American Journal of Operations Research, 2015, 5, 421-434
Published Online September 2015 in SciRes. http://www.scirp.org/journal/ajor
http://dx.doi.org/10.4236/ajor.2015.55035

Mathematical Model and Algorithm for Link Community Detection in Bipartite Networks

Zhenping Li[1], **Shihua Zhang**[2], **Xiangsun Zhang**[2]

[1]School of Information, Beijing Wuzi University, Beijing, China
[2]National Center for Mathematics and Interdisciplinary Sciences, Academy of Mathematics and Systems Science, Beijing, China
Email: lizhenping66@163.com, zsh@amss.ac.cn, zxs@amt.ac.cn

Received 18 August 2015; accepted 19 September 2015; published 22 September 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
http://creativecommons.org/licenses/by/4.0/
Open Access

Abstract

In the past ten years, community detection in complex networks has attracted more and more attention of researchers. Communities often correspond to functional subunits in the complex systems. In complex network, a node community can be defined as a subgraph induced by a set of nodes, while a link community is a subgraph induced by a set of links. Although most researches pay more attention to identifying node communities in both unipartite and bipartite networks, some researchers have investigated the link community detection problem in unipartite networks. But current research pays little attention to the link community detection problem in bipartite networks. In this paper, we investigate the link community detection problem in bipartite networks, and formulate it into an integer programming model. We proposed a genetic algorithm for partition the bipartite network into overlapping link communities. Simulations are done on both artificial networks and real-world networks. The results show that the bipartite network can be efficiently partitioned into overlapping link communities by the genetic algorithm.

Keywords

Bipartite Network, Link Community, Quantity Function, Integer Programming, Genetic Algorithm

1. Introduction

Many interesting systems can be represented as networks [1]-[4]. The networks are composed of nodes and links, each node represents a unit and each link represents a relation between two nodes. Since some nodes or links may have the same function in complex system. One of the most important topics in the area of networks is the community detection, which is a universal problem in many disciplines such as sociology, computer science and biology [5]-[7].

How to cite this paper: Li, Z.P., Zhang, S.H. and Zhang, X.S. (2015) Mathematical Model and Algorithm for Link Community Detection in Bipartite Networks. *American Journal of Operations Research*, **5**, 421-434.
http://dx.doi.org/10.4236/ajor.2015.55035

The communities are dense subgraphs induced by a set of nodes or links. If the community is induced by a set of nodes, we call it node community. If a community is induced by a set of links, we call it link community. When we partition a network into node communities, each node must belong to one or more community, some links might belong to no community. When a network is partition into link communities, each link must belong to one community, and each node might belong to one or more communities. By partition the network into link communities, we can find overlapping node or link.

Although most research paid more attention to node community detection, some researchers have investigated link communities and cliques [8]-[12]. In some real-world networks, a link is more likely to have a unique identity while a node often has multiple functions, so the link communities might be more intuitive and informative than the node communities [13] [14].

Given a unipartite network with M links and N nodes, let $P=\{P_1,\cdots,P_K\}$ be a partition of the links into K subsets. $m_c=|P_c|$ be the number of links in subset P_c, $n_c=\bigcup_{e_{ij}\in P_c}\{i,j\}$ be the number of nodes in subgraph

induced by P_c. Ahn [8] defined the partition density D as follows

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

In [12], the authors proposed another partition density H as follows:

$$H=\frac{2}{M}\sum_c\frac{m_c^2}{n_c(n_c-1)}$$

Obviously, $0\le H\le 1$.

Given the number of communities, we can partition the unipartite network into link communities by maximize D or H.

Besides unipartite networks, there is another special category of network, where nodes are partitioned into two disjoint subsets, there is no link within the same subset. This type of network is called bipartite network. Some real-world relations are more suitable to be represented as bipartite networks [15], such as plant-animal network, scientific publication network, artistic collaboration network, order-item network, paper-author networks, event-attendee networks and so on.

Some research has paid attention to the node community detection problem of bipartite networks [15] [16]. In [15], the authors proposed a projection-based algorithm for node communities detection in bipartite network. In [17], the authors develop a modified adaptive genetic algorithm (MAGA) to detect the node communities in bipartite network. In [18], the authors propose another bipartite modularity detection method which can detect node overlap community. In [19], the authors proposed a hierarchical divisive heuristic for approximate modularity maximization in bipartite graphs. In [20], the authors proposed an algorithm *Bitector* to mine overlapping communities in large scale sparse bipartite networks. In [21], the authors proposed an approach for detecting overlap node communities in a bipartite network based on dual optimization of modularity. In [22] [23], the authors proposed weighted binary matrix factorization framework to detect overlapping communities in bipartite networks. Although the algorithms above can find node communities in bipartite network, current research activity has paid no attention to the link community detection problem in bipartite networks.

In this paper, we will investigate link communities in bipartite network, define the partition density of link communities in bipartite network, and formulate the link community partition problem of bipartite network into an integer programming model. Then we design a genetic algorithm for detecting link communities in bipartite network and conduct validations on some artificial and real-world bipartite networks. By the model and algorithm, the communities including two sets of nodes in bipartite network can be identified simultaneously.

2. Methods

2.1. Link Community Partition Density of Bipartite Networks

Given a bipartite network $G=(U,V,L)$ with $M=|L|$ links and two node sets U and V, where $U\cap V=\varnothing$, $P=\{P_1,\cdots,P_K\}$ is a partition of the links into K subsets. The number of links in subset P_c is $m_c=|P_c|$. The induced node set from link subset P_c is $\bigcup_{l_{ij}\in P_s}\{i,j\}$ (where l_{ij} represents the link connecting node u_i and

v_j), the number of induced nodes in node set *U* is $s_c = \left|\left(\bigcup_{l_{ij}\in P_s}\{i,j\}\right)\cap U\right|$, the number of induced nodes in node set *V* is $t_c = \left|\left(\bigcup_{l_{ij}\in P_s}\{i,j\}\right)\cap V\right|$. The link density H_c of community *c* in bipartite network is defined as follows:

$$H_c = m_c/(s_c t_c).$$

The partition density *H* is the average of H_c:

$$H = \frac{1}{M}\sum_c m_c \cdot H_c = \frac{1}{M}\sum_c \frac{m_c^2}{s_c t_c}.$$

We can see that the maximum of *H* is 1 and the minimum value of *H* is 0. $H=1$ when each community is a complete bipartite network and $H=0$ when each community is an empty bipartite graph. Given the number of communities, we can find the optimal link community partition of bipartite network by maximizing the value of *H*.

2.2. Integer Programming Model for Link Community Detection of Bipartite Network

Given a bipartite network $G=(U,V,L)$ with *M* links and $p+q=|U\cup V|$ nodes (where $p=|U|, q=|V|$), we assume that the number of link communities is *K* and find the optimal link community partition by maximizing the partition density *H*. This problem can be formulated into an integer programming model.

Let $U=\{u_1,u_2,\cdots,u_p\}$, $V=\{v_1,v_2,\cdots,v_q\}$ be two disjoint nodes sets of bipartite network *G*. $A=(a_{i\times j})_{p\times q}$ is the adjacent matrix of the bipartite network, where $a_{ij}=1$ when node u_i and v_j is connected by link l_{ij}, while $a_{ij}=0$ otherwise.

We also define binary variables x_{ijk}, y_{ik} and z_{jk} to represent the membership of link l_{ij}, node u_i and node v_j for link community *k*:

$$x_{ijk}=\begin{cases}1 & \text{if } l_{ij}\in \text{community } k\\ 0 & \text{otherwise}\end{cases}\quad y_{ik}=\begin{cases}1 & \text{if } u_i\in \text{community } k\\ 0 & \text{otherwise}\end{cases}\quad z_{jk}=\begin{cases}1 & \text{if } v_j\in \text{community } k\\ 0 & \text{otherwise}\end{cases}$$

The link community detection problem of bipartite network can be formulated into the following integer programming model—Model 1.

$$\max H = \frac{1}{M}\sum_{k=1}^{K}\frac{\left(\sum_{i=1}^{p}\sum_{j=1}^{q}x_{ijk}\right)^2}{\left(\sum_{i=1}^{p}y_{ik}\right)\left(\sum_{j=1}^{q}z_{jk}\right)} \tag{1}$$

$$s.t.\begin{cases}\sum_{k=1}^{K}x_{ijk}=a_{ij} \quad i=1,2,\cdots,p; j=1,2,\cdots,q & (2)\\ x_{ijk}\le y_{ik} \quad i=1,2,\cdots,p; j=1,2,\cdots,q; k=1,2,\cdots,K & (3)\\ x_{ijk}\le z_{jk} \quad i=1,2,\cdots,p; j=1,2,\cdots,q; k=1,2,\cdots,K & (4)\\ y_{ik}\le\sum_{j=1}^{q}x_{ijk} \quad i=1,2,\cdots,p, k=1,2,\cdots K & (5)\\ z_{jk}\le\sum_{i=1}^{p}x_{ijk} \quad j=1,2,\cdots,q, k=1,2,\cdots K & (6)\\ x_{ijk}\in\{0,1\}; i=1,2,\cdots,p; j=1,2,\cdots,q, k=1,2,\cdots,K & (7)\\ y_{ik}\in\{0,1\}; i=1,2,\cdots,p, k=1,2,\cdots,K & (8)\\ z_{jk}\in\{0,1\}; j=1,2,\cdots,q, k=1,2,\cdots,K & (9)\end{cases}$$

The objective function (1) is to maximize the link partition density *H*. Constraint (2) means that every link belongs to one community. If there is no link between node u_i and v_j, then variables $x_{ijk}=0$ for any community *k*. Constraints (3) and (4) indicate that if link l_{ij} belong to community *k*, then its adjacent nodes u_i and v_j must belong to the same community *k*. Constraint (5) and (6) mean that if a node u_i (or v_j) belongs to community *k*, then there is at least one link adjacent to node u_i (or v_j) belonging to community *k*. Constraints (7) (8) (9) indicate that the variables are binary.

Since there are a great many of variables in Model 1, it may have large memory overhead when solving the model directly. To decrease the number of variables used, Model 1 can be expressed by using relationship matrix.

Suppose that $U=\{u_1,u_2,\cdots,u_p\}$, $V=\{v_1,v_2,\cdots,v_q\}$ are two disjoint nodes sets, and $L=\{l_1,l_2,\cdots,l_M\}$ is the link set of bipartite network. Define two incidence matrix RS and RT as follows:

$$RS=\begin{pmatrix} s_{11} & s_{12} & s_{13} & \cdots & s_{1M} \\ s_{21} & s_{22} & s_{23} & \cdots & s_{2M} \\ s_{31} & s_{32} & s_{33} & \cdots & s_{3M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{p1} & s_{p2} & s_{p3} & \cdots & s_{pM} \end{pmatrix}$$

where

$$s_{im}=\begin{cases} 1 & \text{if } u_i \text{ is an endpoint of link } l_m \\ 0 & \text{otherwise} \end{cases}$$

$$RT=\begin{pmatrix} t_{11} & t_{12} & t_{13} & \cdots & t_{1M} \\ t_{21} & t_{22} & t_{23} & \cdots & t_{2M} \\ t_{31} & t_{32} & t_{33} & \cdots & t_{3M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{q1} & t_{q2} & t_{q3} & \cdots & t_{qM} \end{pmatrix}$$

$$t_{jm}=\begin{cases} 1 & \text{if } v_j \text{ is an endpoint of link } l_m \\ 0 & \text{otherwise} \end{cases}$$

Define the binary variables as follows:

$$x_{mk}=\begin{cases} 1 & \text{if } l_m \in \text{community } k \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ik}=\begin{cases} 1 & \text{if } u_i \in \text{community } k \\ 0 & \text{otherwise} \end{cases}$$

$$z_{jk}=\begin{cases} 1 & \text{if } v_j \in \text{community } k \\ 0 & \text{otherwise} \end{cases}$$

Based on the incidence matrix and the above variables, the link community detection problem of bipartite network can be reformulated into the following integer nonlinear programming model, Model 2.

$$\max H=\frac{1}{M}\sum_{k=1}^{K}\frac{\left(\sum_{m=1}^{M}x_{mk}\right)^2}{\left(\sum_{i=1}^{p}y_{ik}\right)\left(\sum_{j=1}^{q}z_{jk}\right)} \tag{10}$$

$$
s.t.\begin{cases}
\sum_{k=1}^{K} x_{mk} = 1 \quad m = 1, 2, \cdots, M & (11) \\
\sum_{m=1}^{M} s_{im} x_{mk} \le N y_{ik} \quad i = 1, 2, \cdots, p; k = 1, 2, \cdots, K & (12) \\
\sum_{m=1}^{M} t_{jm} x_{mk} \le N z_{jk} \quad j = 1, 2, \cdots, q; k = 1, 2, \cdots, K & (13) \\
y_{ik} \le \sum_{m=1}^{M} s_{im} x_{mk} \quad i = 1, 2, \cdots, p, k = 1, 2, \cdots K & (14) \\
z_{jk} \le \sum_{m=1}^{M} t_{jm} x_{mk} \quad j = 1, 2, \cdots, q, k = 1, 2, \cdots K & (15) \\
x_{mk} \in \{0,1\}; m = 1, 2, \cdots, M; k = 1, 2, \cdots, K & (16) \\
y_{ik} \in \{0,1\}\}; i = 1, 2, \cdots, p, k = 1, 2, \cdots, K & (17) \\
z_{jk} \in \{0,1\}; j = 1, 2, \cdots, q, k = 1, 2, \cdots, K & (18)
\end{cases}
$$

Where N is the number of nodes in the network, $N = p + q$. The objective function (10) is to maximize the link partition density. Constraint (11) means that every link belongs to one community. Constraint (12) (13) mean that, if there is some adjacent links of node u_i (v_j) belonging to community *k*, then node u_i (v_j) must belong to the same community *k*. Constraints (14) (15) mean that if node u_i (v_j) belongs to community *k*, then at least one link adjacent to this node must belong to community *k*. Constraints (16) (17) (18) indicate that the variables are binary.

In Model 1 and Model 2, since every link can belong to one and only one community, we might obtain the result that a pair of nodes belongs to two communities, but the link between this pair of nodes belongs to only one community. To reduce this drawback, we can revise Model 2 into the following model—Model 3.

$$
\max H = \frac{1}{\sum_{k=1}^{K}\sum_{m=1}^{M} x_{mk}} \sum_{k=1}^{K} \frac{\left(\sum_{m=1}^{M} x_{mk}\right)^2}{\left(\sum_{i=1}^{p} y_{ik}\right)\left(\sum_{j=1}^{q} z_{jk}\right)} \tag{10′}
$$

$$
s.t.\begin{cases}
\sum_{k=1}^{K} x_{mk} \ge 1 \quad m = 1, 2, \cdots, M & (11′) \\
\sum_{m=1}^{M} s_{im} x_{mk} \le N y_{ik} \quad i = 1, 2, \cdots, p; k = 1, 2, \cdots, K & (12′) \\
\sum_{m=1}^{M} t_{jm} x_{mk} \le N z_{jk} \quad j = 1, 2, \cdots, q; k = 1, 2, \cdots, K & (13′) \\
y_{ik} \le \sum_{m=1}^{M} s_{im} x_{mk} \quad i = 1, 2, \cdots, p, k = 1, 2, \cdots K & (14′) \\
z_{jk} \le \sum_{m=1}^{M} t_{jm} x_{mk} \quad j = 1, 2, \cdots, q, k = 1, 2, \cdots K & (15′) \\
x_{mk} \in \{0,1\}; m = 1, 2, \cdots, M; k = 1, 2, \cdots, K & (16′) \\
y_{ik} \in \{0,1\}; i = 1, 2, \cdots, p, k = 1, 2, \cdots, K & (17′) \\
z_{jk} \in \{0,1\}; j = 1, 2, \cdots, q, k = 1, 2, \cdots, K & (18′)
\end{cases}
$$

In model 3, the constraint (11') means that every link must belong to at least one community.

Using model 3, we can partition the network in **Figure 1** into two communities, and link (3, 10) belongs to two communities. Each community is a complete bipartite subnetwork, and the optimal objective function value is 1.

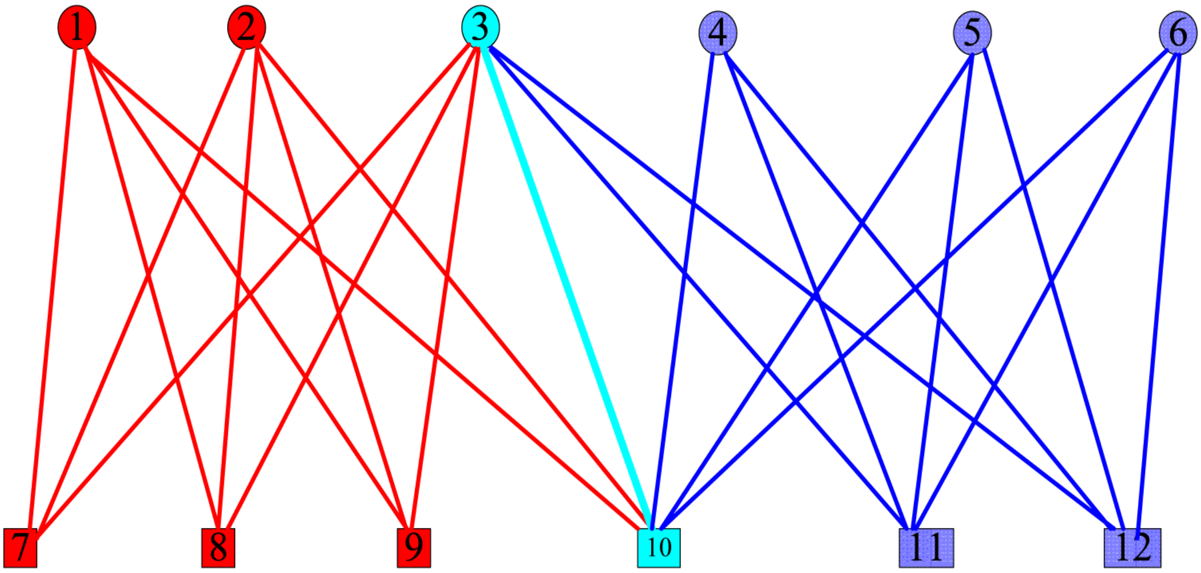

Figure 1. The bipartite network consists of two overlapping communities, each community is a complete bipartite network, they are overlapped by nodes 3,10 and link (3,10). This bipartite network can be partitioned into two communities by model 3, and the objective function value is 1.

2.3. Genetic Algorithm for Link Community Detection of Bipartite Network

Although we can solve Model 2 or Model 3 to partition a bipartite network into link communities for small size of bipartite network. It is difficult to solve the integer programming model for large bipartite networks which might be a NP-hard problem. In addition, most of the algorithms for community detection need some *priori* knowledge about the community structure like the number of communities which is impossible to know in real-life networks. In [12], the authors propose a link community detection algorithm based on the ideas of genetic algorithm and self-organize map (SOM) algorithm, which aims to find the best link community structure by maximizing the network partition density. The algorithm does not need any *priori* knowledge about the number of communities, which makes the algorithm useful in real-life networks. The algorithm outputs the final link community structure and its corresponding overlapping nodes as the result and does not impose further processing on the output. In the following, we will design another genetic algorithm for link community detection of bipartite network.

First of all, we need to design a chromosome representation suitable for the link community detection problem. In our implementation, the chromosome is represented by a matrix $B=\left(b_{m,c}\right)$, where $m=1,2,\cdots,M$, and $c=1,2,\cdots,K$. Each element $b_{m,c}$ is the strength with which a link l_m belongs to a community *c*. Note that $b_{m,c}$ ranges in the interval [0.0, 1.0]. Each link of the bipartite network is subject to the following constraint:

$$\sum_{c=1}^{K} b_{m,c} = 1. \tag{19}$$

Equation (19) represents normalization to 1.0 of link factors of belonging to the communities.

For each chromosome, we design a partition matrix $D=\left(d_{m,c}\right)$, where $m=1,2,\cdots,M$, and $c=1,2,\cdots,K$. Each element $d_{m,c}$ is either 0 or 1. Where $d_{m,c}=1$ if the link l_m is assigned to community *c*, otherwise, link l_m is not assigned to community *c*. Matrix *D* can be calculated from matrix *B* according to the following equation:

$$d_{m,c} = \begin{cases} 1 & \text{if } b_{m,c} = \max\limits_{1\le s\le K} b_{m,s}. \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

The bipartite network is represented by two incidence matrixes RS and RT, two weighted incidence matrixes ZS and ZT, link adjacent matrix *A* and weighted link adjacent matrix *Q*.

$$ZS = \begin{pmatrix} \frac{1}{\sqrt{d(u_1)}} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{\sqrt{d(u_2)}} & 0 & \cdots & 0 \\ 0 & 0 & \frac{1}{\sqrt{d(u_3)}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{\sqrt{d(u_p)}} \end{pmatrix} \cdot RS$$

$$ZT = \begin{pmatrix} \frac{1}{\sqrt{d(v_1)}} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{\sqrt{d(v_2)}} & 0 & \cdots & 0 \\ 0 & 0 & \frac{1}{\sqrt{d(v_3)}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{\sqrt{d(v_q)}} \end{pmatrix} \cdot RT$$

where d_{u_i} and d_{v_j} represent the nodes' degree of nodes u_i and v_j, which is the number of links incident to nodes u_i and v_j respectively.

The link adjacent matrix *A* and the weighted link adjacent matrix *Q* can be calculated by the following equations:

$$A = (RS)^{\mathrm{T}}(RS) + (RT)^{\mathrm{T}}(RT)$$

$$Q = (ZS)^{\mathrm{T}}(ZS) + (ZT)^{\mathrm{T}}(ZT)$$

The weighted link adjacent matrix *Q* means the probability for a random walker go from one link to one of its adjacent links across their common node. And this can be regarded as the possibility of two adjacent links belonging to the same community.

2.3.1. The Genetic Algorithm Main Functions

- Input

Input the number of nodes *p* for node set $|U|$ and *q* for node set $|V|$ respectively, and the number of links *M* of the link set $|E|$ in bipartite network, the maximum number of communities *K*, parameters α, β, θ, where $\alpha \in (0,1), \beta \in (0,1)$.

Input the incident matrixes *RS*, *RT*. Calculate the weighted incident matrixes *ZS* and *ZT*, the link adjacent matrix *A*, and the weighted link adjacent matrix *Q*. Given the number of individuals *N*, the maximum epochs $epoch_{\max}$, mutation probability $prob_{mutation}$.

- Output

Output the link partition matrix D^* and its fitness value H^*, two nodes set partition matrixes F_1, F_2. Partition the network into communities according to F_1, F_2.

- Initialization: $t = 0$.

Randomly generate initial population $B_1(t), B_2(t), \cdots, B_N(t)$, and give random initial values of D^* and its fitness H^*.

- Step 1. Population Fitness

For every individual $B_1(t), B_2(t), \cdots, B_N(t)$, calculate the partition matrix $D_1(t), D_2(t), \cdots, D_N(t)$, and their fitness value (partition link density value) $H_1(t), H_2(t), \cdots, H_N(t)$.

- Step 2. Population Sorting

Sort $B_1(t), B_2(t), \cdots, B_N(t)$ according to their fitness values in decreasing order. Suppose the sorted chromosomes are $B_1(t), B_2(t), \cdots, B_N(t)$, where $H_1(t) \geq H_2(t) \geq \cdots \geq H_N(t)$.

If $H_1(t) > H^*$, then, $D^* = D_1(t)$, $H^* = H_1(t)$.

If $t = epoch_{\max}$, stop, output D^* and H^*, and calculate the two corresponding node sets partition matrix F_1, F_2. Otherwise, go to Step 3.

- Step 3. Population Crossover

For $i = 1, \cdots, \left\lfloor \frac{N}{2} \right\rfloor$, let $B_i(t)$ and $B_{\left\lfloor \frac{N}{2} \right\rfloor + i}(t)$ cross over to produce two temporary individuals (matrixes) $W_i(t)$ and $W_{\left\lfloor \frac{N}{2} \right\rfloor + i}(t)$. If N is an odd number, let $W_N(t) = B_N(t)$.

- Step 4. Population Mutation

Random select $prob_{mutation} N$ temporary individuals (temporary matrices), do mutation operation on each temporary individual.

- Step 5. Population Self Organize Mapping

For each temporary individual, do self organize mapping operation on it.

- Step 6. Population Normalization

For each temporary individual, do normalization on it. Denote the normalized individuals by $B_1(t+1), B_2(t+1), \cdots, B_N(t+1)$. Let $t = t+1$, go to step 1.

2.3.2. Partition Matrix and Fitness Evaluation

For every individual B_i, calculate the partition matrix D_i according to the Formula (20).

For each community *s*, $1 \leq s \leq K$, let $D_i(:, s)$ be the *s*-th column of matrix D_i.

Then $E_{i1}(s) = RS \cdot D_i(:, s)$ is a column vector whose element is a non-negative integer. A non-zero element in $E_{i1}(s)$ represents that the corresponding node of the node set $|U|$ belongs to community *s*. $E_{i2}(s) = RT \cdot D_i(:, s)$ is a column vector whose element is a non-negative integer. A non-zero element in $E_{i2}(s)$ represents that the corresponding node of the node set $|V|$ belongs to community *s*.

Let $F_{i1}(s)$ and $F_{i2}(s)$ be 0-1 vectors, $f_{i1}(j,s) = 1$ (or $f_{i2}(j,s) = 1$) whenever $e_{i1}(j,s) \geq 1$ (or $e_{i2}(j,s) \geq 1$). $f_{i1}(j,s) = 1$ (or $f_{i2}(j,s) = 1$) means that node u_j (or v_j) belongs to community *s*. The fitness value of individual B_i is defined by the link partition density of matrix D_i, which can be calculated by the following equation:

$$H_i = \frac{1}{\sum_{s=1}^{K} \sum_{j=1}^{M} D_i(j,s)} \sum_{s=1}^{K} \frac{\left(\sum_{j=1}^{M} D_i(j,s) \right)^2}{\left(\sum_{j=1}^{p} F_{i1}(j,s) \right) \left(\sum_{j=1}^{q} F_{i2}(j,s) \right)}$$

2.3.3. Population Sorting

Sort $B_1(t), B_2(t), \cdots, B_N(t)$ according to their fitness values in decreasing order. Suppose the sorted chromosomes are $B_1(t), B_2(t), \cdots, B_N(t)$, where $H_1(t) \geq H_2(t) \geq \cdots \geq H_N(t)$.

If $H_1(t) > H^*$, then, $D^* = D_1(t)$, $H^* = H_1(t)$.

2.3.4. Population Crossover

For $i = 1, 2, \cdots, \left\lfloor \frac{N}{2} \right\rfloor$, do crossover operation on $B_i(t)$ and $B_{\left\lfloor \frac{N}{2} \right\rfloor + i}(t)$ by the following rules: Randomly select a column *s*, revise the *s*-th column of $B_{\left\lfloor \frac{N}{2} \right\rfloor + i}(t)$ by the *s*-th column of $B_i(t)$, and obtain two new temporal individuals $W_i(t)$ and $W_{\left\lfloor \frac{N}{2} \right\rfloor + i}(t)$, where $W_i(t) = B_i(t)$.

In this paper, we revised the *s*-th column of $B_{\left\lfloor \frac{N}{2} \right\rfloor + i}(t)$ by adding a fraction of the *s*-th column of $D_i(t)$ (where $D_i(t)$ is the partition matrix corresponding to $B_i(t)$), that is,

$$W_{\left\lfloor \frac{N}{2} \right\rfloor + i}(t)(:,c) = \begin{cases} B_{\left\lfloor \frac{N}{2} \right\rfloor + i}(:,s) + 0.1 \cdot D_i(:,s) & \text{if } c = s. \\ B_{\left\lfloor \frac{N}{2} \right\rfloor + i}(:,c) & \text{if } c \neq s. \end{cases}$$

2.3.5. Population Mutation

According to the mutation probability $prob_{mutation}$, randomly select $prob_{mutation} \cdot N$ temporal individuals, do mutation operation on each selected individual.

For each selected temporal individual $W_i(t)$, randomly select two parameters j_1, j_2, $1 \le j_1, j_2 \le M$. There are three mutation rules can be used in this genetic algorithm, *i.e.* exchange the j_1-th row and the j_2-th row in $W_i(t)$, or replace the j_1-th row by the j_2-th row in $W_i(t)$, or replace the elements of the j_1-th row with a randomly selected number in [0.0,1.0]. Three rules lead to no significant difference in this genetic algorithm. In the following simulation, we replace the j_1-th row with the j_2-th row in $W_i(t)$. The other elements in $W_i(t)$ remain unchanged.

2.3.6. Population Self Organizing Map

For every link, find the community it belongs to and calculate its community ID variance. If the community ID variance of a link is larger than a threshold, then increase the weights of this link to its community and the weights of its neighbor links to the same community. If the community ID variance of a link is smaller than the threshold value, then decrease the weights of the link to its community and the weights of its neighbor links to the same community. This process can improve the quality of the partition by eliminating wrongly placed links.

For $i = 1, \cdots, N$, do Self Organizing Map (SOM) operations on individual (chromosome) W_i as follows:

- According to temporal matrix W_i, calculate its partition matrix D_i';
- For $j = 1, \cdots, M$, do the following operation on link l_j.
- Find the community ID that link l_j belongs to. The community ID corresponds to the maximum element in the *j*-th row of D_i' (the maximum element must be 1). Suppose the maximum element in the *j*-th row of D_i' is in the *s*-th column, which is $D_i'(j,s)$. This means that link l_j belongs to community P_s.
- Calculate the total number $TN(l_j)$ of adjacent links of l_j (including link l_j), and the number of its adjacent links in $TN(l_j)$ belonging to community P_s (denoted by $IN(l_j)$). $TN(l_j)$ is equal to the sum of elements in the *j*-th row of matrix *A*, which can be expressed by $TN(l_j) = A(j,:) \cdot I$, where $I = (1,1,\cdots,1)^{\mathrm{T}}$, and $TN(l_j)$ can be calculated by the equation $IN(l_j) = A(j,:) \cdot D_i'(:,s)$.
- Calculate the community ID variance $CV(l_j)$ of link l_j by the following equation.

$$CV(l_j) = \frac{IN(l_j)}{TN(l_j)}$$

- If $CV(l_j) \ge \theta$, then

$$W_i(:,s) = W_i(:,s) + Q(:,j) \cdot \alpha - (I - A(:,j)) \cdot \beta$$

Else,

$$W_i(:,s) = W_i(:,s) - Q(:,j)\cdot\beta$$

where α and β are adjustable parameters which can decrease with the step *t*. In this paper, we let

$$\alpha = \alpha - \frac{t}{epoch_{\max}}(\alpha - 0.1),\ \ \beta = \beta - \frac{t}{epoch_{\max}}(\beta - 0.05)$$

In the above equation, if an element is negative, then we set it to be 0.01

2.3.7. Normalization

For $i = 1,2,\cdots N$, do normalization on each row of temporal matrix W_i so that the sum of row elements in temporal matrix is 1. Let the normalized matrix be B_{i+1}.

3. Numerical Experiments

In this section, we apply the genetic algorithm to both artificial bipartite networks and several well studied real-world bipartite networks, and analyze the results in terms of classification accuracy and ability of detecting meaningful communities. The algorithm is implemented by Matlab version 7.1.

3.1. Chain of Complete Bipartite Network

We test our algorithm on a type of exemplar networks, that is, chains of complete bipartite network. This network consists of many heterogeneous complete bipartite networks, connected through single nodes (**Figure 2**). Each complete bipartite network $C_i = (U_i, V_i, L_i)$ $(i = 1,2,\cdots,K)$ is a bipartite network, where there is a link between any pair of nodes $(u,v), u\in U_i, v\in V_i$. Assume that C_i has $s_i + t_i$ nodes and $L_i = s_i * t_i$ links, then the network has a total of $N = \sum_{i=1}^{K}(s_i + t_i) - K + 1$ nodes and $M = \sum_{i=1}^{K} L_i$ links. The network has a clear link bipartite modular structure where each community corresponds to a single bipartite complete network, thus the optimal partition density is 1. Using the genetic algorithm above, we can easily detect the optimal partition and identify the overlapping nodes. In this paper, we use a network consists of two (3,4)- complete bipartite networks, one (4,5)- complete bipartite network, one (4,6)- complete bipartite network, and one (5,5) complete bipartite network, the optimal partition results are obtained and described in **Figure 2**.

3.2. Real-World Networks

In this subsection, we validate our algorithm on some real-world networks.

The Southern Women Network During the 1930s, ethnographers Davis, Stubbs Davis, St. Clair Drake, Gardner, and Gardner collected data on social stratification in the town of Natchez, Mississippi. One of their work is collecting data on women's attendance to social events in the town [24]. They constructed the famous women-event bipartite network and analyze it. Since then the women-event bipartite network has become a de facto standard for discussing bipartite networks in the social science [12] [15] [20] [21] [24]-[27].

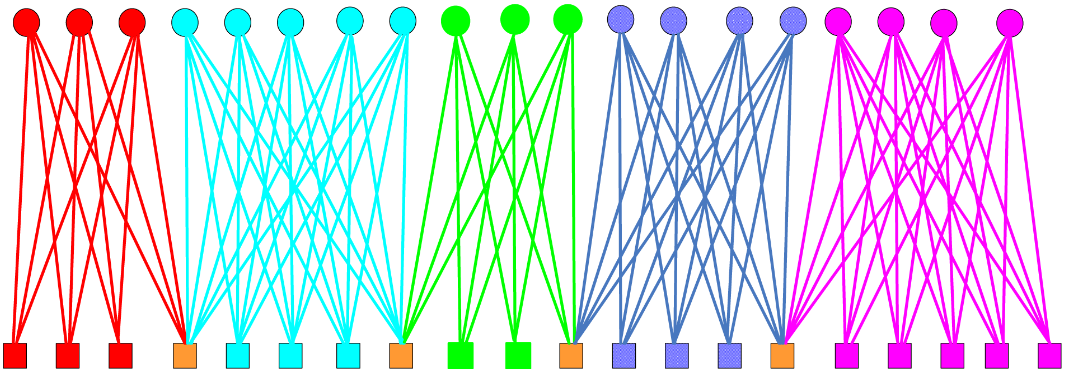

Figure 2. The chain of heterogeneous complete bipartite network. Each community is a complete bipartite network, and two adjacent communities are overlapped by one node.

Guimerà [15] has analyze the modules of both women and events by three methods: unweighted projection, weighted projection, and bipartite approach. The first method did not capture the true modular structure of the network. The second and third methods capture the two-module structure except one woman being partitioned wrong.

We applied the proposed method to the women-event network, using the parameters $K = 2$, $N = 200$, $p = 0.2$, $\theta = 0.3$, $\alpha = 0.7$, $\beta = 0.2$, $T = 800$. The result is illustrated in **Figure 3**. In this result, 18 women and 14 events are partitioned into two communities, where 4 events are overlapped. The average link density is 0.5610. In the women-event bipartite network, four event nodes B6, B7, B8, B9 colored by yellow are overlapped and belong to two communities. Comparing with the results obtained by Guimerà [15], the overlapped communities obtained by our method are more reasonable. When we partition the women-event network into four link communities using the parameters $K = 4$, $N = 200$, $p = 0.2$, $\theta = 0.3$, $\alpha = 0.7$, $\beta = 0.2$, $T = 800$, we can obtain the maximum average link density 0.683, The result is illustrated in **Figure 4**. Six yellow nodes are overlapped, where B_5, B_{11} belong to two communities, $B_6, B_7 B_8$ belong to three communities and B_9 belong to four communities.

3.3. The Scotland Corporate Interlock Network

The Scotland corporate interlock network describe the corporate interlocks in Scotland in the beginning of the twentieth century (1904-1905). The network consists of 244 nodes and 356 edges. The 244 nodes are divided into two parts, where 136 nodes indicate the board members who held multiple directorships, and 108 nodes indicate the firms). The edges exist between each firm and its board members. The largest component of the Scotland corporate interlock network contains 131 directors and 86 firms, forming many communities.

We applied the proposed method to the largest component of Scotland corporate interlock network, using the parameters $K = 20$, $N = 400$, $p = 0.2$, $\theta = 0.3$, $\alpha = 0.8$, $\beta = 0.2$, $T = 2000$. In the experiment, we divides the network into 20 communities, and the link community density is 0.24777. With the number of communities K increasing, the link community density obtained by our algorithm increase. When we use the para-

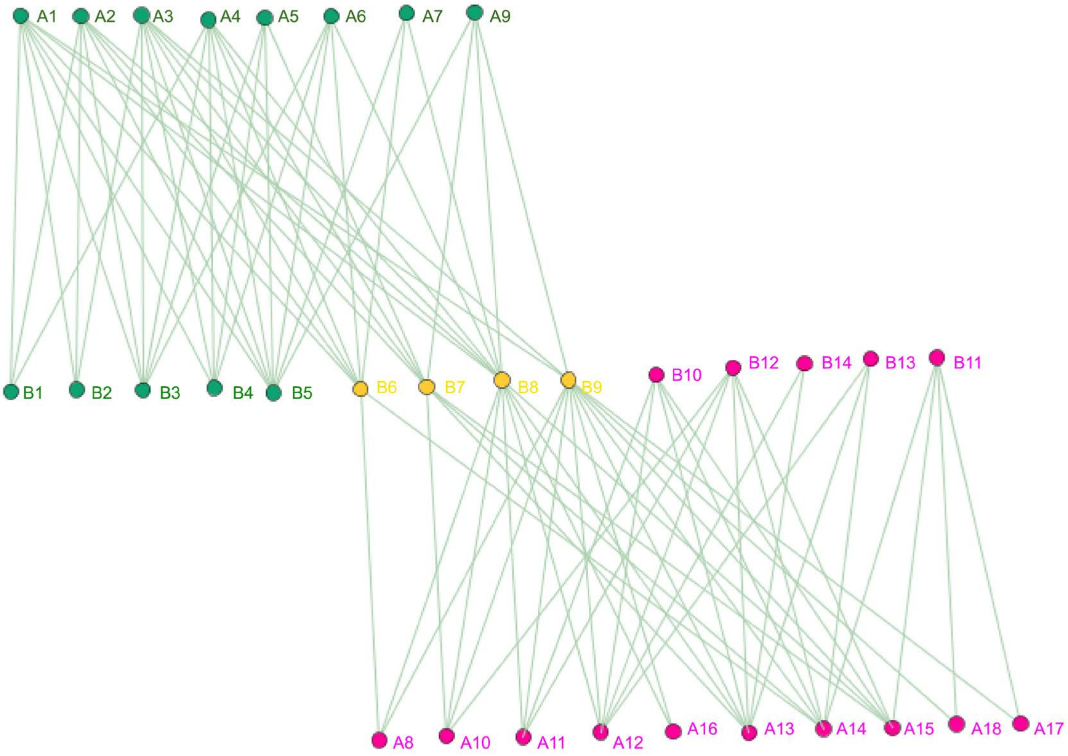

Figure 3. Result of the women-event networks partition into two link communities, where four yellow nodes *B*6; *B*7; *B*8; *B*9 belong to two communities.

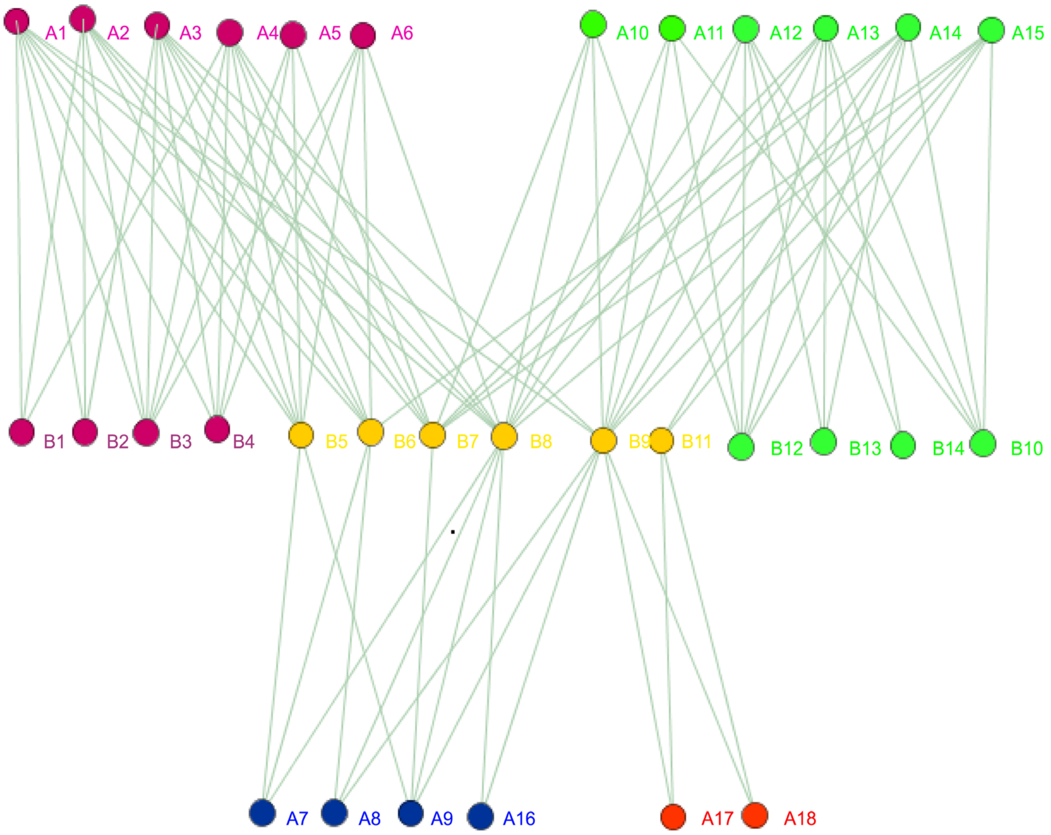

Figure 4. Result of the women-event networks partition into 4 link communities. Six yellow nodes are overlapped, where B_5; B_{11} belong to two communities, B_6; B_7; B_8 belong to three communities and B_9 belong to four communities.

meters $K = 36$, $N = 100$, $p = 0.2$, $\theta = 0.3$, $\alpha = 0.8$, $\beta = 0.2$, $T = 2000$. We can obtained the maximum link community density 0.3553. If we increasing parameter K from 36 to 40, we can also partitioned the network into 36 link communities, the maximum link community density is also 0.3553. Since the real number of communities is 36 [25], Our results mean that we can find the optimal community solution by our algorithm.

4. Conclusion and Discussion

Bipartite network community structure is one of the main characteristics of bipartite networks and very helpful for understanding the functions of these networks. In this paper, we investigate the link community detection problem of bipartite network and propose a quantity function for link community detection of bipartite network. We formulate the link community identification problem of bipartite network into an integer programming model by maximizing the quantity function. Furthermore, we design a genetic algorithm for solving the link community detection problem and conduct validation experiments on some simulated and real-world networks. The extensive computational results demonstrate that our model and algorithm can detect overlapping communities. Using our model and algorithm, we can not only find the node overlapping communities but also the link overlapping communities in bipartite networks. Although we only investigate the unweighted bipartite networks, the model and algorithm can also be extended to deal with weighted bipartite networks.

Acknowledgements

This work is supported by National Natural Science Foundation of China under Grant No. 11131009. It is also supported by the Funding Project for Academic Human Resources Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality (CIT&TCD20130327).

References

[1] Albert, R. and Barabási, A.L. (2002) Statistical Mechanics of Complex Networks. *Reviews of Modern Physics*, **74**, 47-97. http://dx.doi.org/10.1103/RevModPhys.74.47

[2] Newman, M.E.J. (2003) The Structure and Function of Complex Networks. *SIAM Review*, **45**, 167-256. http://dx.doi.org/10.1137/S003614450342480

[3] Newman, M.E.J. and Girvan, M. (2004) Finding and Evaluating Community Structure in Networks. *Physical Review E*, **69**, Article ID: 026113. http://dx.doi.org/10.1103/physreve.69.026113

[4] Hu, Y., Chen, H., Zhang, P., Li, M., Di, Z. and Fan, Y. (2008) Comparative Definition of Community and Corresponding Identifying Algorithm. *Physical Review E*, **78**, Article ID: 026121. http://dx.doi.org/10.1103/PhysRevE.78.026121

[5] Fortunato, S. (2010) Community Detection in Graph. *Physics Reports*, **486**, 75-174. http://dx.doi.org/10.1016/j.physrep.2009.11.002

[6] Newman, M.E.J. (2012) Communities, Modules and Large-Scale Structure in Networks. *Nature Physics*, **8**, 25-31. http://dx.doi.org/10.1038/nphys2162

[7] Zhang, S., Jin, G., Zhang, X.S. and Chen, L. (2007) Discovering Functions and Revealing Mechanisms at Molecular Level from Biological Networks. *Proteomics*, **7**, 2856-2869. http://dx.doi.org/10.1002/pmic.200700095

[8] Ahn, Y.Y., Bagrow, J.P. and Lehmann, S. (2010) Link Communities Reveal Multi-Scale Complexity in Networks. *Nature*, **466**, 761-764. http://dx.doi.org/10.1038/nature09182

[9] Evans, T.S. and Lambiotte, R. (2009) Line Graphs, Link Partitions and Overlapping Communities. *Physical Review E*, **80**, Article ID: 016105. http://dx.doi.org/10.1103/PhysRevE.80.016105

[10] Evans, T.S. (2010) Clique Graphs and Overlapping Communities. *Journal of Statistical Mechanics*: *Theory and Experiment*, 12037. http://dx.doi.org/10.1088/1742-5468/2010/12/P12037

[11] Evans, T.S. and Lambiotte, R. (2010) Line Graphs of Weighted Networks for Overlapping Communities. *The European Physical Journal B*, **77**, 265-272. http://dx.doi.org/10.1140/epjb/e2010-00261-8

[12] Li, Z., Zhang, X.S., Wang, R.S., Liu, H. and Zhang, S. (2013) Discovering Link Communities in Complex Networks by an Integer Programming Model and a Genetic Algorithm. *PLoS ONE*, **8**, e83739. http://dx.doi.org/10.1371/journal.pone.0083739

[13] Zhang, S., Wang, R.S. and Zhang, X.S. (2007) Identification of Overlapping Community Structure in Complex Networks Using Fuzzy *c*-Means Clustering. *Physica A*, **374**, 483-490. http://dx.doi.org/10.1016/j.physa.2006.07.023

[14] He, D.X., Liu, D., Zhang, W., Jin, D. and Yang, B. (2012) Discovering Link Communities in Complex Networks by Exploiting Link Dynamics. *Journal of Statistical Mechanics*, **2012**, Article ID: P10015. http://dx.doi.org/10.1088/1742-5468/2012/10/P10015

[15] Guimmerà, R., Sale-Pardo, M. and Nunes Amaral, L.A. (2007) Module Identification in Bipartite and Directed Networks. *Physical Review E*, **76**, Article ID: 036102. http://dx.doi.org/10.1103/PhysRevE.76.036102

[16] Barber, M.J. (2007) Modularity and Community Detection in Bipartite Networks. *Physical Review E*, **76**, Article ID: 066102. http://dx.doi.org/10.1103/physreve.76.066102

[17] Zhan, W., Zhang, Z., Guan, J. and Zhou, S. (2011) Evolutionary Method for Finding Communities in Bipartite Networks. *Physical Review E*, **83**, Article ID: 066120. http://dx.doi.org/10.1103/PhysRevE.83.066120

[18] Murata, T. and Ikeya, T. (2010) A New Modularity for Detecting One-to-Many Correspondence of Communities in Bipartite Networks. *Advances in Complex Systems*, **13**, 19-31. http://dx.doi.org/10.1142/S0219525910002402

[19] Costa, A. and Hansen, P. (2014) A Locally Optimal Hierarchical Divisive Heuristic for Bipartite Modularity Maximization. *Optimization Letters*, **8**, 903-917. http://dx.doi.org/10.1007/s11590-013-0621-x

[20] Du, N., Wang, B., Wu, B. and Wang, Y. (2008) Overlapping Community Detection in Bipartite Networks. *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Sydney, 9-12 December 2008, 176-179. http://dx.doi.org/10.1109/WIIAT.2008.98

[21] Souam, F., AÏtelhadj, A. and Baba-Ali, R. (2013) Dual Modularity Optimization for Detecting Overlapping Communities in Bipartite Network. *Knowledge and Information Systems*, **40**, 455-488. http://dx.doi.org/10.1007/s10115-013-0644-8

[22] Zhang, Z.Y., Wang, Y. and Ahn, Y.Y. (2013) Overlapping Community Detection in Complex Network Using Symmetric Binary Matrix Factorization. *Physical Review E*, **87**, Article ID: 062803. http://dx.doi.org/10.1103/PhysRevE.87.062803

[23] Zhang, Z.Y. and Ahn, Y.Y. (2015) Community Detection in Bipartite Networks Using Weighted Symmetric Binary Matrix Factorization. *International Journal of Modern Physics C*, **26**, Article ID: 1550096.

[24] Freeman, L.C. (2003) Finding Social Groups: A Meta-Analysis of the Southern Women Data. In: Breiger, R., Carley, C. and Pattison, P., Eds., *Dynamic Social Network Modeling and Analysis*: *Workshop Summary and Papers*, National Research Council, The National Academies Press, Washington DC, 39-97.

[25] Chen, B.L., Chen, L., Zhou, S.R. and Xu, X.L. (2013) Detecting Community Structure in Bipartite Networks Based on Matrix Factorization. *International Journal of Wireless and Mobile Computing*, **6**, 599-607. http://dx.doi.org/10.1504/IJWMC.2013.057576

[26] Liu, X. and Murata, T. (2010) An Efficient Algorithm for Optimizing Bipartite Modularity in Bipartite Networks. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **14**, 408-415.

[27] Murata, T. (2009) Detecting Communities from Bipartite Networks Based on Bipartite Modularities. *Proceedings of the* 2009 *International Conference on Computational Science and Engineering*, Vancouver, 29-31 August 2009, 50-57.