misc/1496957.html ESPN.com: NBA—Wizards making a point with Hughes

2992 http://espn.go.com/page2/s/questions/bellamy.html ESPN.com—Page2—Bill Bellamy

2993 http://espn.go.com/nba/news/2002/0107/1307442.html ESPN.com: NBA—Jordan's wife Juanita files for divorce

2994 http://espn.go.com/nba/preview2002/columns/ramsay_drjack/1452621.html ESPN.com: NBA—A cure-all for all 29 NBA teams

2995 http://espn.go.com/talent/danpatrick/s/2002/0412/1367309.html Dan Patrick:And the winner is ...

2996 http://espn.go.com/nba/columns/walton_bill/1481670.html ESPN.com: NBA—Spreading some Christmas cheer (and jeer)

2997 http://espn.go.com/page2/s/questions/yamaguchi.html ESPN.com—Page2—Kristi Yamaguchi

2998 http://espn.go.com/nba/columns/aldridge_david/1440531.html ESPN.com: NBA—Crystal basketball: 15 things that could happen

2999 http://espn.go.com/page2/s/whitlock/021010.html ESPN.com—Page2—Barry easily outslugs the Babe

3000 http://espn.go.com/nba/columns/walton_bill/1510753.html ESPN.com: NBA—Plenty of Presidents' Day pondering

3001 http://espn.go.com/nba/columns/ramsay_drjack/1476123.html ESPN.com: NBA—LeBron's best individual quality: team play

3002 http://sports.espn.go.com/nba/players/profile?playerId=1035&amp;avg=48 ESPN.com: Michael Jordan

3003 http://sports.espn.go.com/nba/teamstats?team=was ESPN.com: NBA Washington Wizards Team Statistics

3004 http://espndeportes.espn.go.com/nba/deportes/clubhouse?team=was ESPNdeportes.com: NBA—Washington Wizards

3006 http://espn.go.com/nba/playoffs2002/columns/bembry_jerry/1394263.html ESPN.com—2002 NBA Finals - Shaq, Kobe, Phil make Lakers perennial favorites

3007 http://espn.go.com/nba/preview2002/columns/aldridge_david/1452455.html ESPN.com: NBA—Young talent who can determine a franchise's fate

3008 http://espn.go.com/page2/s/rosen/021002.html ESPN.com—Page2—NBA's summer of discontent

3009 http://espn.go.com/nba/columns/ramsay_drjack/1434127.html ESPN.com: NBA—My secrets to NBA head coaching success

3010 http://espn.go.com/nba/columns/aldridge_david/1427992.html ESPN.com: NBA—Solving USA Basketball's long list of problems

3011 http://espn.go.com/page2/s/wiley/020219.html ESPN.com—Page2—Payback is a bitch

3012 http://espn.go.com/nba/columns/walton_bill/1482822.html ESPN.com: NBA—Hoping these wishes come true in 2003

3013 http://dmoz.org/Sports/Basketball/Professional/NBA/Players/J/Jordan,_Michael Open Directory—Sports: Basketball: Professional: NBA: Players: J: Jordan, Michael

3015 http://espn.go.com/nba/playoffs2002/columns/aldridge_david/1392640.html ESPN.com—NBA—PLAYOFFS2002—Let's hope NBA continues with fluid style

3016 http://espn.go.com/nba/columns/stein_marc/1510267.html ESPN.com: NBA—No fines? No trades? Not for Cuban ... yet

3017 http://espn.go.com/dickvitale/vcolumn010924jordan.html ESPN.com—Dick Vitale—vcolumn010925jordan

3018 http://espn.go.com/page2/s/wiley/020530.html ESPN.com—Page2—It's crunch time ... C-Webb disappear

3019 http://espn.go.com/page2/s/wiley/020627.html ESPN.com - Page2—Uncensored thoughts about NBA draft

3020 http://espn.go.com/nba/columns/stein_marc/1521167.html ESPN.com: NBA—MJ shouldn't be blowing stack at Stackhouse

3021 http://sports.espn.go.com/nba/teamsched?team=was ESPN.com: NBA Washington Wizards Team Schedule

3022 http://espn.go.com/page2/s/closer/020212.html ESPN.com—Page2—Love Triangle: Michael, Phil and Kobe

3023 http://espn.go.com/page2/s/wiley/020919.html ESPN.com—Page2—Polished NFL outshines NBA

3024 http://espn.go.com/nba/allstar/2003/news/2003/0209/1506552.html ESPN.com: NBA—A star among stars: Garnett earns All-Star MVP

3025 http://espn.go.com/nba/steinline/030217.html ESPN.com: NBA—The Stein Line
3026 http://espn.go.com/nba/camp2002/columns/stein_marc/1447644.html ESPN.com: NBA—Lakers already pumped up over four-peat
3027 http://espn.go.com/nba/news/2002/0911/1430627.html ESPN.com: NBA—Stackhouse dealt to Wizards in six-player deal
3028 http://espn.go.com/page2/s/rosen/021115.html ESPN.com—Page2—Stockton, Malone master fading away
3029 http://espn.go.com/page2/wash/s/questions/ptiguys.html ESPN.com—Page2—10 Burning Questions for 'PTI' duo
3030 http://games.espn.go.com/cgi/fba/request.dll?PLAYERCARD&amp;nPlayerID=1035 Fantasy Basketball: Error

ACKNOWLEDGMENTS

We would like to express our thanks to the anonymous reviewers of our article for their constructive comments and suggestions that have helped us in further improving the article.

REFERENCES

Bonacich, P. F. 1972. Factoring and weighting approaches to status scores and clique identification. *J. Math. Sociol. 2*, 113–120.

Bonacich, P. F. 1987. Power and centrality: A family of measures. *Amer. J. Sociol. 92*, 1170–1182.

Brandes, U. 2001. A faster algorithm for betweenness centrality. *J. Mathe. Sociol. 25*, 163–177.

Chakrabarti, D., Papadimitriou, S., Modha, D. S., and Faloutsos, C. 2004. Fully automatic cross-associations. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Seattle, WA. 79–88.

Chakrabarti, S., van der Berg, M., and Dom, B. 1999. Focused crawling: A new approach to topic-specific Web resource discovery. In *Proceedings of the 8th International Conference on World Wide Web*. Toronto, Canada. Elsevier North-Holland. 1623–1640.

Dhillon, I. S., Mallela, S., and Modha, D. S. 2003. Information-theoretic co-clustering. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Seattle, WA. 89–98.

Fiedler, M. 1973. Algebraic connectivity of graphs. *Czech. Math. J. 23*, 298–305.

Flake, G.W., Lawrence, S., Giles, C. L., and Coetzee, F. 2002. Self-organization and identification of Web communities. *IEEE Comput. 35*, 66–71.

Freeman, L. C. 1977. A set of measures of centrality based upon betweenness. *Sociometry 40*, 35–41.

Girvan, M. and Newman, M. E. J. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences 99*, 7821–7826.

Hotelling, H. 1936. Simplified calculation of principal component. *Psychometrika 1*, 27–35.

Imrich, W. and Klavzar, S. 2000. *Product Graphs: Structure and Recognition*, John Wiley.

Kernighan, B. W. and Lin, S. 1970. An efficient heuristic procedure for partitioning graphs. *Bell Syst. Tech. 49*, 291–307.

Kleinberg, J. M. 1999. Authoritative sources in a hyperlinked environment. *J. ACM 46*, 604–632.

Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. 1999. Trawling the Web for emerging cyber communities. In *Proceedings of the 8th International Conference on World Wide Web*. Toronto, Canada. Elsevier North-Holland. 1481–1493.

Liu, J., Jin, X. L., and Tsui, K. C. 2004. *Autonomy Oriented Computing (AOC): From Problem Solving to Complex Systems Modeling*. Kluwer Academic Publishers.

Liu, J., Jin, X. L., and Tsui, K. C. 2005. Autonomy oriented computing (AOC): Formulating computational systems with autonomous components. *IEEE Trans. Syst., Man, Cybern. Part A: Syst. Humans 35*, 879–902.

Lusseau, D. 2003. The emergent properties of a dolphin social network. In *Proceedings of the Royal Society of London B 270*, S186–S188.

Newman, M. E. J. 2001. Scientific collaboration networks II: Shortest paths, weighted networks, and centrality. *Phys. Rev. E 64*, 016132.

Newman, M. E. J. 2004a. Detecting community structure in networks. *European Phys. J. B 38*, 321–330.

Newman, M. E. J. 2004b. Fast algorithm for detecting community structure in networks. *Phys. Rev. E, 69*, 066133.

Newman, M. E. J. and Girvan, M. 2004. Finding and evaluating community structure in networks. *Phys. Rev. E 69*, 026113.

Pirolli, P., Pitkow, J., and Rao, R. 1996. Silk from a sow's ear: Extracting usable structures from the Web. In *Proceeding of Human Factors in Computing Systems*. Vancouver, Canada. ACM Press, New York. 118–125.

Pothen, A., Simon, H., and Liou, K. P. 1990. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl. 11*, 430–452.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D. 2004. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences 101*, 2658–2663.

Reichardt, J. and Bornholdt, S. 2004. Detecting fuzzy community structure in complex networks with a Potts model. *Phys. Rev. Lett. 93*, 218701.

Scott, J. 2000. *Social Network Analysis: A Handbook.* Sage Publications, London, UK.

Shi, J. and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Trans. Patt. Anal. Mach. Intell. 22*, 888–904.

Tyler, J. R., Wilkinson, D. M., and Huberman, B. A. 2003. Email as spectroscopy: automated discovery of community structure within organizations. In *Proceeding of 1st International Conference on Communities and Technologies*. Amsterdam, Netherlands, Sept. Huysman, M., Wenger, E., and Wulf, V. Eds. Kluwer. 81–96.

Wu, F. and Huberman, B. A. 2004. Finding communities in linear time: A physics approach. *European Phys. J. B 38*, 331–338.

Yang, B., Cheung, W. K., and Liu, J. 2007. Community mining from signed social networks. *IEEE Trans. Knowl. Data Engin. 19*, 1333–1348.

Yang, B. and Liu, J. 2007. An autonomy oriented computing (AOC) approach to distributed network community mining. In *Proceedings of the 1st International Conference on Self-Adaptive and Self-Organizing Systems*. Boston, MA. 151–160.

Zachary, W. W. 1977. An information flow model for conflict and fission in small groups. *J. Anthro. Res. 33*, 452–473.

Received January 2006; revised February 2007, September 2007, October 2007; accepted October 2007