Data & Knowledge Engineering xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Data & Knowledge Engineering

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

A link clustering based overlapping community detection algorithm

Chuan Shi *, Yanan Cai, Di Fu, Yuxiao Dong, Bin Wu

School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China

ARTICLE INFO

Available online xxxx

Keywords:
Community detection
Overlapping community
Genetic algorithm
Link community

ABSTRACT

There is a surge of community detection study on complex network analysis in recent years, since communities often play important roles in network systems. However, many real networks have more complex overlapping community structures. This paper proposes a novel algorithm to discover overlapping communities. Different from conventional algorithms based on node clustering, the proposed algorithm is based on link clustering. Since links usually represent unique relations among nodes, the link clustering will discover groups of links that have the same characteristics. Thus nodes naturally belong to multiple communities. The algorithm applies genetic operation to cluster on links. An effective encoding schema is designed and the number of communities can be automatically determined. Experiments on both artificial networks and real networks validate the effectiveness and efficiency of the proposed algorithm.

Crown Copyright © 2013 Published by Elsevier B.V. All rights reserved.

1. Introduction

Nowadays, community detection, as an effective way to reveal the relationship between structure and function of networks, has drawn lots of attention and been well developed. To do so, networks are modeled as graphs, where nodes represent objects and edges represent interactions among them. Community detection divides a network into groups of nodes, where nodes are densely connected inside but sparsely connected outside. However, in real world, objects often have diverse roles and belong to multiple communities. For example, a professor collaborates with researchers in different fields and a person has his family group as well as friend group at the same time. All of these objects represent the interaction between communities and play an important role in the stability of the network. In community detection, these objects should be divided into multiple groups, which are known as overlapping nodes [1]. The aim of overlapping community detection is to discover such overlapping nodes and communities.

Until now, lots of overlapping community detection approaches have been proposed, which can be roughly divided into two categories: node-based and link-based algorithms. The node-based overlapping community detection algorithms [1–7,9] directly divide nodes of the network into different communities. Based on an intuition that a link in networks usually represents the unique relation, the link-based algorithms firstly cluster on edges of network, and then map the link communities to node communities by gathering nodes incident to all edges within each link community [8]. The newly proposed link-based algorithms have shown its superiority on detecting complex multi-scale communities. However, they have the high computational complexities and bias on the discovered communities.

In this paper, we propose a genetic algorithm to detect overlapping communities with link clustering, which is named *G*enetic *a*lgorithm for *o*verlapping *C*ommunity *D*etection (GaoCD). The algorithm first finds the link communities by optimizing the objective function: partition density D [8], and then maps link communities to node communities based on a novel genotype representation method. The number of communities found by GaoCD can be automatically determined, without any prior information. Experiments on both artificial networks and real networks are designed to validate the algorithm. Experiments on artificial networks show that GaoCD works well on networks with typical overlapping structure. Experiments on real networks compare GaoCD with well-established algorithms. It shows that GaoCD always achieves higher partition density D and finds denser communities.

* Corresponding author.

E-mail addresses: shichuan@bupt.edu.cn (C. Shi), caiyanan@bupt.edu.cn (Y. Cai), fudi_lydia@qq.com (D. Fu), dongyuxiao@bupt.edu.cn (Y. Dong), wubin@bupt.edu.cn (B. Wu).

0169-023X/$ – see front matter. Crown Copyright © 2013 Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.datak.2013.05.004

ARTICLE IN PRESS

The paper is organized as follows. In the next section, we introduce the related works. Section 3 describes the proposed algorithm, including algorithm framework, objective function, genetic representation, and operators. The experimental results are illustrated in Section 4. Finally, Section 5 concludes the paper.

2. Related work

Community detection has been well studied in the last ten years, and a number of algorithms have been developed. These community detection algorithms can be roughly divided into two categories, optimization based methods (e.g. GN fast [19]) and heuristic methods (e.g. GN [29]). Both of the algorithms need a criterion to evaluate the community partition. A widely accepted criterion is the modularity Q proposed by Girvan–Newman [29], which quantitatively defines community structure as a node group that is densely intra-connected and sparsely inter-connected. The modularity Q has been widely used as the optimization objection in many algorithms, such as the algorithm using extremal optimization [10] and GN fast [19].

However, there are many overlapping networks in real world. For example, a person belongs to more than one social group at the same time. With the increase of the number of communities that overlapping nodes belong to, the internal connection among these communities becomes denser. As a result, the modularity Q does not fit for overlapping communities any more [8]. To discover the overlapping structure of networks, many algorithms have been proposed, which can be roughly divided into two categories: node-based algorithms and link-based algorithms.

The node-based algorithms cluster nodes of network directly, utilizing the structure information of nodes. Many well-established algorithms belong to this type. Some algorithms utilize the local expansion by optimizing a local benefit function, such as IS [13], LFM [7], MONC [15], CIS [20], OSLOM [21]. Some fuzzy community detection algorithms calculate the possibility of each node belonging to every community, such as the algorithm proposed by Gregory [4], spectral clustering framework based algorithm [22], SPAEM [23], SSDE [24], SBM [25], OSBM [26] and MOSES [27], etc. Some label propagation based algorithms allow multiple labels for each node to detect overlapping structure, such as COPRA [17], SLPA [18], multi-state spin model [28], etc. Moreover, the clique percolation can also be used to overlapping community detection, which includes the well-known CPM [1], SCP [33], EAGLE [16]. Recently, Shen et al. [14] detect the overlapping community through partitioning the maximal clique network constructed from the original network. Most of these algorithms need prior information to obtain the final result. For example, IS [13] needs to initialize seed nodes at the beginning, and LFM [7], as well as MONC [15], needs an appropriate parameter α to control the size of communities. For fuzzy community detection algorithms, the number of communities should be determined in advance [4], as CPM and SCP do. COPRA and SLPA can determine the number of community automatically, while COPRA is blamed for discovering communities with small size in some networks.

Genetic algorithm has also been adopted to detect overlapping community, such as GA-Net+ [9] proposed by Pizzuti. The algorithm first transfers the network to line graph and encodes based on nodes on the line graph. At each iteration, GA-Net+ converts the communities in the line graph into communities of the original network and uses the community score [9] to evaluate the community partition. Although GA-Net+ proposes the concept of line graph, it is still a node-based overlapping community detection algorithm in nature.

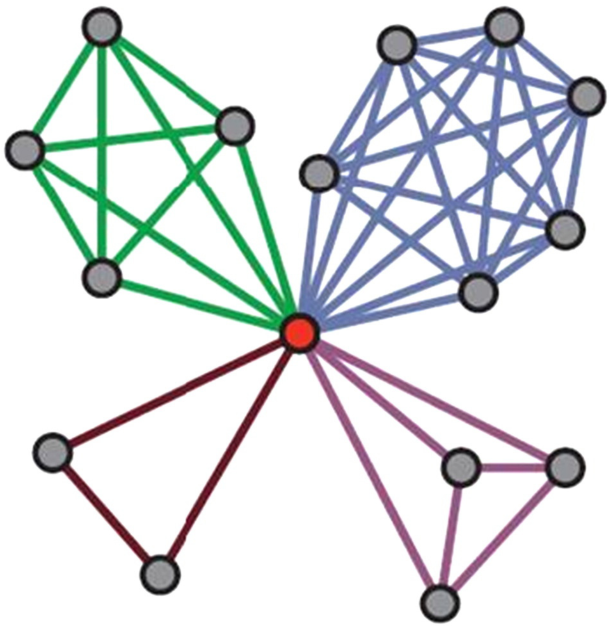


Fig. 1. An example of overlapping communities.

ARTICLE IN PRESS

The link-based method was newly proposed by Ahn, Bagrow and Lehmann (ABL) [8] in 2010 and applied by Ye et al. for massive networks [34]. ABL utilizes link structure information to detect overlapping communities. First, it discovers the link community partition, and then converts the link communities into nodes communities based on the incident relationship between edges and nodes. To evaluate the quality of link community, ABL proposes a novel criterion, partition density D. This criterion emphasizes the community density and ignores the connection among communities, which could result in bias on small communities in theory. As shown in Section 4, this algorithm tends to find small communities and cannot provide the view of the global community structure of the network.

In all, most of the node-based algorithms need prior information to detect overlapping communities and usually they could not provide the global structure of networks. The link-based method emphasizes on the unique role that each link represents, and provides a new way for overlapping community detection. However, the basis introduced by the partition density D still remain a challenge.

Algorithm 1. Main framework of GaoCD

```
1:  procedure GAOCD($size, gens, p_c, p_m$)
2:      //$size$ is the size of the population.
3:      //$gens$ is the running generation.
4:      //$p_c$ and $p_m$ are the ratio of crossover and mutation, respectively, with $p_c \in [0,1]$,
    $p_m \in [0,1]$ and $p_c + p_m = 1$.
5:      $P_1 = \Phi$
6:      for each $i$ in 1 to $size$ do
7:          $g_i = generate\_individual()$
8:          $evaluate(g_i)$; $P_1 = P_1 \cup \{g_i\}$
9:      end for
10:     for each $t$ in 1 to $gens$ do
11:         $n = 0$; $P_{t+1} = \Phi$
12:         while $n < size$ do
13:             randomly select two individuals $g_j$ and $g_k$ from $P_t$
14:             generate a random value $r \in [0,1]$
15:             if $r < p_c$ then
16:                 $(g_j^{'}, g_k^{'}) = crossover(g_j, g_k)$
17:             else $g_j^{'} = mutate(g_j)$; $g_k^{'} = mutate(g_k)$
18:             end if
19:             $n = n + 2$
20:             $evaluate(g_j^{'})$; $P_{t+1} = P_{t+1} \cup \{g_j^{'}\}$
21:             $evaluate(g_k^{'})$; $P_{t+1} = P_{t+1} \cup \{g_k^{'}\}$
22:         end while
23:         $selection(P_{t+1}, P_t \cup P_{t+1})$
24:     end for
25:     return $P[1]$
26: end procedure
        $generate\_individual()$ //initialize an individual according to the genetic representation schema.
        $evaluate(g)$ //evaluate the fitness of individual $g$ according to the partition density $D$.
        $crossover(g_j, g_k)$ //crossover operator.
        $mutate(g_j)$ //mutation operator.
        $selection(P_{t+1}, P_t \cup P_{t+1})$ //select the top $size$ individuals from $P_t \cup P_{t+1}$ as new $P_{t+1}$.
```

3. The link-based overlapping community detection approach

This section presents the proposed algorithm GaoCD in detail. The GaoCD employs a genetic algorithm to detect link communities and then converts link communities to node communities. It includes the main components of GaoCD: algorithm framework, objective function, genetic representation and corresponding operators.

3.1. Algorithm framework

Genetic algorithm (GA), derived from evolutionary biology, is a searching technique to find exact or approximate solutions for optimization problems. GA employs a population of individuals to represent candidate solutions. These solutions evolve towards approximate solutions based on the production and selection schema. The essential of GA is the survival of fittest for individuals. GA has shown its superiority in community detection [11,12]. However, there is no work to detect link community with GA until now. We first propose such an algorithm, called GaoCD. It is not a trivial work to design a GA to detect the link community, since the genetic representation and operators determine the efficiency of the algorithm greatly. Following the general GA framework, the pseudocode is shown in Algorithm 1. In the following sections, we will present the details of GaoCD's main components.

ARTICLE IN PRESS

3.2. Objective function

In GaoCD, the value of objective function represents the fitness of individuals, which determines the merit of individuals. The objective function guides the random search of GaoCD, which is used in the *evaluate()* function of Algorithm 1. In overlapping communities, one node is allowed to belong to more than one community, which makes the conventional community definitions unreasonable [8]. As shown in Fig. 1, the network contains four obvious communities, all of which are cliques. However, all communities are overlapping in the central node. Although these communities are densely intra-connected, they are not sparsely connected with other communities. As a result, new community definition should be adopted for overlapping community detection algorithms.

Although several evaluation criteria for overlap communities have been proposed, most of them are based on node communities. Recently, Ahn et al. [8] proposed the partition density D for overlapping community, which evaluates the link density within communities.

For a network with M links, suppose $P = \{P_1, \cdots, P_C\}$ as a partition of the network's links into C subsets. $m_c = |P_c|$ is the number of links in subset c. $n_c = \left|\cup_{e_{ij} \in P_c} \{i,j\}\right|$ represents the number of nodes incident to links in subset c. D_c refers to the link density of subset c, which is define as follows.

$$D_c = \frac{m_c - (n_c - 1)}{\frac{n_c(n_c-1)}{2} - (n_c - 1)} \tag{1}$$

The partition density D is the average of D_c over all communities, weighted by the fraction of links presenting in each community. It is defined as follows.

$$D = \sum_c \frac{m_c}{M} D_c = \frac{2}{M} \sum_c m_c \frac{m_c - (n_c - 1)}{(n_c - 2)(n_c - 1)} \tag{2}$$

Different from the conventional community evaluation criteria that a community should be densely intra-connected and sparsely connected with the rest communities, partition density D evaluates the link density within the community, which is suitable for overlapping community detection. In this paper, GaoCD will employ the partition density D as the objective function.

3.3. Genetic representation

When GA is applied to community detection, a community partition (i.e. individual) should usually be encoded in a character string (i.e., genotype) with a genetic representation, and inversely a genotype can also be decoded into a community partition. In this section, we describe the genetic representation of link communities in detail, including the encoding and decoding phase.

Some genetic representations have been proposed (e.g., locus-based adjacency representation [9,12]) for those node-based algorithms, in which a gene represents a node in network. Different from those node-based genotypes, GaoCD encodes on links of the network. In this link-based representation, an individual g in the population consists of m genes $\{g_0, g_1, \cdots g_i, \cdots g_{m-1}\}$, where $i \in \{0, \cdots, m-1\}$ is the identifier of edges (i.e., links), m is the number of edges, and each g_i randomly takes one of the adjacent edges of edge i. According to graph theory, two edges are adjacent if they share one node in undirected graph. For example, in Fig. 2(a), edge 0 has four adjacent edges {1,5,2,6} and a possible value of g_0 is 1. The encoding schema guarantees that any community partition can be encoded into a corresponding genotype and any genotype can be decoded to a valid community partition. Besides, the encoding schema makes GaoCD determine the number of community automatically, without any prior information.

The decoding phase transfers a genotype to a link community partition. The value of gene g_i is j, which means that edge i and edge j have one common node. So they should be classified to the same component. In the decoding phase, it needs to identify all connected

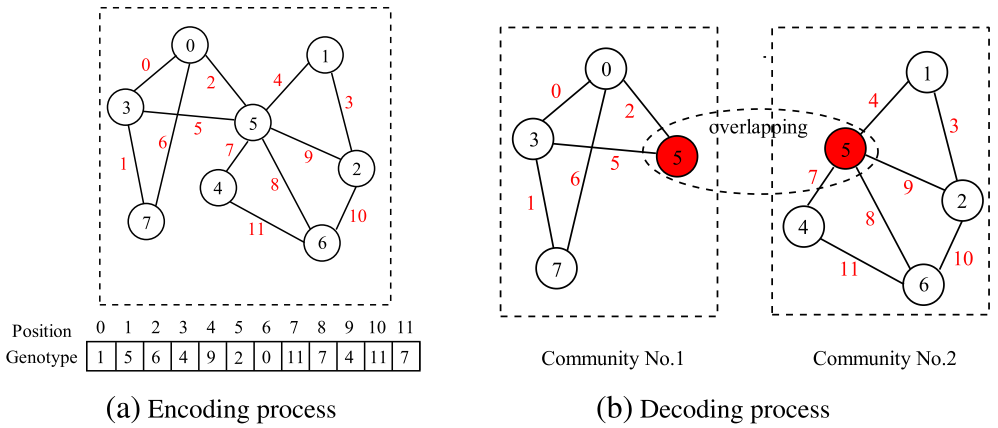


Fig. 2. Illustration of the genetic representation. (a) A simple network and its possible genotype. (b) The corresponding decoded partition.

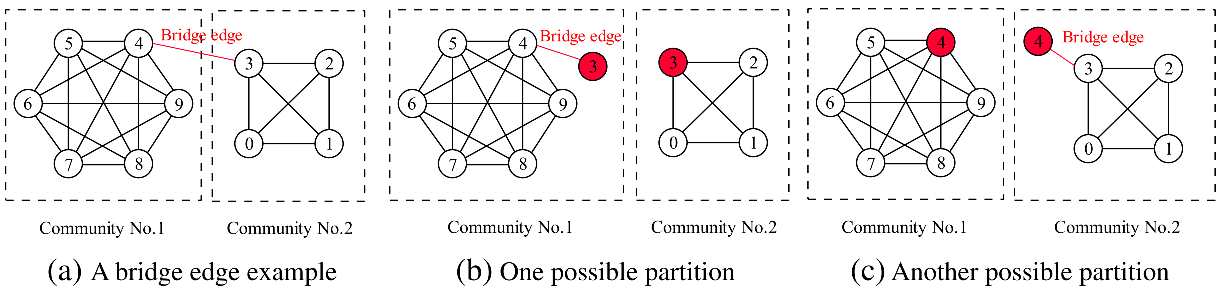

Bridge edge
Community No.1
Community No.2
(a) A bridge edge example
(b) One possible partition
(c) Another possible partition

Fig. 3. Illustration of bridge edge. (a) A sample network containing bridge edge <3, 4>. (b) An incorrect partition which wrongly divides node 3 to both communities. (c) Another incorrect partition which wrongly divides node 4 to both communities.

components. All edges belonging to the same connected component are then assigned to one community. An example is shown in Fig. 2(b), using a simple backtracking scheme, this decoding step can be performed in a linear time. After obtaining link communities, overlapping communities can be obtained by gathering the nodes incident to the edges in each link community. However, this strategy cannot be applied directly, because it will generate unsuitable partitions in the bridge edge situation.

The bridge edge is defined as the edge connecting two communities. An example is the edge <3,4> shown in Fig. 3(a). It is obvious that Fig. 3 contains two communities, both of which are cliques. Because the bridge edge must belong to one unique link community, it will be classified to either of the two cliques, as shown in Fig. 3(b) and (c). By simply gathering nodes incident to edges of link community to form node communities, we will obtain overlapping communities as shown in Fig. 3(b) and (c), respectively. Obviously, it is not the real partition. To avoid this flaw, we propose the fine tuning method.

Fine tuning adjusts the node's membership by a mapping schema. It is designed for nodes with multiple memberships. The method first finds the list of nodes with multiple memberships, and then for each node i in the list, it tests that whether node i contributes to the adjacent communities $\{c_{i1}, c_{i2}, \cdots, c_{in}\}$ by adding to them, where c_{ij} is the community containing node i. Here, we adopt the average degree of community to evaluate the contribution which is defined as follows:

$$AD(c) = 2 * \frac{|E(c)|}{|c|} \tag{3}$$

where c is a community, $E(c)$ is the number of edges in the community, and $|c|$ is the number of nodes of the community. If adding a node to the community c makes $|AD(c)|$ increase, we think that the node contributes to the community. If node i contributes to community c_{ij}, then the community keeps this node, otherwise node i is removed. If the average degree decreases for all adjacent communities, the community with the least decreasing value will keep this node and other communities will remove it. For example, in Fig. 3(b), node 3 is a potential overlapping node and it may be contained in community No.1 and community No.2. Because node 3 decreases the $|AD(c)|$ of community No.1 and increases the $|AD(c)|$ of community No.2, node 3 should be removed from community No.1. In Fig. 3(c), node 4 decreases the $|AD(c)|$ of community No.2 and increases that of community No.1, so node 4 would be removed from community No.2.

3.4. Genetic operators

According to the genetic representation, we further propose the corresponding genetic operators. These operators guarantee that new generated individuals obey the encoding rule. That is, the gene value of position i is a adjacent edge of edge i. It means that these new individuals always are valid ones.

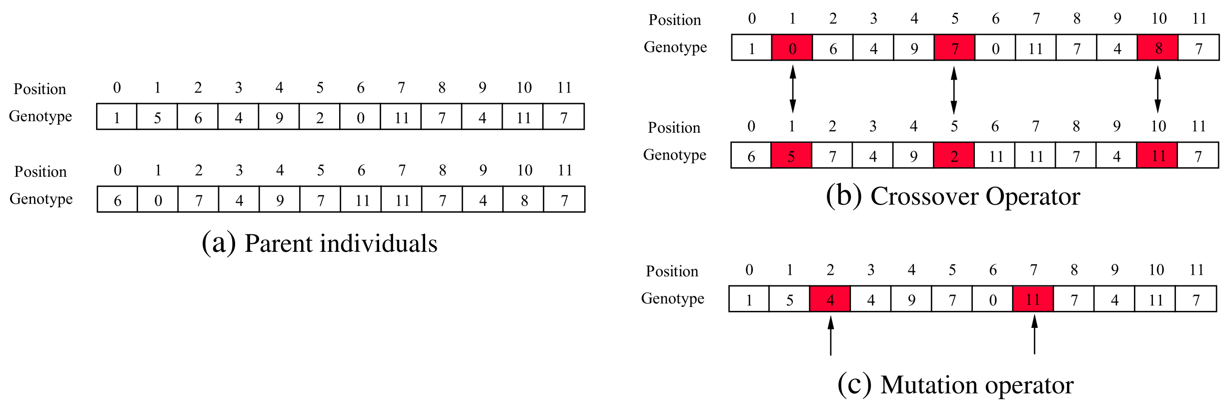

Position 0 1 2 3 4 5 6 7 8 9 10 11
Genotype 1 5 6 4 9 2 0 11 7 4 11 7
Position 0 1 2 3 4 5 6 7 8 9 10 11
Genotype 6 0 7 4 9 7 11 11 7 4 8 7
(a) Parent individuals
Position 0 1 2 3 4 5 6 7 8 9 10 11
Genotype 1 0 6 4 9 7 0 11 7 4 8 7
Position 0 1 2 3 4 5 6 7 8 9 10 11
Genotype 6 5 7 4 9 2 11 11 7 4 11 7
(b) Crossover Operator
Position 0 1 2 3 4 5 6 7 8 9 10 11
Genotype 1 5 4 4 9 7 0 11 7 4 11 7
(c) Mutation operator

Fig. 4. Illustration of genetic operators. (a) Shows two individuals of the population. (b) Illustrates the crossover operator. (c) Illustrates the mutation operator.

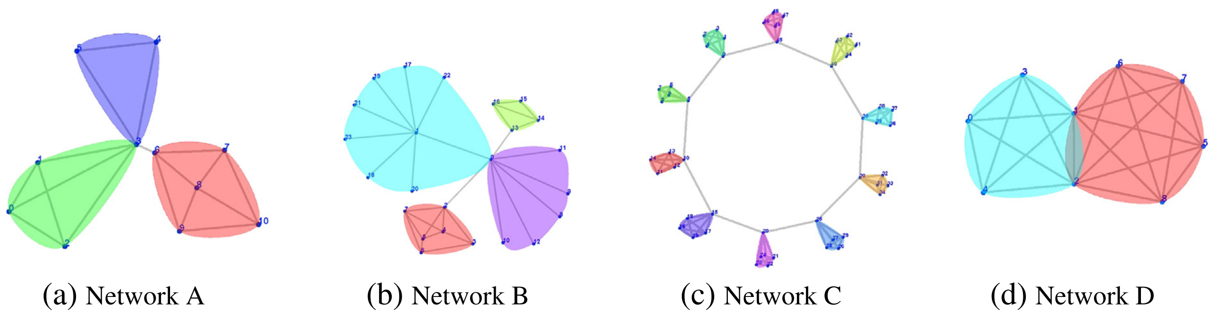


Fig. 5. Four typical overlapping networks.

In the crossover operation, we randomly select two individuals from the current population. The exchanging positions are randomly generated and then we exchange the genes in these positions between these two individuals. Fig. 4(b) shows such an example. Since the gene in the position *i* (i.e., g_i) always is the identity of the adjacent edges of edge *i*, the exchanged individuals also follow the encoding rule: g_i is an adjacent edge of edge *i*. The crossover process is the *crossover()* function in Algorithm 1.

In the mutation operation, an individual is randomly selected from the current population and the positions are randomly generated. Then we reassign the gene values on these positions with a random adjacent edge. An example is shown in Fig. 4(c). The mutation operation is the *mutate()* function in Algorithm 1.

In the population selection process (i.e., the *selection()* function in Algorithm 1), we sort the individuals according to their fitness (i.e., the partition density *D*), and then select the top *size* individuals as the new population.

In the population initialization process (i.e., the *generate_individual()* function in Algorithm 1), we randomly generate a set of individuals. For each individual, each gene is randomly assigned one of its adjacent links.

4. Experiments

This section validates the effectiveness of GaoCD with extensive experiments. We first assess the ability of GaoCD to discover the overlapping nodes on typical networks, and then evaluate effectiveness of GaoCD on the artificial and real networks compared with well-established algorithms. All of the experiments are carried out on a 2.66-GHz and 2-G RAM Pentium IV computer.

4.1. Experiments on typical overlapping structure

In this section, we evaluate GaoCD on four toy networks with typical overlapping structures [34], as shown in Fig. 5. Because these networks are small, we do not set *gens* to control the running generation but let GaoCD run until convergence. Other parameters are set as follows: $size = 50$, $p_c = 0.6$, $p_m = 0.4$. We can see that network *A* and network *C* contain bridge edges which should not be divided into either of the communities. GaoCD successfully distinguishes the bridge edges and deal with them properly. Network *B* is a hierarchical core network, where GaoCD successfully finds the overlapping node. In network *D*, GaoCD correctly divides the sharing nodes of the two cliques. Overall, GaoCD accurately reveals overlapping communities for all these networks and ensures that all nodes of the network are covered in the partition. Although these networks are toy examples, real networks compose of these basic structures. GaoCD does a good job on these networks, which guarantees that it has the potential to perform well on real networks.

4.2. Experiments on artificial networks

To test the performance of GaoCD quantitatively, we adopt LFR [30] as benchmark and use Fraction of Vertices Identified Correctly (*FVIC*) as measure criterion. LFR benchmark [30] simulates real networks in degree and community size distribution, and has been widely used to evaluate algorithms for overlapping community detection [35,36]. The distributions of degree and

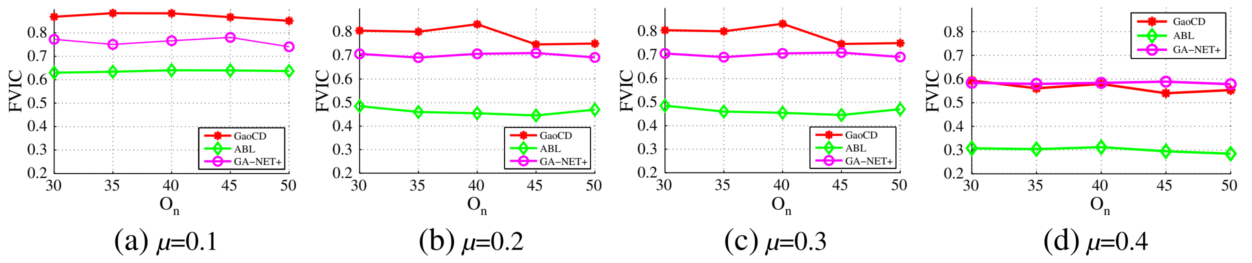


Fig. 6. The *FVIC* values of three algorithms on artificial networks. *FVIC* is the fraction of vertices identified correctly and the large *FVIC* means the better performance. O_n is the number of overlapping nodes. μ controls the ratio of links connecting with outside communities and the large μ means the more fuzzy community structure.

ARTICLE IN PRESS

Table 1
Real networks used in experiments.

	Karate (N1)	Polbooks (N2)	Dolphins (N3)	Football (N4)	Lesmis (N5)	Adjnoun (N6)	Enron (N7)	Email (N8)	Power (N9)
# of nodes	34	105	62	115	77	112	150	1133	4941
# of edges	78	441	159	613	254	425	1526	5451	6594

community size are governed by power laws with two parameters τ_1, τ_2. Parameter O_n controls the overlapping nodes in the whole network, and O_m is the number of communities that each overlapping node belongs to. LFR also provides the mixing parameter μ, the average degree $\overline{k}$, the maximum degree k_{max}, the maximum community size c_{max}, and the minimum community size c_{min} to control the network topology. The *FVIC* represents the faction of vertices identified correctly, and has been used in several researches [31,32]. The larger the *FVIC*, the better the partition is.

In our experiments, we use networks with the following parameters: $n = 1000, \overline{k} = 10, \tau_1 = 2, \tau_2 = 1, k_{max} = 50, c_{min} = 4, c_{max} = 30$. The parameters of GaoCD is set as follows: $size = 100, gens = 100, p_c = 0.6, p_m = 0.4$. Two representative algorithms are also included: ABL [18] and GA-Net+ [9]. ABL is a link-based overlapping community algorithm. As a node-based algorithm, GA-Net+ uses the same GA framework with GaoCD. The results are shown in Fig. 6. It is obvious that GaoCD always achieves the best performance on most experiments compared with ABL and GA-Net+. When μ varies from 0.1 to 0.4, the community structure becomes fuzzy, so the performance of all the algorithms decreases. When μ is 0.4, the community structure becomes very fuzzy. In this condition, GA-Net+ is slightly better than GaoCD. The experiments also show that the number of overlapping nodes O_n has little effect on the performance of all these algorithms.

4.3. Experiments on real networks

Experiments on real networks include three parts. (1) The effectiveness of GaoCD is evaluated on nine real networks. (2) We investigate structural characteristics of communities discovered by GaoCD. (3) We further give the intuitive view of communities discovered by GaoCD on two real networks.

In the first experiment, we used nine widely-used real networks [9,33], as shown in Table 1. ABL and GA-Net+ are also employed to compare their performance with GaoCD. We use the partition density D [8] as the evaluation criterion, which evaluates the link density inside the communities. For all the real networks, we set $p_c = 0.6$ and $p_m = 0.4$. When the number of edges is less than 500, we set $size = 100$, $gens = 100$, while for those large networks (i.e., the number of edges is larger than 500), we set $size = 200$, $gens = 200$. The results are shown in Table 2. It is obvious that GaoCD achieves the highest partition density D for most real networks. It indicates that the communities found by GaoCD is denser than that of ABL and GA-Net+.

To explore the structure properties of communities found by GaoCD, we analyze the distribution of community size for all real networks. We omit those cliques with size two, since they are just a link and do not contribute to partition density D. We classify the communities into three types according to the community size (donated as *cs*): small communities ($cs \in [3,5]$), middle communities ($cs \in [6,10]$), and large communities ($cs > 10$). Here, we compare GaoCD with ABL, since they both are link-based algorithms and adopt the partition density D as evaluation criterion. Fig. 7 shows the ratio of three types of communities. We can find that the ratio of small communities found by ABL are large, while it is not the case for GaoCD. The results show that ABL tends to find tiny communities. However, these tiny communities usually are meaningless and it cannot reflect the whole structure of the network. In contrast, GaoCD finds denser communities in all sizes, which implies that it can discover the macro-structure as well as the micro-structure of the network.

We further visualize the community partition of two real networks (i.e., polbooks and dolphins) in Fig. 8. Polbooks is a co-purchasing network of books about US politics by the online bookseller Amazon.com. Nodes represent books, and the edges between nodes represent frequent co-purchasing of books by the same buyers. Fig. 8(a) obviously shows that the network constitutes two large communities, with each of them surrounded by small ones. Nodes 3, 6, and 58 have obvious overlapping membership. The dolphins is a social network of frequent association between 62 dolphins in a community living in Doubtful Sound, New Zealand. As shown in Fig. 8(b), the network may fall into two parts because of the leave of SN100. GaoCD successfully distinguishes the special role of SN100. SN100 is the core of the community, which connects two communities.

Table 2
The partition density D on real networks.

	N1	N2	N3	N4	N5	N6	N7	N8	N9
GaoCD	**0.5167**	**0.3115**	**0.4183**	**0.5815**	**0.6317**	0.1421	**0.2631**	0.1572	**0.1792**
ABL	0.2848	0.2867	0.3203	0.5500	0.5821	0.1310	0.2464	**0.2200**	0.1540
GA-Net+	0.4624	0.1926	0.3308	0.1507	0.5881	**0.2361**	0.1265	0.1066	0.1248

ARTICLE IN PRESS

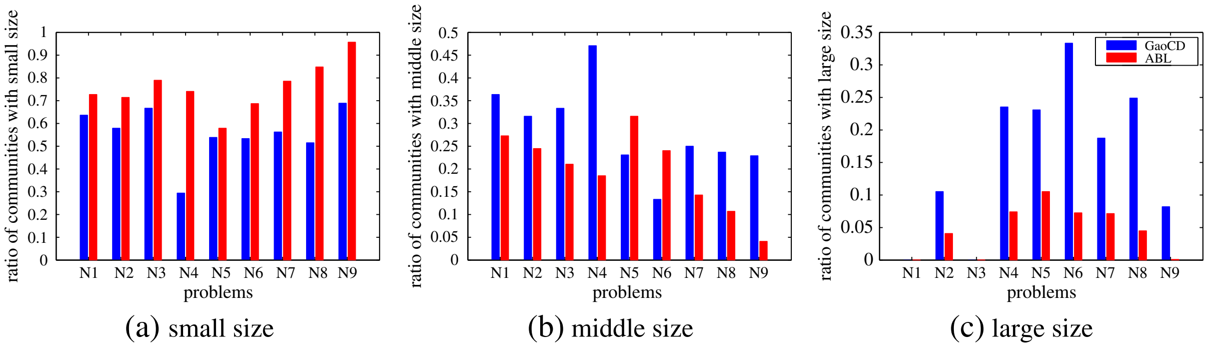


Fig. 7. Comparison of GaoCD and ABL on the distribution of community size on real networks.

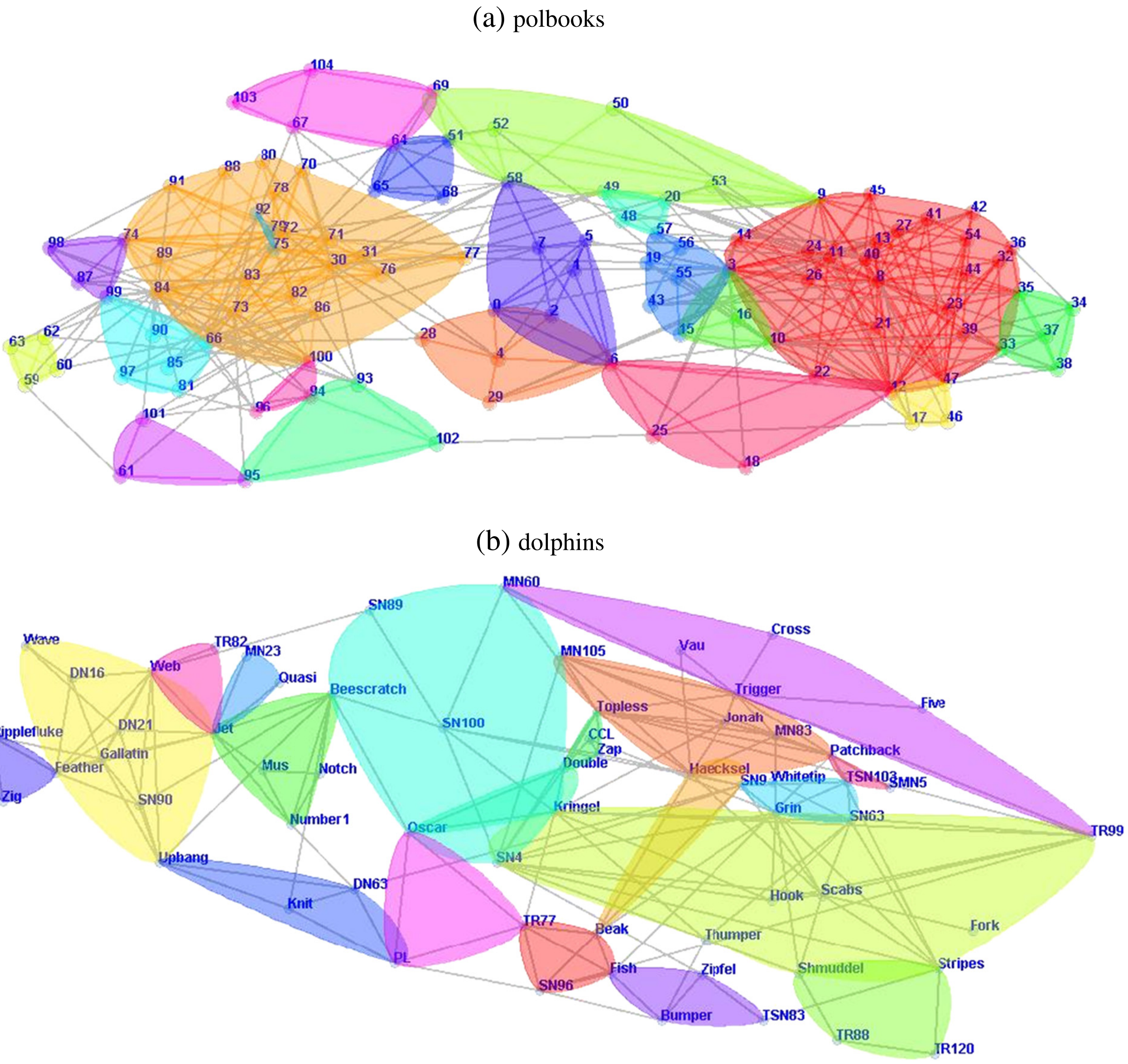


Fig. 8. Partitions found by GaoCD. (a) Polbooks network. (b) Dolphins network.

4.4. Discussion

Through experiments on both artificial and real networks, we can find that GaoCD always has the best performance on most networks. Moreover, although ABL and GaoCD share the common optimization objective (i.e., the partition density D), GaoCD not only achieves better objective values but also reveals large communities which is more meaningful in real networks. We think the reason lies in the fine tuning operation in GaoCD. The maximization of the partition density D leads to discover the cliques. However, real networks are usually very sparse. The cliques in these networks are very small. So ABL tends to find tiny communities. Although GaoCD has the same tendency, the fine tuning operation in GaoCD adjusts the assignment of overlapping nodes on edge bridges. The operation improves the link density of communities, and thus enhances the fitness of individuals (i.e., D). As a result, those individuals without the fine tuning operation will be eliminated during the population evolution due to the low fitness. These eliminated individuals have the common characteristics: discovering tiny communities. So the individuals discovering large communities will increase during population evolution. That is why GaoCD can find large communities.

Let's consider the time and space complexity of GaoCD. Suppose n is the number of nodes, m the number of edges, k the number of communities, d the average number of degrees of a node. For each iteration and each individual, the main time-consuming components of GaoCD include encoding, decoding, fine tuning and calculating fitness. Since the GaoCD encodes on edges, the encoding and decoding processes both have the $O(m)$ time complexity. The complexity of calculating fitness is $O(k)$. The complexity of the fine tuning operation is $O(m+dn)$. It means that the fine tuning operation leads to good performance with the sacrifice in time. As a consequence, the time complexity of single individual is $O(3\,m+dn+k)$. After omitting the const and tiny variable, the time complexity is $O(m+n)$. The whole time complexity of GaoCD is $O(gs(m+n))$ (g is the running generation and s is the population size). The GA-Net+ has the same GA framework with GaoCD and the same time complexity $O(gs(m+n))$. ABL is based on hierarchical clustering which has the $O(m^3)$ time complexity. The results show that ABL has a higher time complexity than GaoCD and GA-Net+. GaoCD employs a population with s individuals and each individual encodes on edges, so the space complexity is $O(sm)$. GA-Net+ encodes on the line graph which has the m nodes, so the space complexity is also $O(sm)$. ABL only needs to store m edges, so its space complexity is $O(m)$, which is smaller than that of GaoCD and GA-Net+.

All parameters of GaoCD come from genetic algorithm. It obeys the general rules of the parameter setting in genetic algorithm. Generally speaking, more population size and running generation (i.e., the parameter *size* and *gens* in GaoCD) usually lead to better performance with more running time. In real applications, we need to make a tradeoff on the setting of these parameters. For large-scale networks, we can set the not large values for these parameters to achieve acceptable performances with not long running time. In our experiments, we set the proper parameters according to the scale of problems. The crossover operation helps to converge quickly, but may cause prematurity. The mutation operation has the opposite tendency. In our experiments, the ratio of crossover and mutation operation is fixed with $p_c=0.6$ and $p_m=0.4$, since we found that they are suitable for all networks.

5. Conclusion

In this paper, we propose a genetic algorithm for overlapping community detection based on the link clustering framework. Different from those node-based overlapping community detection algorithms, GaoCD utilizes the property of the unique role of links and applies a novel genetic algorithm to cluster on edges. The genetic representation and the corresponding operators effectively represent the link communities and make the number of the communities determined automatically. Experiments on artificial and real networks show that GaoCD can effectively reveal overlapping structure.

Acknowledgments

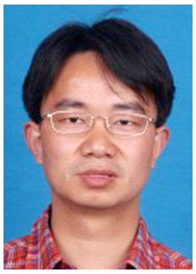
It is supported by the National Natural Science Foundation of China (nos. 60905025, 61074128, 61035003, 71231002). This work is also supported by the National Key Basic Research Program (973 Program) of China (2013CB329603) and the Fundamental Research Funds for the Central Universities.

References

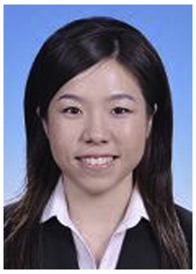
[1] G. Palla, I. Derenyi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814–818.
[2] J.B. Pereira, A.J. Enright, C.A. Ouzounis, Detection of functional modules from protein interaction networks, Proteins: Structure, Functions, and Bioinformatics 54 (2004) 49–57.
[3] J. Baumes, M.K. Goldberg, M.S. Krishnamoorthy, M.M. Ismail, N. Preston, Efficient Identification of Overlapping Communities, ISI, 2005, 27–36.
[4] S.H. Zhang, R.S. Wang, X.S. Zhang, Identification of overlapping community structure in complex networks using fuzzy c-means clustering, Physica A 374 (2007) 483–490.
[5] S. Gregory, An Algorithm to Find Overlapping Communities Structure in Networks, PKDD, 2007, 91–102.
[6] S. Gregory, A Fast Algorithm to Find Overlapping Communities in Networks, PKDD, 2008, 408–423.
[7] A. Lancichinetti, S. Fortunato, J. Kertesz, Detecting the overlapping and hierarchical community structure in complex networks, New Journal of Physics 11 (2009) 033015.
[8] Y.Y. Ahn, J.P. Bagrow, S. Lehmann, Link communities reveal multi-scale complexity in networks, Nature 466 (2010) 761–764.
[9] C. Pizzuti, Overlapping Community Detection in Complex Networks, GECCO, 2009, 859–866.
[10] J. Duch, A. Arenas, Community detection in complex networks using extremal optimization, Physical Review E 72 (2005) 027104.
[11] M. Tasgin, H. Bingol, Community Detection in Complex Networks using Genetic Algorithm, 2006, (arXiv:cond-mat/0604419).
[12] C. Shi, Z.Y. Yan, Y. Wang, Y.N. Cai, B. Wu, A genetic algorithm for detecting communities in large-scale complex networks, Advance in Complex System 13 (1) (2010) 3–17.

ARTICLE IN PRESS

[13] J. Baumes, M. Goldberg, M. Krishnamoorthy, M. Magdon-Ismail, N. Preston, Finding Communities by Clustering a Graph into Overlapping Subgraphs, IADIS, 2005, 97–104.
[14] H.W. Shen, X.Q. Cheng, J.F. Guo, Quantifying and identifying the overlapping community structure in networks, Journal of Statistical Mechanics (2009) P07042.
[15] F. Havemann, M. Heinz, A. struck, J. Glaser, Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels, Journal of Statistical Mechanics: Theory and Experiment 01 (2011) 01023.
[16] H.W. Shen, X.Q. Cheng, K. Cai, M.B. Hu, Detect overlapping and hierarchical community structure in networks, Physica A 388 (8) (2009) 1706–1712.
[17] S. Gregory, Finding overlapping communities in networks by label propagation, New Journal of Physics 12 (2010) 10301.
[18] J. Xie, K. Szymanski, X. Liu, SLPA: Uncovering Overlapping Communities in Social Networks via a Speaker–Listener Interaction Dynamic Process, ICDMW, 2011, 344–349.
[19] R. Guimera, L.A.N. Amaral, Functional cartography of complex metabolic networks, Nature 433 (2005) 895–900.
[20] S. Kelley, The existence and discovery of overlapping communities in large-scale networks, Ph.D. thesis Rensselaer Polytechnic Institute, Troy, NY, 2009.
[21] A. Lancichinetti, F. Radicchi, J.J. Ramasco, S. Fortunato, Finding statistically significant communities in networks, PLoS One 6 (4) (2011) 18961.
[22] S. Zhang, R.S. Wang, X.S. Zhang, Uncovering fuzzy community structure in complex networks, Physical Review E 76 (2009) 046103.
[23] W. Ren, G. Yan, X. Liao, L. Xiao, Simple probabilistic algorithm for detecting community structure, Physical Review E 79 (2009) 036111.
[24] M. Magdon-Ismail, J. Purnell, Fast overlapping clustering of networks using sampled spectral distance embedding and GMMs, Tech. rep., Rensselaer Polytechnic Institute, 2011.
[25] K. Nowicki, T.A.B. Snijders, Estimation and prediction for stochastic blockstructures, Journal of the American Statistical Association 96 (455) (2001) 1077–1087.
[26] P. Latouche, E. Birmele, C. Ambroise, Overlapping stochastic block models with application to the French political blogosphere, The Annals of Applied Statistics 5 (2011) 309–336.
[27] A. McDaid, N. Hurley, Detecting highly overlapping communities with model-based overlapping seed expansion, Advances in Social Networks Analysis and Mining (2010) 112–119.
[28] Q. Lu, G. Korniss, B.K. Szymanski, The naming game in social networks: community formation and consensus engineering, Journal of Economic Interaction and Coordination 4 (2009) 221–235.
[29] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proceedings of the National Academy of Sciences of the United States of America 99 (2002) 7821–7826.
[30] A. Lancichinetti, S. Fortunato, Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities, Physical Review E 80 (2009) 016118.
[31] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, Journal of Statistical Mechanics: Theory and Experiment (2005) P09008.
[32] S. Fortunato, M. Barthelemy, Resolution limit in community detection, PNAS 104 (1) (2007) 36–41.
[33] J.M. Kumpulä, M. Kivel, K. Kaski, J. Saramäki, Sequential algorithm for fast clique percolation, Physical Review E 78 (2008) 026109.
[34] Q. Ye, B. Wu, Z.X. Zhao, B. Wang, Detecting Link Communities in Massive Networks, ASONAM, 2011, 71–78.
[35] J. Xie, S. Kelly, B.K. Szymanski, Overlapping community detection in networks: the state of the art and comparative study, ACM Computing Surveys, 2012.
[36] I. Psorakis, S. Roberts, M. Ebden, Overlapping community detection using Bayesian nonnegative matrix factorization, Physical Review E 83 (2011) 066114.



Chuan Shi received the B.S. degree from the Jilin University, Changchun, Jilin, 2001, the M.S. degree from the Wuhan University, Hubei, in 2004, and Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 2007. He joined the School of Computer of Beijing University of Posts and Telecommunications as a lecturer in 2007, and is an associate professor at present. His research interests are in machine learning, data mining, and evolutionary computation. He has published more than 30 papers in refereed journals and conferences.



Yanan Cai received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, Sichuan, 2009. Then she studied in Beijing University of Posts and Telecommunications as a postgraduate student from 2009, and graduated in 2012. Her research interests are machine learning, data mining, and evolutionary computation.

Di Fu is an undergraduate student in the Computer Science Department in Beijing University of Post and Telecommunication. She will study in Tsinghua University as a master student. Her research interests are data mining and cloud computing.

ARTICLE IN PRESS

Yuxiao Dong was a master student at Beijing University of Posts and Telecommunications when this work was done, and is a Ph.D. student at the University of Notre Dame now. His research interests focus on data mining and social network analysis with an emphasis on network evolution.

Bin Wu received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 2002. He is a senior member of CCF. He joined the School of Computer of Beijing University of Posts and Telecommunications as a lecturer in 2002, and is a professor at present. His research interests are in data mining, complex network, and cloud computing. He has published more than 100 papers in refereed journals and conferences.