A evolutionary method for finding communities in bipartite networks

Weihua Zhan[1],* Zhongzhi Zhang[2,3],† Jihong Guan[1],‡ and Shuigeng Zhou[2,3]
[1]*Department of Computer Science and Technology, Tongji University, 4800 Cao'an Road, Shanghai 201804, China*
[2]*School of Computer Science, Fudan University, Shanghai 200433, China and*
[3]*Shanghai Key Lab of Intelligent Information Processing, Fudan University, Shanghai 200433, China*
(Dated: November 16, 2010)

Common practice in community structure detection is to develop different methods for different classes of networks. Here, we first show that unipartite networks and directed networks can be uniformly represented as bipartite networks, and their modularity completely consist with that for bipartite networks. To optimize the bipartite modularity, we then present a modified adaptive genetic algorithm, called as MMOGA, which is especially suited for community structure detection. In MMOGA, we introduce a new measure for the informativeness of a locus instead of the standard deviation, which can exactly determine those loci to mutate. This measure is the bias between the distribution of a locus over the current population and the uniform distribution of the locus, i.e., Kull-back Divergence between them. Moreover, we develop a reassignment technique for differentiating the informative state a locus has attained from the random state at initial phase. Also we present a modified mutation rule which incorporating related operation can guarantee MMOGA the convergence to the global optima and can speed up the convergence process. Experimental results show that MMOGA is superior to MOGA and standard genetic algorithms as well as BRIM when applied to bipartite networks.

PACS numbers: 89.75.Hc, 02.10.Ox, 02.50.-r

I. INTRODUCTION

Complex network has gain overwhelming popularity as a tool for understanding various complex systems from diverse fields, including technical, natural and social science, etc., which provides a unified perspective and unified methods for studying these systems through modeling them as networks with nodes and edges respectively representing their units and interactions between units [1–6]. Generally, networks can be classified into unipartite, bipartite and multipartite networks, considering types of their nodes. As a typical class of real-world networks, bipartite networks, compared to unipartite ones, consist of two types of nodes and edges presenting only between distinct types. Examples of bipartite networks come from various fields, including scientific collaborate networks, actor-movie networks and protein-protein interaction networks [1, 2, 9–11]. Multipartite networks with more than three types of nodes, are occasionally seen[7, 8].

It has been discovered [9] that most of real networks share the local clustering feature, i.e., community structure, which describes that groups of tight-knit nodes are mutually connected to each other with sparser edges. These groups of nodes are generally referred to as communities or modules. From a topological point of view, a community may correspond to a function unit because of its structurally relative independence. In turn, community structure can critically affect dynamics of networks. Therefore, identification of communities is playing the key role in numerous studies such as predicating protein function[12] and determing dynamics of systems [13–15]. The last few years have witnessed tremendous efforts in this domain [10, 15–26](useful reviews include Refs. [27, 28]). Most of these are dedicated to deal with unipartite networks while little attention has been paid to directed networks [23, 24] and bipartite ones [24–26].

It is of interest that unipartite networks and directed networks can be represented by bipartite networks as we shall show. Thus, detecting communities in unipartite networks or in directed networks can be transformed into the task in bipartite networks. Given a bipartite modularity, those methods based on modularity maximization [16–19], in principle, can be applied to bipartite networks. However, they are expected to affected by the resolution limit [29, 30] as in the unipartite case, which may result in the degeneracy problem [31] there existing an exponential number of distinct of high-scoring partitions. This poses a challenge for the methods returning one solution. Instead, we present a modified adaptive genetic algorithm to optimize the bipartite modularity [26]. The evolutionary method can return a better solution in a shorter time. Moreover, the method also can return multiple better solutions, which enables us to have a chance to evaluate the reliability of solutions. Additionally, it bears a potential merit that it admits to obtain a better solution by combining different solutions when the degeneracy problem is severe.

For a bipartite network, there exists different conceptual understanding of the community structure. A point of view for communities in the network is to consider each composed of two types of nodes with dense edges across

*Electronic address: 08zhanwh@tongji.edu.cn
†Electronic address: zhangzz@fudan.edu.cn
‡Electronic address: jhguan@tongji.edu.cn

them, which is similar to that in unipartite cases [26]. Alternative view is that any community should contain only one type of nodes, which are closely connected through co-participation in many communities that consist of another type of nodes [24]. Guided by this view, the usual approach to identify communities is to project the bipartite network onto one specific unipartite network as needed, and then identify communities in the projection. Guimerà et al [24] recently presented a method for identifying communities of one type of nodes against other type of nodes with a known community structure.

In this paper, we focus on dealing with the problem of identifying communities in the former view. We present a modified adaptive genetic algorithm based on mutation only genetic algorithm (MOGA) that are parameter-free unlike traditional genetic algorithm. The method has no need to know the number of communities and their sizes in advance. In section II, we first give a short review on Barber's modularity and then show unpartite networks and directed networks can be uniformly represented by bipartite networks. In section III A, we describe the frame of MOGA, and introduce a new measure for selecting loci to mutate in III B, as well as we develop the reassignment technique in III C. We also discuss the problem of selecting population size in III D and that of convergence in III E. In section IV, we apply the new algorithm to several networks including model bipartite network, southern women network and Scotland interlock network. At last, the conclusion is given.

II. BIPARTITE MODULARITY

Modularity introduced by Newman and Girvan [10] aims at quantifying the goodness of a particular division of a given network, which has been widely accepted as a benchmark index to measure and to compare accuracy of various methods of community detection. The definition of this quantity is based on the idea that community structure definitely means a statistically surprising arrangement of edges, that is, the number of actual edges within communities should be significantly beyond that of expected edges of a null model. In turn, a null model should have the same number of nodes and degree distribution as the original network while edges are placed by chance.

Let k_i be the degree of nodes i, M be the total number of edges, then in the null model [18] the probability for an edge presenting between nodes i and j is $\frac{k_i k_j}{2M}$. Then modularity quantifies the extent, relative to the null model network, to the number of actual edges exceeding the expectation, which can be formulated as follows:

$$Q = \frac{1}{2M}\sum_{i=1}^{n}\sum_{j=1}^{n}(A_{i,j} - \frac{k_i k_j}{2M})\delta(g_i, g_j) \qquad (1)$$

where Q is the sum of the difference over all groups of the particular partition, $A_{i,j}$ is the adjacent matrix element, g_i represents the group the node i is assigned to, and the δ function takes the value of 1 if g_i equals to g_j, 0 otherwise.

The maximum value of Q is 1, and values approaching 1 indicate strong community structure. On the contrary, when the number of within-community edges is no better than random, $Q \leq 0$ and values approaching 0 imply weaker community structure or indivisibility. Consequently, community structure detection can be formulated as a problem of maximization of modularity, which has prevailed in the research area.

The modularity above-mentioned is actually designed for unipartite networks. To be suitable for various networks, several varieties of modularity based on different null model has been proposed, including weighted [32], directed [23] and bipartite modularity [24, 26]. A bipartite network with n nodes can be conveniently denoted by a duality (p,q) (p+q=n), where p and q respectively represent the numbers of the two types of nodes. We can renumber nodes such that in the sequence $1, 2, \cdots, p, p+1, \cdots, n$, the leftmost p indices represent the first type of nodes and the remainder represent the second type of nodes. Then, Barber's bipartite modularity [26] that considers a community composed of distinct types of nodes in the network, can be written as

$$Q_b = \frac{1}{M}\sum_{i=1}^{p}\sum_{j=p+1}^{n}(A_{i,j} - \frac{k_i k_j}{M})\delta(g_i, g_j) \qquad (2)$$

Immediately, subtle difference between the two modularities in (1) and (2) can be observed. It is of interest that we will see a unipartite network can be equivalently represented as a bipartite one, and the bipartite modularity can recover the modularity for the original network. If each node i is represented by two nodes A_i and B_i, each edge $i-j$ represented by two edges $A_i - B_j$ and $A_j - B_i$, then a unipartite network with n nodes and M edges is transformed into a corresponding bipartite network with $2n$ nodes and $2M$ edges. For example, the transformation of a simple unipartite network is shown in Figs. 1. Further, if we label n nodes A_i with 1, 2, ...,n and label B_i with n+1, n+2, ..., 2n, then an edge $i-j$ in the original network corresponds to two edges, $i-(n+j)$ and $j-(n+i)$. Using the bipartite modularity introduced in Eq. (2) on the induced bipartite network, we have

$$\begin{aligned} Q_b &= \frac{1}{2M}\sum_{i=1}^{n}\sum_{j=n+1}^{2n}(\tilde{A}_{i,j} - \frac{k_i k_j}{2M})\delta(g_i, g_j) \\ &= \frac{1}{2M}\sum_{i=1}^{n}\sum_{j'=1}^{n}(\tilde{A}_{i,n+j'} - \frac{k_i k_{n+j'}}{2M})\delta(g_i, g_{n+j'}) \\ &= \frac{1}{2M}\sum_{i=1}^{n}\sum_{j'=1}^{n}(A_{i,j'} - \frac{k_i k_{j'}}{2M})\delta(g_i, g_{j'}) = Q \qquad (3) \end{aligned}$$

where we have made use of the facts the node A_i and B_i should be in an identical community and have the same

(a)

(b)

FIG. 1: The transformation of a simple unipartite network into bipartite one.(a)An unipartite network with 5 nodes and 6 edges.(b)The bipartite network corresponding to (a)

degree. Thus, bipartite modularity can be also used to community detection in unipartite networks after being transformed.

We then turn to the modularity for directed unipartite networks that are another important class of networks. The directed network can analogously be transformed to a bipartite network. A node i is represented by two nodes A_i and B_i as that in unipartite networks, while a directed edge from i to j is represented as an edge between A_i and B_j, that is, set $\{A_i\}$ and set $\{B_i\}$ are the sources and the sinks. Again, using the Eq. (2) and the facts above, we obtain

$$\begin{aligned} Q_b &= \frac{1}{M}\sum_{i=1}^{n}\sum_{j=n+1}^{2n}(\tilde{A}_{i,j}-\frac{k_i k_j}{M})\delta(g_i,g_j) \\ &= \frac{1}{M}\sum_{i=1}^{n}\sum_{j'=1}^{n}(\tilde{A}_{i,n+j'}-\frac{k_i k_{n+j'}}{M})\delta(g_i,g_{n+j'}) \\ &= \frac{1}{M}\sum_{i=1}^{n}\sum_{j'=1}^{n}(A_{i,j'}-\frac{k_i^{out}k_{j'}^{in}}{M})\delta(g_i,g_{j'}) \end{aligned} \tag{4}$$

where the right in the last equation is just the modularity for directed networks presented in [23]. The method for transforming directed networks into bipartite ones has been proposed by Guimerà et al [24], but their bipartite modularity is distinct from Barber's one as mentioned before.

Consequently, bipartite networks can be considered as a widely representative class of networks that provides a generic case for the problem of community structure detection. And the Barber's bipartite modularity can be served as a uniform objective for these methods of identifying communities that based on optimization.

III. EVOLUTIONARY METHOD FOR COMMUNITIES DETECTION

As a class of general-purpose tools to solve various hard problems, genetic algorithms have found wide application in bioinformatics, computer science, physics, engineer and other fields. They are, based on the Darvinian principle of survival of the fittest, a kind of global optimization method simulating evolutionary process of species in nature [34].

The evolutionary methods are easy to be implemented, and the process can be described as follows. They first create a stochastic initial population with predefined size wherein individuals are known as chromosomes representing a set of feasible solutions to the problem at hand, with each associated with a fitness. Then select chromosomes in proportion to corresponding fitness so that those better solutions would make themselves have multiple copies and worse ones would be discarded in the new population. Next, genetic operators such as crossover and mutation are performed according to respective specified ratios on the population. After these operations, the population of next generation has been reproduced. Iterate the above process to evolve the current population towards better offspring until the termination criteria is met.

Since the number of partitions on any given network grows at least exponentially in the number of nodes, the optimization of modularity is clearly a NP hard problem that has been given a rigid proof in [33], which has motived an array of heuristic methods including greedy agglomeration [11], simulated annealing [16], spectral relaxation [17, 18], extremal optimization [19] and mathematical programming [35]. All these methods perform point-point search, that is, transformation from one solution to a better one, which are susceptible to trapping in local optima. In contrast, genetic algorithms work with a population of solutions instead of a single solution. This implies that genetic algorithms are more robust because they perform multiple directional search that would make them effectively find better solutions.

However, for practitioners, a fundamental important problem is to choose appropriate parameters such as crossover rate and mutation rate, because they would seriously affect the performance of genetic algorithms. Further, these parameters are closely related to the studied problem, and even for the same problem, they should adjust themselves with the course of the search. In the following, we would like to introduce an adaptive genetic algorithm recently presented by Szeto and Zhang [36] and then propose a modified version suited for community structure detection.

A. Mutation only Genetic algorithm

Traditional genetic algorithms assume that genetic operators indiscriminately act on each locus constituted the chromosome, but this is always not the case. Indeed, the recent research in human's DNA [37] shows mutation rates at different loci are very different from one another. Inspired by this, Ma and Szeto [38] reported on locus oriented adaptive genetic algorithm (LOAGA) that make use of the statistical information inside the population to tune the mutation rate at individual locus. Szeto and Zhang [36] further presented a new adaptive genetic algorithm, called MOGA (mutation only genetic algorithm), which generalized their method by incorporating the information on the loci statistics with mutation operator. In MOGA, mutation was the only genetic operator, and the only required parameter is the size of population. MOGA was readdressed by Law and Szeto in [39], wherein it was extended to include crossover operator. Here, the description for MOGA is given on the basis of the later version.

The population matrix P is stacked by N_P chromosomes with length L, with its entries $P_{ij}(t)$ representing the allele at locus j of the chromosome i at time (or generation) t. The rows of this matrix are ranked according to the fitness of the chromosomes in descending order, i.e., $f(i) \geq f(k)$ for $i < k$. The columns are ranked according to the standard deviation $\sigma_t(j)$ (its definition see below) of alleles at locus j such that $\sigma_t(j) \leq \sigma_t(k)$ for $j < k$. In MOGA, the fitness cumulative probability, as an informative measure for chromosome i relative to the landscape of fitness of the whole population, was introduced and can be defined as follows:

$$C(i) = \frac{1}{N_P} \sum_{g \leq f(i)} N(g) \tag{5}$$

where $N(g)$ is the number of chromosomes whose fitness equals to g. Subsequently, the standard deviation $\sigma_t(j)$ over the allele distribution, as a useful informative measure for each locus j, is defined as

$$\sigma_t(j) = \sqrt{\frac{\sum_{i=1}^{N_P} (P_{ij}(t) - \overline{h_t(j)})^2 \times C(i)}{\sum_{i=1}^{N_P} C(i)}} \tag{6}$$

where the weighted factor $C(i)$ reflects the informative usefulness of the chromosome i, and $\overline{h_t(j)}$ is the mean of the alleles at locus j, given by

$$\overline{h_t(j)} = \frac{1}{N_P} \sum_{i=1}^{N_P} P_{ij}(t) \tag{7}$$

A locus with a smaller allele standard deviation is considered to be more informative than the other loci, vice versa. Indeed, this really makes sense in limited situations. For the initial population, the alleles at each locus j should satisfy a uniform distribution, so the standard deviation $\sigma_t(j)$ would very high while the locus present is not informative. A typical optimization problem generally allows of a few global optima, so the loci with higher structural information are liable to take fewer alleles than allowed, thereby having smaller allele standard deviations. Therefore, the loci with higher deviations prefer mutating while the other loci (informative loci) remain for guiding the evolution process.

Now, we can describe the process of MOGA. In each generation, we sweep the population matrix from top to bottom. Each row (a chromosome) is selected for mutation, with probability, $\alpha(i) = 1 - C(i)$. Together with Eq. (5), we have $\frac{1}{N_P} \leq C(i) \leq 1$, and a chromosome with higher fitness has fewer chance to be selected, vice versa. Particularly, the first chromosome that has the highest fitness will never be selected for mutation, while the last one will almost always undergo mutation for large enough N_P, if N_P normally takes a value from 50-100 as De Jong's suggestion [43], for example, $\alpha(N_P) = 1 - \frac{1}{N_P} = 0.98$ for $N_P = 50$. If the current chromosome i selected, then the number $N(i)$ of loci for mutation is prescribed with $N(i) = \alpha(i) \times L$. Thus, a selected chromosome with higher fitness has fewer loci to mutate, thereby most of the informative loci remains; while a selected chromosome with lower fitness has more less-informative loci to mutate. In practice, we can mutate the $N(i)$ leftmost loci because they are less informative relative to others according to the above arrangement of loci.

Overall, MOGA is expected to have a two-fold advantage over traditional genetic algorithms: first of all, no need to input parameters except the population size can make it more available for solving various problem; secondly, the mechanism of adaptive adjusting parameters can make it more effectively perform to obtain better solutions if it can work as expected.

B. A new measure for the informative of loci

Despite these possible advantages, MOGA cannot be directly applied on community structure detection due to the drawback that will be shown. Instead, we present a modified version of MOGA, called as MMOGA, which is especially suited for the problem of community structure detection. Genetic algorithms have been applied to this problem in [44, 45], but these applications are based on standard version (SGA).

We begin with the encoding schema of the genetic algorithm for finding communities in a bipartite network. A useful representation is the locus-based adjacency representation presented by Park and Song in [46] where used in clustering data and has been used for community detection [45]. In this encoding schema, a chromosome consists of n loci with a locus for a node in the network, and the allele at a locus j is the label of one neighbor of node j in the network. In this way, a chromosome actually will induce a graph that are often disconnected

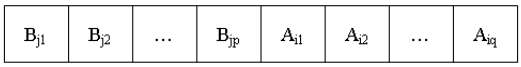

FIG. 2: An example of a chromosome for the bipartite network (p,q). B_{j_k} for $k \leq p$ represents picking a node of type B as a neighbor of the kth nodes of type A. Similarly,A_{i_k} for $k \leq q$ represents picking a node of type B as a neighbor of the kth nodes of type B.

TABLE I: Example of a population with three chromosomes. Fitness is calculated from the partition induced from decoding the chromosome. Values in each column are the alleles at the locus.

Chro.	Fitness	Loc.1	Loc.2	Loc.3	Loc.4
R1	0.5	100	20	4	8
R2	0.3	100	50	5	12
R3	0.2	10	50	6	7
σ		36.7243	15.8114	0.8165	2.0412

because of the reduction in connectivity relative to original network. Given the connectivity of the community, the clustering solution encoded by a chromosome then amounts to find all the connected components of the induced graph. For simplicity, we also call them as the connected components of the chromosome.

Now, apply the encoding schema to the case of bipartite networks. Assume that a bipartite network with n nodes consists of p nodes of type A and q nodes of type B, denoted by (p,q). For the bipartite network, we always can label nodes of type A with $1, 2, \cdots, p$ while label another type of nodes with $p+1, \cdots, n$. Then a chromosome R for the network can be represented as Fig. 2. Since our objective is to find a partition with higher modularity as possible, fitness can be defined directly in terms of modularity. Based on the above representation for the chromosome, the fitness becomes

$$f(R) = Q_b(\pi_R) = \frac{1}{M}\sum_{i=1}^{p}\sum_{j=p+1}^{n}(A_{i,j} - \frac{k_i k_j}{M})\delta(g_i, g_j) \quad (8)$$

where the parameter of Q_b is to stress the partition on which the modularity is calculated and π_R is the partition encoded by the chromosome R.

Recall that in MOGA, the allele standard deviation is used to pick the loci to mutate. When applied to community structure detection, however, the measure generally will misguide the algorithm. Consider such a simple case in which the population consists only of three chromosomes, R_1, R_2 and R_3, with fitness $f(R_1) > f(R_2) > f(R_3)$. We would like to select some of the four loci for mutation whose allele distributions are shown as Table I. From Eqs. (5) and (6), the allele standard deviations for the four loci, $\sigma_1, \sigma_2, \sigma_3$ and σ_4, can be calculated, with the result

$$\sigma_1 > \sigma_2 > \sigma_4 > \sigma_3 \quad (9)$$

. According to the selection criteria of the loci to mutate in MOGA, σ_1 has the highest standard deviation and will be picked out.

Initial population is generated randomly and when each locus follows an approximately random distribution. From the uniform distribution, we have nothing on the structure of optimal solution to the given problem. With the gradual evolution, more and more fitter members of the population will assume same alleles at some loci, which may suggest some structural information of the optimal solutions; that is, the bias (or deviation) from random distribution indicates the informativeness of the locus. In the simplest case such as knapsack problem where each locus takes 1 or 0, the allele standard deviation amounts to the bias and can work well [36].

For current case, loci 3 and 4 should be selected with equally higher priority because their allele distributions are closer to their respective random distribution. Both loci 1 and 2 appear certain bias on their alleles, indicating that they are more informative than others. If the informativeness of each chromosome taken account of, however, they are evidently different from one another. Locus 1 has a larger bias since the chromosomes with the same allele 100, i.e., R_1 and R_2 have higher fitness. In contrast, locus 2 has a smaller bias since the chromosomes with the same allele 50, i.e., R_2 and R_3 have lower fitness. Therefore, the correct order to mutate is

$$\text{locus 3=locus 4} > \text{locus 2} > \text{locus 1} \quad (10)$$

where the equal signs is meant to the pair of loci have the same priority to mutate. Obviously, the allele standard deviation would severely misguide MOGA in current case.

The failure of the allele standard deviation stems from this measure is closely related to alleles at loci. However, the information contained in loci is actually without regard to the particular values but solely determined by the bias relative to the random distribution. It is thus crucial how to measure the bias. Fortunately, we can use Kullback-Leibler divergence to describe the bias.

In the formalism of MMOGA, we explicitly represent a locus j as a discrete random variables X_j, and an allele at the locus is viewed as a value of X_j. Then the random distribution over the locus can formally given by

$$\mathcal{Q}(X_j = x) = \begin{cases} \frac{1}{|X_j|}, & \text{for each x} \in X_j \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

Let the allele distribution over the population be $\mathcal{P}$, defined by

$$\mathcal{P}(X_j = x) = \frac{\sum_{P_{ij}=x} f(i)}{\sum_i f(i)}. \quad (12)$$

We can mathematically define the bias μ as the Kullback-Leibler divergence between the two distribution, $\mathcal{P}$ and $\mathcal{Q}$:

$$\mu(j) = \sum_{x \in NBs(j)} \mathcal{P}(X_j = x) log \frac{\mathcal{P}(X_j = x)}{\mathcal{Q}(X_j = x)} \quad (13)$$

where NBs(j) represents the set of nodes adjacent to j in the original network. It is noteworthy that the quantity 0log0 should be interpreted as zero. As a Kullback-Leibler divergence, the bia is always non-negative, and is zero if and only if $\mathcal{P} = \mathcal{Q}$.

Assuming that there are three alleles at each of the four locus in the above example, we obtain $\mu_1 = 0.863$, $\mu_2 = 0.585$ and $\mu_3 = \mu_4 = 0.5145$. As a smaller bias indicates poorer information a locus contain, the locus would undergo mutation. Conversely, a larger bias means richer informativeness, the locus should remain. Therefore, guided by the bias, the order of mutation is locus 3,4,2,1 or 4,3,2,1, which completely match the order in Eq.(10).

Further, it can be observed that locus 2 has zero bias if it has only two alleles. The difference coming from the change of number of alleles would be normally concealed by the standard allele deviation. For these reasons, the bias appears superior to the standard allele deviation and is used in our MMOGA for determining the loci to mutate.

C. The reassignment technique for locus static

It is so far acknowledged that the loci with random distribution should have priority for mutation. However, this presupposition does not always hold. After the evolution of certain generations, some communities or their main-bodies would have appeared at the population scale. At present, a locus with a random distribution does not imply that it contains no information and should undergo mutation immediately. Generally, there exists in the network many nodes whose neighbors are all (or almost) in the same communities and have a similar connection pattern or even are structurally equivalent nodes [42] that are connected to the same nodes. For such a node, if all (or most) of its neighbors presenting in the same connected component predominates on the current population, then the locus has the random distribution or a approximately random distribution. Therefore, we are required to differentiate the case to avoid such misguiding.

The reassignment technique is designed to deal with this problem. For a chromosome R, the element x is the allele at the locus j which is a neighbor of the node j. Check whether the component that j is assigned to includes other neighbors with smaller labels. If it is true and the neighbor with the smallest labels is y, then the contribution from R, $\frac{f(R)}{\sum_i f(i)}$ that should be assigned to x now is reassigned to y if $x \neq y$. In this way, forward sweeping the population matrix can obtain the distribution of the locus over the population. Thus, the allele distribution over the population in Eq. (12) can be reformulated as follows:

$$\mathcal{P}^*(X_j = x) = \frac{\sum_{\mathcal{S}(i,j)=x} f(i)}{\sum_i f(i)}. \tag{14}$$

where $\mathcal{S}(i, j)$ is the node j's neighbor with the smallest label that lies as j in the same component of the chromosome i.

An example using the technique as shown in Table II. Using Eq. (12), it is obvious that the locus 1 has approximately random distribution and thus the bias approximates 0. Recalculating the distribution with the reassignment technique, however, we have $\mathcal{P}^*(X_1 = 2) = 0.53, \mathcal{P}^*(X_1 = 3) = 0.47$ and $\mathcal{P}^*(X_1 = 4) = \mathcal{P}^*(X_1 = 5) = 0$, which is very different from the random distribution with the bias 1.0026.

TABLE II: Example of reassignment technique. Column 1 is four chromosomes,column 2 is fitness of the chromosome, column 3 is the alleles of locus 1, and the right four columns show whether the corresponding nodes are in the same connected component as node 1 with 1 yes and 0 no.

Chro.	Fitness	Loc.1	Loc.2	Loc.3	Loc.4	Loc.5
R1	0.28	2	1	1	0	0
R2	0.25	3	0	1	1	0
R3	0.25	4	1	0	1	1
R4	0.22	5	0	1	0	1

The idea behind the technique is well understood. Given a locus j, we can replace its allele present with any of other alleles (i.e. other neighbors in the original network) that lie in the same component in a such way that does not alter the connectivity of the component hence causing no change in the partition encoded by the chromosome. To show the feasibility of this, we focus on the component that j lies in. Recall that a locus represents a node and the allele at the locus represent the unique neighbor which the node adhere to. Consequently, the component is in the form of a directed graph with unitary out-degree for each node . There exist two possible schemes as shown in Fig. 3 (Note that the undirected edges are irrelevant to the reassignment process thus disregard their direction).

In the scheme depicted in Fig. 3(a), we can directly change the allele from 3 to 1 but still maintain the connectivity of the component. For the schema in Fig. 3(b), however, such directly altering allele will split the original partition. To deal with this case, we study the traveling in the component along directed edges, starting from the node j. Since the subgraph elicited from node j is connected to the rest of the component through j, this traveling must end in a node that has passed. Let the path is $j \to x_1 \to x_2 \to \cdots \to x_{k-1} \to x_k$. When $x_k \neq j$, we can reestablish the connectivity by removing the last edge, reversing the direction of each edge in the path, and adding a new edge $x_1(3) \to j$. Note the resultant graph meets with the constraint that any node has only an outgoing edge. Therefore, we can reset the alleles at these locus evolved in the path. For example, in Fig. 3(b), the entire path is $j \to 3 \to 2 \to 6 \to 5 \to 7 \to 2$, so we can set the alleles according to the path, $7 \to 5 \to 6 \to 2 \to 3 \to j$,. Now, the allele at locus can be set 1. As to the case

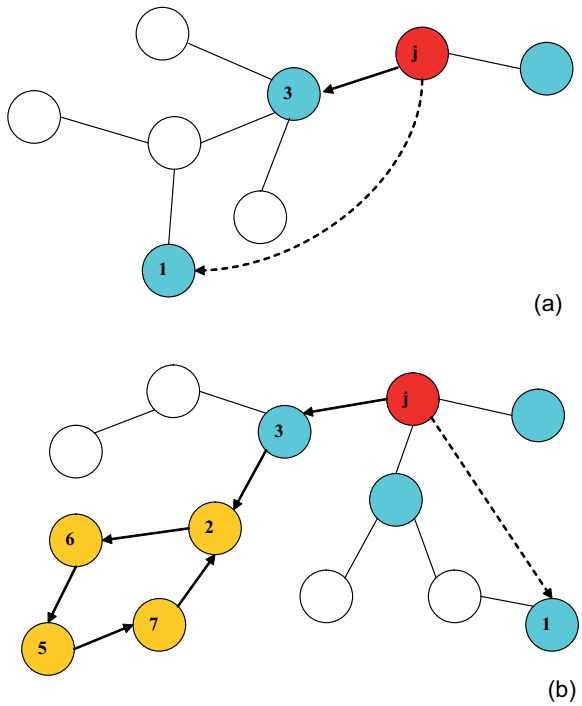

FIG. 3: Two possible schemes of changing the allele at locus j, where nodes represent loci and the directed edge $j \to i$ represent the allele present at locus j is i while undirected edges are irrelevant to the reassignment process. The red node is the node(locus) j, the blue nodes are the allele nodes in the same component, the yellow are the influenced nodes and the white are indifferent ones. (a) The new target node 1 (new allele) is in the subgraph elicited from the node 3(the allele present at locus j). (b)The new target node 1 is not in the subgraph elicited from the node 3.

$x_k = j$, we can directly alter the alleles as the schema in Fig. 3(a).

In the reassignment technique, we can also reassign the contribution from the chromosome to the allele with the maximum label that lies in the same component when performing locus statistics. More generally, it also can work as long as we arbitrarily specify a fixed reassignment order for each locus although different prescriptions may produce different bias.

Clearly, the reassignment technique is very useful for community structure detection although it would not work when applied to the loci that have a single allele, i.e., the corresponding nodes in the network are leaves. Moreover, this special case can be readily eliminated by forbidding mutation, which may bring about an unexpected merit that naturally reduces the complexity of the problem. Since most of real-world networks are scale-free networks where exists substantial leaf nodes, this merit will be significant for finding communities in these networks.

D. Population size

As in MOGA, the unique parameter in MMOGA required to be provided is the population size. The parameter may have significance influences on the application of genetic algorithms. With smaller values, the algorithms would search smaller region of solution to the problem so as to converge quickly but end with poorer solutions. While with larger values, they have more powerful search ability so as to end with better solutions but require massively computational efforts. Thus, how to select an appropriate population size for specified problems is important to practitioners.

De Jong's experiment on a small suit of test functions showed [43] the best population size was 50-100 for these functions. There are also other empirical study and theoretic analysis on this parameter [47, 48]. In practice, De Jong's setting has been widely adopted, which may be due to this choice fulfilling good tradeoff between the quality of the solution and the requirement of computation in many cases.

This popularity of the setting, however, does not exclude the development of genetic algorithms working with variable population size. A few examples of the class of algorithms can be found in [49–51]. Although one of these mechanism may be beneficial to be incorporated into MOGA*, in this work we does not take it into account.

Since we expect all alleles at a locus can simultaneous appear at the population, the population size would be preferable which is greater than the degrees of most nodes in the network. As mentioned before, most real-world networks are scale-free, so the degrees of most of nodes in these networks are less than 50. Allowing for this and the cost of large population size, we would like to take a fixed value from the interval between 50 and 200.

E. Convergence and its speeding up

MOGA and its extending to include crossover have been reported to perform well in the applications to solve knapsack problem [36] and to find the minimum energy of one dimensional Ising spin glass [39], where all the loci have two alleles, 0 and 1. For many cases, however, their performance would be hindered by two factors. One factor is the misguiding from the allele standard deviations above-mentioned. Another factor is that in the evolution of each generation the fittest individual(s) actually will not participate in the mutation unless others supersede it (them).

In fact, despite fulfilling the elite preservation strategy[40, 41] that for SGA assures the convergence toward the global optimum, MOGA dos not guarantee such convergence even it may be end with a not local optimum solution. Consider such a case that N_P-1 fittest chromosomes have identical fitness and the remaining one has lower fitness. Those fittest should be passed to next

generation while the remaining one would mutate with very high probability. If the mutation happens to reproduce a chromosome with the same fitness as others, this would unexpectedly terminate the evolutionary process.

It is helpful to notice that the fittest ones present, if not local optimum, always can perform local search to reach a local optima. Consequently, it is preferable to modify the rule for mutation so as to allow for local search. We still select the chromosome i with the same probability 1-C(f(i)) as in MOGA, if chosen then perform the mutation aforementioned, or randomly select a locus to mutate. Given that this may destroy the elite preservation strategy, MMOGA first duplicates 10% the fittest chromosomes to a new generation, then select from the current population the remaining in proportion to their fitness to prepare for mutation. These individuals temporarily generated by selection mutate according to the new rule to reproduce the 90% of the new generation. The fittest individuals are probably selected and perform local search. In this way, MMOGA not only can converge to the global optima, it also speeds up the convergence.

IV. RESULTS

In this section, we empirically study the effectiveness of MMOGA by applying it to model networks with predefined community structure and several real-world networks. In both cases, we show that the tailored adaptive genetic algorithm is superior to SGA and MOGA, and it also can compete with the famous BRIM algorithm that dedicated to bipartite network.

A. Model bipartite networks

To test how well our algorithm perform, we have applied it to the model bipartite networks with a known community structure. A model network can be constructed with two steps. First step is to determine its layout of nodes in the network, i.e., to specify the number of communities NC, and the numbers of nodes of two types included in each community N_A and N_B, as well as to assign group membership to these nodes. Next, determine the dispersion of edges by specifying the intra-community and intercommunity link probabilities, p_{in} and p_{out}, such that $p_{in} \geq p_{out}$.

For simplicity, all communities assume the same values of N_A and N_B. We set NC=5, $N_A = 12$ and $N_B = 8$ as used in [26]. One may expect that as p_{in} is markedly greater than p_{out} the networks instantiated from the model have significant community structure that tend to be detected. Conversely, as p_{out} approaches p_{in}, the network instantiations become more uniform and their modular structure becomes more obscured. In this experiment, p_{in} is fixed with the value of 0.9 while p_{out} varied by tuning p_{out}/p_{in} from 0.1 to 1 with step 0.1. We have tested on such models the performance of MMOGA as well as of SGA and MOGA, each instantiated with ten networks. On each instantiation ran these algorithms ten times.

For evaluating the quality of solutions, modularity and normalized mutual information (NMI) both are useful. But NMI is more suitable for current case since the optimal (correct) partition of the model network is known in advance. This measure takes its maximum value of 1 as the found partition perfectly matches with the known partition while it takes 0 the minimum value as they are totally independent of each other. Accordingly, we employed the stop criteria that the algorithms reach the predefined maximum generation or NMI reaches its maximum value.

Fig. 4 and Fig. 5 display the performance comparison between such genetic algorithms as $p_{out}/p_{in} = 0.1$ and $p_{out}/p_{in} = 0.2$, respectively. The maximum generation is set 2000. For both cases, MMOGA and SGA remarkably outperform MOGA. From Fig. 4 (a), we can see MMOGA is appreciably faster than SGA, as well as both perform well since the mutual information rapidly exceeds 0.9. In our test, each run of MMOGA on all 10 instantiated networks consistently gave the optimal partition, i.e., producing 100 numbers of generation less than 2000. For SGA, 97 runs gave the optimal partition. Their distributions of generations to reach the optima in Fig. 4(b) and (c) further reveal their difference in speed (in terms of generations).

As $p_{out}/p_{in} = 0.2$, the instantiated networks are, relative to the former ones, more difficult to identify their community structure. SGA succeeded in obtaining the optimal partition in 32 runs. In sharp contrast, each run of MMOGA gave the optimal partition. More information on the distributions of the generations of SGA and MMOGA are provided in Fig. 5 (a). Also in Fig. 5 (b), the variations of the mutual information with regard to SGA and MMOGA illuminate there existing a greater performance difference between them than in the case of $p_{out}/p_{in} = 0.1$.

Even as $p_{out}/p_{in} = 0.1$, MOGA reaching the optimal solution in its first 2000 generation was not observed. Actually, MOGA performed so poorly that it even did much more slowly than SGA as shown in Fig. 5 (b). The main reason for this we argue is that the use of incorrect informative measure for loci has misguided the algorithm.

We have made a more extensive performance comparison. Fig. 6 shows the variation of accuracy of MMOGA and SGA as well as BRIM against the changes of p_{out}/p_{in}. For the model networks, assigning each of the nodes from the smaller groups to its own module is a better(or may be the best) strategy for BRIM that will lead to a precise partition. To be fair [52], we picked the best partition from the ten runs on each instantiated network then average over ten instantiations for a particular p_{out}/p_{in}.

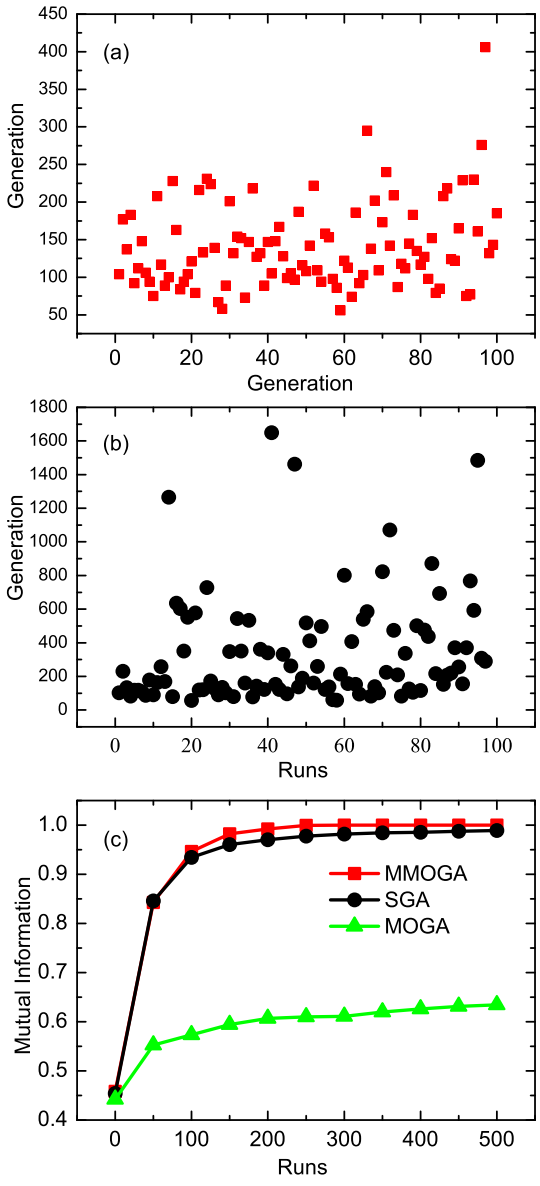

FIG. 4: Performance on bipartite model networks with $p_{in} = 0.9$, $p_{out}/p_{in} = 0.1$. The generation size is set 2000. (a)Evolutionary process of mutual information over first 500 generations. (b)Distribution of generation to reach optima using SGA. More than half of generations are over 200. (c)Distribution of generation to reach optima using MMOGA. There are 83 runs which generation to reach optima is less than 200.

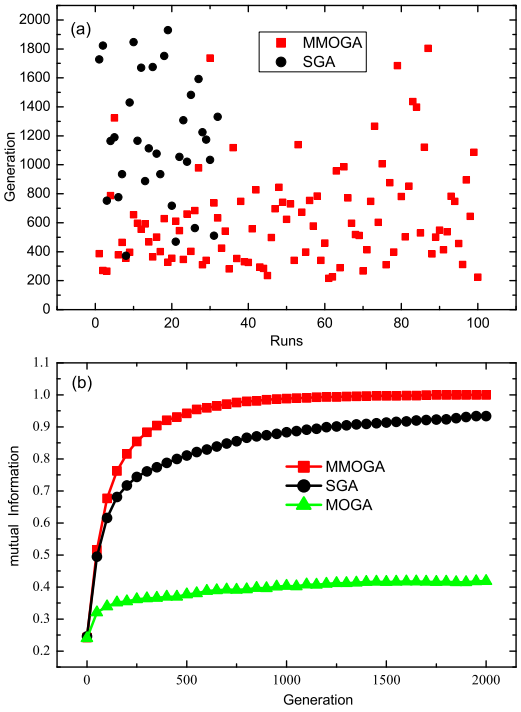

FIG. 5: Performance on model networks with $p_{in} = 0.9$ and $p_{out}/p_{in} = 0.2$. The generation size is set 2000. (a)Distributions of generation to reach optima using SGA and MMOGA. There are 32 black points and 100 red points respectively representing the generation to reach optima partition using SGA and MMOGA. Most of black points are distributed above 1000 generation while most of red points are below 800 generation. (b)Variation of mutual information with generation. Each point is the average over the 100 runs.

B. Southern women network

As the first example of real world bipartite network, we study Southern women network [53]. The social network consists of 18 women and 14 events of which the data were collected by Davis et al in the 1930s, describing their participation in these events. It has been extensively used as a typical instance for investigating the problem of finding cohesive groups hidden in social networks, see Ref. [25] for a useful review.

We have performed MMOGA ten times on this network,with population size 100 and the generation size 3000. Unlike BRIM algorithm which initial state is important, genetic algorithms generally show irrelevancy (or weaker relevancy) to initial states and can success in finding a considerable good solution. For each run, MMOGA found the best solution so far, with Q=0.34554.

Fig. 7 shows the identified community structure in the Southern women network using MMOGA. This partition is exactly the same as BRIM has found with the initial strategy that begins with assigning all events to a single community. Also we have applied SGA and MOGA to this network with the same population size and the generation size. A simple performance comparison between them is shown in Table III, which lists the success times for reaching the best solution, the minimum gener-

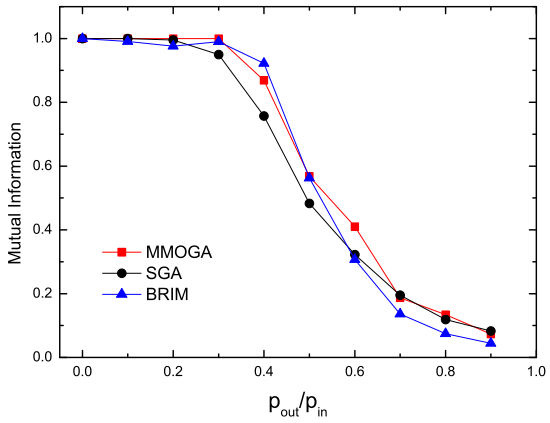

FIG. 6: Variation of performance of the algorithms with different p_{out}/p_{in}. Each point is the average over ten instantiated networks. For $p_{out}/p_{in} = 0.1, 0.2$, generation size is set 2000; for other values, generation size is set 3000.

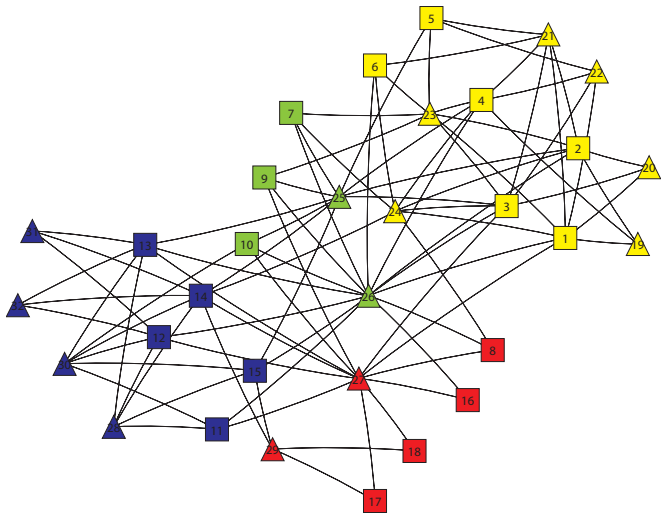

FIG. 7: Community structure in the Southern women network. Each community consists of those nodes with same color, including women and events represented by box and triangle events respectively.

ation (MinGen) and maximum generation (MaxGen) to reach the optima, as well as the average normalized mutual information (I^*norm) and average modularity (Q^*)

No matter what we are concerned about, the speed or the quality, MMOGA again takes the evident advantage over SGA and MOGA. Table IV shows the accuracy of MMOGA in comparison with other methods.

Most of previous studies dedicated to assign these women to groups for their interests. Davis et al [53] assigned the women to two groups, with 1-9 and 9-18. Women 9 can be considered as an overlapped node of the two groups in a sense, but it should be exclusively included in one group by the currently used community definition. We may label the partition with 9 and 1-8 in the same group as "Davis 1", and label the alternative partition (9 is grouped with 10-18) as "Davis 2".

Doreian et al [54] took the definition of bipartite community composed of two types nodes and proposed several partitions, with the accuracy of the partition with the highest modularity shown in Table IV. We call BRIM algorithm using strategy (1) assigning all events to a single module and (2) assigning each event to its own module as "BRIM 1" and "BRIM 2", respectively. Barber [26] reported its accuracy when using such strategies on the network, which also can be found in Table IV.

TABLE III: Performance comparison between SGA, MOGA and MMOGA. Each algorithm runs 10 times. Here, success means that the algorithms have found the best solution before the maximum generation.

Method	Succ.	MinGen.	MaxGen.	I^*_{norm}	Q^*
SGA	4	2011	2924	0.8923	0.34539
MOGA	0			0.7997	0.34477
MMOGA	10	87	1830	1	0.34554

TABLE IV: Performance comparison on the Southern women network, where partial data are drawn from [26].

Method	Communities	Q^*	I^*_{norm}
MMOGA	5	0.34554	1
BRIM 1	5	0.34554	1
BRIM 2	2	0.32117	0.58032
Davis 1	2	0.31057	0.44657
Davis 2	2	0.31839	0.45126
Doreian	3	0.29390	0.60766

C. Scotland corporate interlock network

The second real world bipartite network we have tested on is Scotland corporate interlock network [55]. This network describes the corporate interlock pattern between 136 directors and 108 largest joint stock companies during 1904-1905. As it is disconnected, we focus merely on its largest component which comprises 131 directors and 86 firms. And in the following, the network consistently indicates this component.

BRIM algorithm found the poorer partitions of this network with Q=0.56634 and Q=0.39873, using the strategies assigning all directors to unique modules or to the same module. With the adaptive binary search technique, BRIM algorithm, when using the strategy of randomly assigning directors to modules, may find a remarkable better solution with Q=0.663($\pm$0.002). Based on the experimental results, the author suggested the network comprise approximately 20 communities.

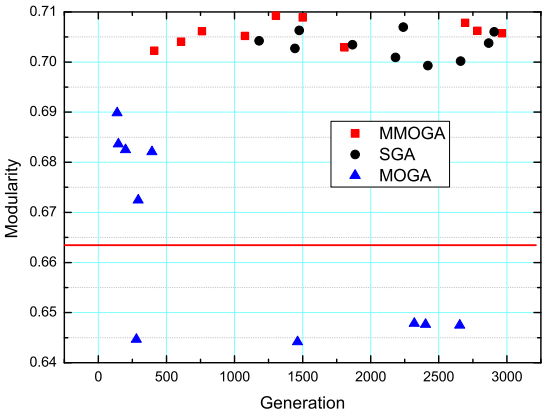

FIG. 8: Distributions of the partitions returned by SGA, MOGA and MMOGA.

Similarly, we have examined the performance of such three algorithms on this network by each running ten times with the same setting as before. Figure 8 shows the distributions of the solutions returned by SGA, MOGA and MMOGA. Obviously, both SGA and MMOGA definitely exhibit higher accuracy than BRIM and MOGA.

Moreover, MMOGA appears more preferable to SGA. In the experiment, the modularity of the best partition found by SGA, Q =0.70699, is less than those of best two partitions (π_1 and π_2) found by MMOGA with Q=0.709275 and Q=0.708887. On the other hand, as shown in Fig. 8, for MMOGA most of ten partitions including the best two are found during the first 2000 generations while for SGA six of ten partitions are found after 2000 generations.

In closing, we would like to give a simple evaluation of the reliability of the solutions. We calculated the normalized mutation information between any pairs of the solutions returned by MMOGA. The maximum value is the NMI between π_1 and π_2, which is equals to 0.91913 indicating that they are very similar. Simultaneously, for each solution, we calculated the mean of the NMI between the partition with other partitions. We found π_2 has the largest value 0.84586 and π_1 has the third largest value 0.82483. These facts lend confidence to the reliability to the optimum partitions obtained, π_1 and π_2. Figure 9 shows the community structure of this network according to π_2. Clearly, MMOGA indeed has given a very accurate partition of this network.

V. CONCLUSION

We have shown both unipartite networks and directed networks can be equivalently represented as bipartite networks, and their modularity are just the corresponding bipartite modularity. This means that bipartite networks can be considered as a more extensive class of networks that includes unipartite networks and directed networks. Therefore, methods for detecting community structure of bipartite networks generally can be applied to unipartite networks and directed networks. Here, we have presented an adaptive genetic algorithm, MMOGA, for the task of community structure detection. Although it is addressed in the case of bipartite networks, it obviously can hopefully succeed in the application to unipartite networks and directed networks.

This algorithm is based on the frame of MOGA that was presented with the aim to improve performance of traditional genetic algorithms. But we have shown that MOGA has a poor performance as applied to this task. In fact, we have shown MOGA does not guarantee the convergence to global optima. In MMOGA, we introduce the new measure for the informativeness of loci, the modified rule for mutation and the reassignment technique. These ingredients jointly make MMOGA more effectively optimize objective function for community structure detection, fulfilling the motive of MOGA. The experiments on model networks and real networks consistently show MMOGA outperforms MOGA, SGA and BRIM. In contrast with BRIM, another advantage is that MMOGA can automatically determine the number of communities.

A potential advantage of the evolutionary method is that it can return multiple better solutions with no bias, which can provide some useful information on the reliability of the solutions that we are interested. We look forward to the technique that can effective combine the solutions to obtain a more accurate partition. Finally, we believe that as an effective discrete optimization method (the special reassignment technique can be switch off as needed) it will find more application in many fields.

Acknowledgments

This research was supported by the National Basic Research Program of China under grant No. 2007CB310806, the National Natural Science Foundation of China under Grant Nos. 60704044, 60873040 and 60873070, Shanghai Leading Academic Discipline Project No. B114, and the Program for New Century Excellent Talents in University of China (NCET-06-0376).

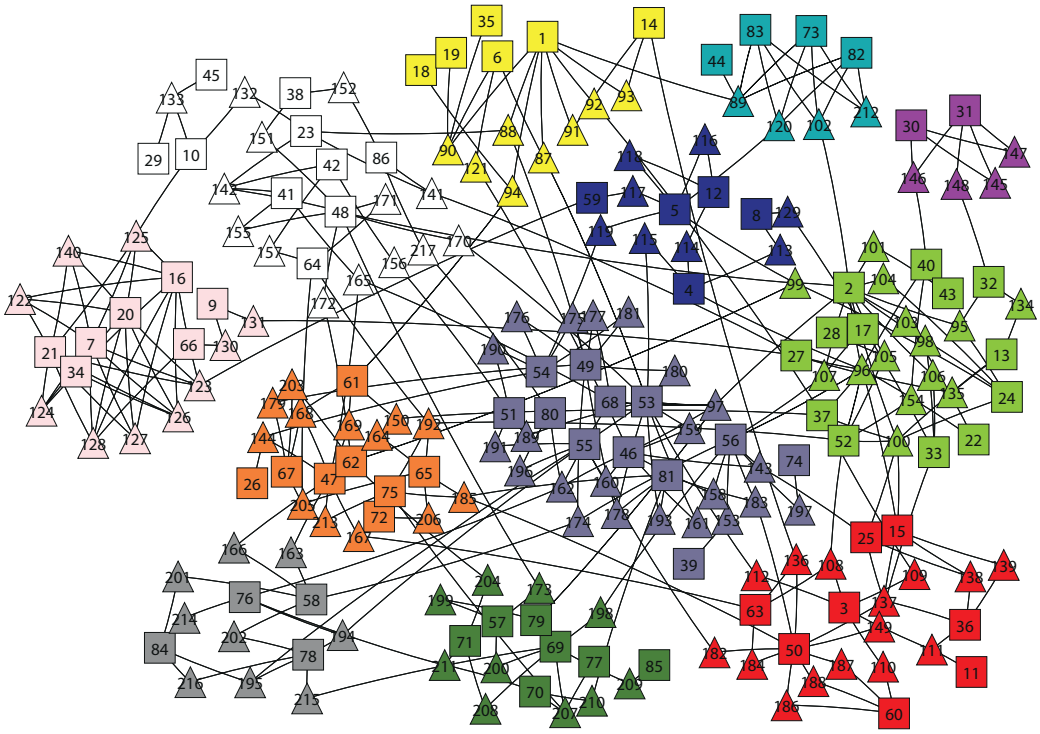

FIG. 9: Community structure of the Scotland corporate interlock network. Each community consists of those nodes with same color, including firms and directors represented by box and triangle events respectively.

[1] D. J. Watts and S. H. Strogatz, Nature (London), **393**, 440 (1998).
[2] A.-L. Barabási and R. Albert, Science **286**, 509 (1999).
[3] R. Albert and A.-L. Barabási, Rev. Mod. Phys. **74**, 47 (2002).
[4] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of Networks: From Biological Nets to the Internet and WWW* (Oxford University Press, New York, 2003).
[5] M. E. J. Newman, SIAM Rev. **45**, 167 (2003).
[6] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwanga, Phy. Rep. **424**, 175 (2006).
[7] N. Neubauer and K. Obermayer, in *Proceedings of the NIPS Workshop on Analyzing networks and learning in graphs*, 2009.
[8] T. Murata, in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 1159-1160.
[9] M. Girvan and M. E. J. Newman, Proc. Nat. Acad. Sci. U.S.A. **99**, 7821 (2002).
[10] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).
[11] M. E. J. Newman, Phys. Rev. E **69**, 066133 (2004).
[12] A. Vazquez, A. Flammini, A. Maritan and A. Vespignani, Nature Biot. **21**, 6 (2003).
[13] S. Gupta, R. M. Anderson, and R. M. May, AIDS **3**,807 (1989).
[14] G. Yan, Z. Q. Fu, J. Ren, W.-X. Wang, Phys. Rev. E **75**, 016108 (2007).
[15] A. Arenas, A. Díaz-Guilera, and C. J. Pérez-Vicente, Phys. Rev. Lett. **96**, 114102 (2006).
[16] R. Guimerà and L. A. N. Amaral, Nature (London) **433**, 895 (2005).
[17] M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **103**, 8577 (2006).
[18] M. E. J. Newman, Phys. Rev. E **74**, 036104 (2006).
[19] J. Duch and A. Arenas, Phys. Rev. E **72**, 027104 (2005).
[20] H. Zhou, Phys. Rev. E **67**, 041908 (2003).
[21] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, Proc. Nat. Acad. Sci. U.S.A. **101**, 2658 (2004).
[22] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Nature (London) **435**, 814 (2005).
[23] E. A. Leicht and M. E. J. Newman, Phys. Rev. Lett. **100**, 118703 (2008).
[24] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, Phys. Rev. E **76**, 036102 (2007).
[25] L. C. Freeman, in Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers, edited by R. Breiger, C. Carley, and P. Pattison (The National Academies Press, Washington, DC, 2003), pp. 39-97.
[26] M. J. Barbe, Phys. Rev. E **76**, 066102 (2007).
[27] L. Danon, A. Díaz-Guilera, J. Duch and A. Arenas, J. Stat. Mech. (2005) P09008.
[28] S. Fortunato, Phys. Rep. **486**, 75 (2010).
[29] S. Fortunato and M. Barthelemy, Proc. Nat. Acad. Sci. U.S.A. **104**, 36 (2007).
[30] J. M. Kumpula, J. Saramaki, K. Kaski, and J. Kertesz, Eur. Phys. J. B **56**, 41 (2007).
[31] B. H. Good, Y.-A. Montjoye and A. Clauset, Phys. Rev. E **81**, 046106(2010).
[32] A. Arenas, J. Duch, A. Fernández, and S. Gómez, New J. Phys. **9**, 176 (2007).

[33] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner, IEEE Tran. on Know. and Data Eng. **20**, 2 (2008).

[34] J. H. Holland, Adaptation in natural and artificial systems (University of Michigan Press, 1975)

[35] G. Agarwal, D. Kempe, Eur. Phys. J. B, **66**, 409 (2008).

[36] K. Y. Szeto and J. Zhang, in *Lecture Notes in Computer Science*, edited by I. Lirkov, S. Margenov, and J. Wasniewski (Springer-Verlag, 2006), Vol. 3743, pp. 189-196.

[37] B. Brinkmann et al, Am. J. Hum. Genet **62**, 1408 (1998).

[38] C. W. Ma and K. Y. Szeto, in *Proceedings of the International Conference on Recent Advances in Soft Computing, Nottingham, 2004*, edited by L. Lofti (Springer-Verlag, 2004), pp.410-415.

[39] N. L. Law and K. Y. Szeto, in *Proceeding of the International Joint Conference on Artifical Intelligence, Hyderabad, 2007*, edited by Manuela Veloso (AAAI Press, 2007), pp. 2330-2334.

[40] J. c. Bean, ORSA J. Comput. **6**, 154(1994).

[41] From the discussion in III A, MOGA will duplicate the fittest individual(s) to a next generation.

[42] F. Lorrain and H. White, J. Math. Sociol. **1**, 49 (1971).

[43] K. A. De Jong, Ph. D. thesis, University of Michigan, 1975.

[44] M. Tasgin and H. Bingol, arXiv:0711.0491v1.

[45] C. Pizzuti, in *Lecture Notes in Computer Sciences*, edited by G. Rudolph et al (Springer-Verlag, 2008), Vol. 5199, pp. 1081-1090.

[46] Y. Park, M. Song, in *Proceedings of the Annual Conference on Genetic Programming, Wisconsin, 1998*, edited by J. R. Koza et al (Morgan Kaufmann, 1998), pp. 568-575.

[47] J. J. Grefenstette, IEEE T on Systems, Man, Cybernetics **16**, 1 (1986).

[48] D. E. Goldberg, in *Proceedings of the 3rd International Conference on Genetic Algorithms*, edited by J. Schaffer (Morgan Kaufmann Publishers, Los Altos, CA, 1989), pp. 70-79.

[49] J. Arabas, Z. Michalewicz, J. Mulawka, in *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, edited by D.B. Fogel (IEEE Press, Los Altos, 1994), pp. 73-78.

[50] M. Affenzeller, S. Wagner, and S. Winkler, in *Lecture Notes in Computer Science*, edited by R. Moreno-Díaz, F. Pichler and A. Quesada-Arencibia (Springer-Verlag, Berlin, Heidelberg 2007), Vol. 3749, pp. 820-828.

[51] T. Hu, S. Harding and W. Banzhaf, Genetic Programming and Evolvable Machines, **11** 2 (2010).

[52] Picking the best one of the run on each instantiations is reasonable because the population size are moderate so that it can be increased and hence improving the accuracy of each run; that is, we adopted a better strategy as BRIM do.

[53] A. Davis, B. B. Gardner, and M. R. Gardner, *Deep South* (University of Chicago Press, Chicago, 1941).

[54] P. Doreian, V. Batagelj and A. Ferligoj, Soc. Networks **26**, 29 (2004).

[55] J. Scott and M. Hughes, *The Anatomy of Scottish Capital: Scottish Companies and Scottish Capital,1900-1979* (Croom Helm, London, 1980).

[56] A. Clauset, M. E. J. Newman and C. Moore, Phys. Rev. E **70**, 066111 (2004).