Physica A 392 (2013) 1215–1231

Contents lists available at SciVerse ScienceDirect

Physica A

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

Community detection based on modularity and an improved genetic algorithm

Ronghua Shang*, Jing Bai, Licheng Jiao, Chao Jin

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

ARTICLE INFO

Article history:
Received 18 April 2012
Received in revised form 8 July 2012
Available online 16 November 2012

Keywords:
Complex network
Community detection
Prior information
Genetic algorithm

ABSTRACT

Complex networks are widely applied in every aspect of human society, and community detection is a research hotspot in complex networks. Many algorithms use modularity as the objective function, which can simplify the algorithm. In this paper, a community detection method based on modularity and an improved genetic algorithm (MIGA) is put forward. MIGA takes the modularity Q as the objective function, which can simplify the algorithm, and uses prior information (the number of community structures), which makes the algorithm more targeted and improves the stability and accuracy of community detection. Meanwhile, MIGA takes the simulated annealing method as the local search method, which can improve the ability of local search by adjusting the parameters. Compared with the state-of-art algorithms, simulation results on computer-generated and four real-world networks reflect the effectiveness of MIGA.

1. Introduction

With the development of computer technology, complex networks are widely applied in all walks of life. Community detection is one of the hot topics in the study of complex networks, and the notes of different properties and different types constitute the structure called community in the network. Communities are groups of nodes having dense intra-connections and sparse inter-connections, which is one of the characteristics of complex network [1]. Community detection is important for understanding network structures and analyzing network characteristics. Community structure analysis is widely used in biology, physics, computer graphics and sociology.

In order to accurately analyze the community structure in networks, many excellent community detection methods have been put forward. Pothen, Simon and Liou proposed the spectral bisection method, which is a method based on hierarchical clustering to detect community structure in networks [2]. Newman and Girvan proposed the GN algorithm, which is a kind of split method [3]. Newman put forward the modularity Q [4] and proposed a fast algorithm based on the GN algorithm, which is a kind of condensing algorithm. Newman also proposed a spectrum algorithm based on the concept of the modularity matrix [5]. Leicht and Newman extended the spectrum algorithm based on the modularity matrix to the directed network [6]. Rosvall and Bergstrom proposed a community detection method based on information theory. It is a method which takes the modularity of the network as a lossy compression of the network structure, and then the problem of community detection is converted into a foundation problem in information theory [7]. Raghavan, Albert, and Kumara put forward the LPA algorithm [8]. Palla, Dereny and Farkas proposed a group penetration algorithm, which is the first algorithm that can detect overlapping communities [9]. On that basis, Kumpula, Kivea and Kaski proposed the SCP algorithm [10] and Duch and Arenas put forward the EO algorithm [11]. Via an improved spectral method, Xie et al. proposed the detection of community structure in a network, which proved to be successful in clustering nodes [12]. For

* Corresponding author.
E-mail address: rhshang@mail.xidian.edu.cn (R. Shang).

0378-4371/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.physa.2012.11.003

huge real-world networks, Wu et al. proposed an efficient overlapping community detection method, which is better than other algorithms both in the quality of results and in computational performance [13]. Zhang et al. proposed a fuzzy analysis of community detection in complex networks, which can quickly produce the desired results [14]. Faqeeh et al. proposed a community detection method based on the "clumpiness" matrix, which gives accurate results for many computer-generated and real-world networks [15]. Pan et al. proposed a detecting community structure via node similarity, which is rather efficient in discovering the community structure in complex networks [16].

In recent years, researchers have gradually tended to use artificial intelligence technology to improve the accuracy of community detection. One of the fastest development methods in recent years is to optimize the modularity to find an ideal community structure [17,18]. One of the most widely used methods is the genetic algorithm (GA).

The GA is a famous adaptive heuristic search algorithm inspired by the evolutionary ideas of natural selection and genetics and pioneered by Holland at the University of Michigan [19]. The GA is designed to simulate processes in natural system necessary for evolution and to follow the principles first laid down by Charles Darwin of survival of the fittest. When a GA is used to solve a problem, first, a population of chromosomes or individuals should be maintained where each chromosome represents a potential solution to the problem. Second, each chromosome is evaluated to give some measure of its fitness. And then some chromosomes undergo stochastic transformations by means of genetic operators (crossover and mutation) to form new chromosomes called offspring, where crossover creates new chromosomes by combining parts from two chromosomes and mutation creates new chromosomes by making changes in a single chromosome. Then, a new population is formed by selecting the more fit chromosomes from the parent population and the offspring population. Further iterations are carried out until the stopping criterion is satisfied [20].

The GA has been widely studied and applied in many fields in the engineering world. Goldberg studied GAs in search, optimization and machine learning [20]. Koza proposed Genetic Programming, and has published two books [21]. McKinney proposed groundwater management models and obtained the solutions by using a GA [22]. Schwefel proposed Evolution Strategies for Evolution and Optimum Seeking [23]. Reeves used a GA to solve flowshop sequencing [24]. Yang used a GA to select such subsets and to achieve multicriteria optimization with the features [25]. Jones et al. used a genetic algorithm for flexible docking [26]. Deb et al. successfully applied a GA to multiobjective optimization problems [27]. Stoico et al. designed a genetic algorithm for the 1D electron gas [28]. Wu et al. modeled interaction networks in genetic algorithms and analyzed the scale-free properties of information flux networks [29]. In recent years, many researchers have proposed algorithms [30,31] based on the GA for community detection. The existing algorithms based on the GA for community detection have some advantages such as parallel search and some drawbacks such as slow convergence and low accuracy. Theory and experiments have shown that a GA can find the local optimal solution and can hardly find the global optimal solution.

Gong et al. have proposed a memetic algorithm (MA) for community detection in networks based on the GA [32]. In MAs, a meme is defined as the learning process capable of performing local refinements for an individual. MAs are named differently such as genetic local searchers and Lamarckian genetic algorithms. An MA for community detection [32] has been proposed to optimize the modularity density. The modularity density includes a tunable parameter. The algorithm uses a hill-climbing strategy for the local search procedure. From the optimization point of view, the MA has greatly improved the detection effect of the GA and has improved the accuracy rate of community detection. However, the modularity density D brings in a parameter λ [33]. Only by adjusting λ can one find relatively ideal detection results [32]. Meanwhile, the hill-climbing strategy is a simple greed research algorithm, which selects the optimal solution as the current solution from the adjoining solution space of the current solution each time. The algorithm will end when a local optimal solution is found. The hill-climbing strategy can be easily realized. Its main disadvantage is low efficiency, local optimum and not always searching the global optimum. The simulated annealing method can move around with a probability, namely the method will accept a worse solution than the current one with a probability. The algorithm will perhaps jump away from the local optimum and get the global optimum after moving several times. So this paper takes the simulated annealing method as the local search method.

In order to improve the precision ratio for community detection and the ability of handling community detection problems, this paper proposes a community detection method based on the modularity and an improved genetic algorithm (MIGA). First, the calculation of the modularity Q is fast and simple, which can well guide the performance of the results for community detection. So MIGA takes the modularity Q as the objective function which can simplify the algorithm. Second, MIGA uses prior information (the number of community structures) in the initialization, which makes the algorithm more targeted and improves the stability and accuracy of community detection. On the one hand, the number of community structures is known for most real-world networks. The algorithm can immediately use prior information. On the other hand, for some networks whose classes are unknown, we can get the general classes of the network by using the traditional GA or the other state-of-art algorithms. Then, we select the numbers around the number of obtained general classes. Last, because the simulated annealing method has the advantages of good local searching ability of the network and lower computational complexity, MIGA takes the simulated annealing method as the local search method which can improve the ability of local search by adjusting the parameters. MIGA detects some classic networks, such as computer-generated networks, Zachary's karate club [34], the dolphin social network [35], American college football [3] and books about US politics [5,36]. Simulation results show that MIGA can greatly improve the stability of the detection results and the accuracy of the final results.

The structure of this paper is as follows. Section 2 expounds the content of community detection based on the modularity and an improved genetic algorithm. Section 3 presents the simulation experiments and results analysis. Some classic networks are detected by MIGA, the MA, and the GA, respectively, and experimental results are analyzed. Section 4 presents the summary.

2. Proposed method for community detection

For the problem proposed by the first section of community detection, the algorithm for community detection is proposed based on the modularity and an improved genetic algorithm. In the literature [32], based on the GA, the MA takes the modularity density D as the objective function. The MA uses the hill-climbing strategy for local searching. The strategy can enhance the ability of local search and help the algorithm find a better chromosome. However, in the MA, in order to find a better chromosome the objective function D must adjust λ, which leads to large amount of calculation, and along with the community is randomly detected into several classes, which leads to the difference between the detection results and practical results; with the MA it is difficult to find the optimal solution when the hill-climbing strategy handles large-scale problems. In order to overcome these problems, on the basis of the MA, we propose a new algorithm called MIGA. First, MIGA takes the modularity Q as the objective function. Second, MIGA uses prior information: the number of community structures. Finally, our algorithm takes the simulated annealing method as the local search method. Comparing MIGA with the MA and the GA, by analyzing, MIGA is more targeted and has lower computational complexity in both computer-generated networks and real-world networks. On the whole, MIGA improves the stability and accuracy of community detection.

The following sections describe the representation and initialization, modularity Q [4], the genetic operator (the crossover operator and the mutation operator) and the local search.

2.1. Representation and initialization

In the initialization, a chromosome is encoded as an integer string. A chromosome in the population can be expressed as follows [32]:

$$\boldsymbol{r}_m = [r_m^1\, r_m^2 \cdots r_m^i \cdots r_m^N]. \tag{1}$$

Here, the vector $\boldsymbol{r}_m$ represents the m-th chromosome in the population and r_m^i represents which class the i-th node belongs to. All the nodes in the chromosome are positive integers and N indicates the number of nodes. Here, it should be noted that nodes which belong to the same class mean that they are in the same community; otherwise, the nodes are in different communities.

After the encoding method is determined, the algorithm performs the population initialization, in which step MIGA uses prior information: the number of community structures based on the given classes of the network. The class labels of nodes in each chromosome are randomly designated within the scope of the given classes. This will make the community detection more accurate and improve the performance of the algorithm.

2.2. Modularity Q

In the literature [32], the modularity density D brings in a parameter λ, and only by adjusting λ can one find relatively ideal detection results. In order to reduce the complexity and improve the precision of community detection, the modularity Q is taken to replace D as the objective function in this paper. The modularity Q is defined as follows [4]:

$$Q = 1/2M \sum_{ij} \left(a_{ij} - \frac{k_i k_j}{2M} \right) \delta(i, j), \tag{2}$$

where M represents the total number of the edges in the network. i and j are two nodes in the network. k_i and k_j are the degree of the i-th node and the degree of the j-th node. a_{ij} is the element in the i-th row and the j-th column in the adjacent matrix. $\delta(i, j)$ represents the relationship between the i-th node and the j-th node. If node i and node j are in the same community, $\delta(i, j) = 1$; otherwise, $\delta(i, j) = 0$.

2.3. Genetic operators

Genetic operators are used to alter the genetic composition of chromosomes during representation. Crossover and mutation are two common genetic operators [20]. Following are the details of the two genetic operators.

2.3.1. Crossover operator

Crossover operates on two chromosomes at one time and creates new chromosomes by combining parts from two chromosomes. So the offspring generated by the crossover operator combine the features of both of the chromosomes [20]. Traditionally, the process of the crossover operator generally includes three steps: (1) it selects two chromosomes, (2) takes them together, and (3) outputs two new chromosomes. A crossover point is selected in a chromosome, and then the two chromosomes' elements after that selection point are exchanged. The traditional GA based algorithms include different crossover operators such as the one-way crossover, two-way crossover, flat crossover, blend crossover, and so on. In the proposed algorithm, the crossover operator takes the two-way crossing introduced in Ref. [32]. The two-way crossing is based on one-way crossing over introduced in Ref. [37]. Two-way crossing can increase the diversity of chromosomes and expand the optimal space. More details of two-way crossing are as follows.

Table 1
Two-way crossing.

v	$\mathbf{X}_a$	$\mathbf{X}_b$	$\mathbf{X}_c$	$\mathbf{X}_d$	$\mathbf{X}_a$	$\mathbf{X}_b$	v
1	④	→ 2	→ ④	4	4	2	1
2	3	6	6	⑥	← 3	← ⑥	2
③	→ ④	→ 6	→ ④	⑥	← 4	← ⑥	← ③
4	5	4	4	5	5	4	4
5	2	6	6	⑥	← 2	← ⑥	5
6	④	→ 2	→ ④	4	4	2	6

Table 2
Mutation operation.

v	$\mathbf{X}$	$\mathbf{X}'$
1	2	2
2	1	1
3	→ ②	→ ①
4	2	2
5	1	1
6	1	1

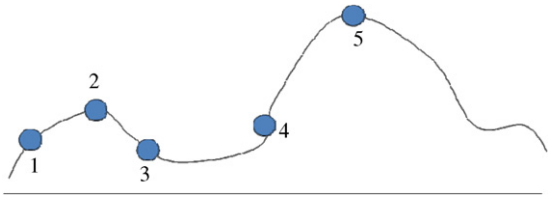

Fig. 1. Illustration of local search.

Two chromosomes, called respectively $\mathbf{X}_a$ and $\mathbf{X}_b$, are randomly selected, where $\mathbf{X}_a$ is the source chromosome and $\mathbf{X}_b$ is the target chromosome. First, a vertex v_i is picked randomly in $\mathbf{X}_a$. Second, which class v_i belongs to is determined in chromosome $\mathbf{X}_a$. Finally, all the vertices with this class of $\mathbf{X}_a$ are also assigned to the same class label in chromosome $\mathbf{X}_b$. Meanwhile, we take $\mathbf{X}_b$ as source chromosome and $\mathbf{X}_a$ as target chromosome, and then try to repeat the steps described above. The above process is called a two-way crossover operator, which is shown in Table 1 [32].

2.3.2. Mutation operator

The mutation operator is the main operator in the GA; it produces spontaneously random changes in various chromosomes [1]. So the mutation operator can increase the diversity of the population and speed up the convergence. The mutation operator is to make the genic values in some certain genic positions of chromosomes inverse with a certain probability. The GA based algorithms include different crossover operators such as boundary mutation, plain mutation, nonuniform mutation, directional mutation, Gaussian mutation, and so on [20].

In the proposed algorithm, the process of the mutation operator is as follows: a vertex is picked randomly and the class label is assigned randomly. This mutation operation can reduce the search space and simplify the mutation process, which is shown in Table 2. As shown in Table 2, taking a two-class network as an example, a chromosome $\mathbf{X}$ is selected first; then a vertex in the chromosome is picked randomly. Take vertex 3 as an example. The class label of vertex 3 is 2 before mutating. Then, after the mutation operation, the class label of vertex 3 becomes one of the other class labels (here the class label is 1). This can promote the diversity of chromosomes, and there is more chance to find a Q with larger value.

2.4. Local search procedure

Both the hill-climbing strategy and the simulated annealing method are local search methods. In this paper, the simulated annealing method is used instead of the hill-climbing strategy [20]. The reasons are stated as follows.

The hill-climbing strategy can be easily realized. Its main disadvantage is low efficiency, local optimum and not always searching the global optimum [32]. As shown in Fig. 1, assuming point 1 is the current optimal solution, the hill-climbing strategy will stop when the local optimal solution point 2 is searched. The main reason for this is that the strategy could not find a better solution no matter where it moved at small amplitude in point 2.

However, the simulated annealing method which comes from the solid annealing principle can move around with a probability, namely the method will accept a worse solution than the current one with a probability. The algorithm will perhaps jump away from the local optimum and get the global optimum after moving several times [38]. As one can see in Fig. 1, the simulated annealing method will move to point 3 with a probability when it finds point 2 as the local optimal

Table 3
Local search procedure.

Algorithm: Local search procedure
Step 1: Input T, k, l, chromosome g_1, g_2; $i := 1$; Step 2: Chromosomes $G1$, $G2 \leftarrow$ Crossover operation(g_1, g_2); Step 3: $Q(G1)$, $Q(G2) \leftarrow$ Objective function Q of $G1$ and $G2$; $p = Q(G1) - Q(G2)$; Step 4: $r = \text{rand}(1, 1)$; If $p > 0$ or $\exp(-p)/T > r$, return to step 5; otherwise: $i := i + 1$, return to step 2; Step 5: $i := i + 1$; Update the chromosomes, that is let $g_1 = G1$; Step 6: If $i < l$, return to step 2; otherwise, go to step 7; Step 7: $T = k^*T$, If $T < 0$, output results; otherwise, return to step 2.

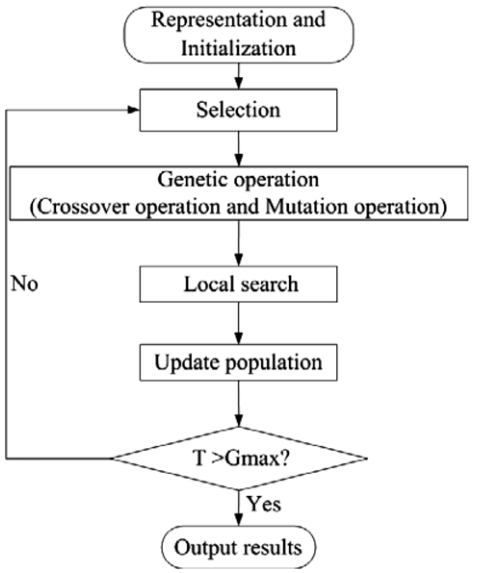

Fig. 2. The flow chart of MIGA.

solution. Maybe it arrives at point 4 after moving several times, which will jump away from point 2 of the local optimal solution. The method will finally find point 5 of the global optimum with a probability. So we adapt the simulated annealing method as the local search method.

The simulated annealing method comes from the solid annealing principle in metallurgy [38]. This technique involves heating the solid, making sure that the temperature in the solid is high enough, and then controlling the cooling of the material to increase the size of its crystals and reduce the number of defects. When the solid is heated, the atoms become unstuck from their initial positions, which is a local minimum of the internal energy. When the temperature decreases slowly, the atoms become ordered and will get more chances of finding configurations with lower internal energy than the initial one. Inspired by the above process, the simulated annealing algorithm can find a better chromosome [38].

According to the idea of the combination of the GA and simulated annealing method [38–40], in this paper, the simulated annealing method is taken as the local search method after the genetic operation. This method can overcome the drawbacks of poor local searching ability in a complicated network and help the algorithm find the real community structure. The main steps of simulated annealing operation are the following.: (1) Initialize some parameters including the initial temperature T (large enough), the cycle number l in each T, the iteration number $i = 1$ and the constant k ranging from 0 to 1, and then select two chromosomes g_1 and g_2 which have the two largest values of modularity Q. The definition and the calculation of modularity Q are given in Section 2.2. (2) Two new chromosomes marked $G1$ and $G2$ are obtained through performing the crossover operator with g_1 and g_2 and then the new chromosomes $G1$ and $G2$ are output. Meanwhile, calculate the modularity Q of $G1$, $G2$, denoted by $Q(G1)$ and $Q(G2)$, and then get $p = Q(G1) - Q(G2)$. (3) Generate a number $r \in [0, 1]$ randomly. When $p > 0$ or $\exp(-p)/T > r$, output $G1$ and let g_1 equal to $G1$; otherwise, $i = i + 1$; return to step (2). (4) $i = i + 1$; if i is less than l, return to step (2). Otherwise, perform the next step. (5) $T = k^*T$; output the results when $T < 0$; otherwise return to step (2). The step-by-step flow chart of the simulated annealing method is shown in Table 3.

2.5. The flow chart of MIGA

Based on the descriptions above, the step-by-step flow chart of MIGA is shown in Fig. 2.

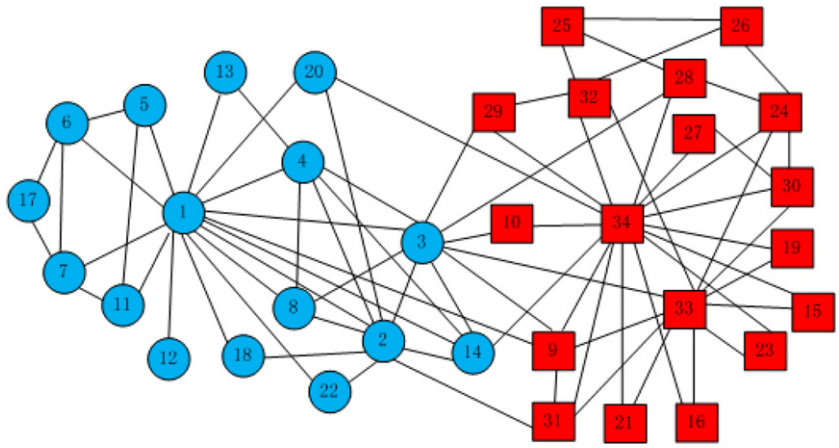

Fig. 3. Zachary's karate club.

3. Simulation experiments and results analysis

3.1. Test problems

In this paper, five networks are used to evaluate the performance of the algorithms. They are a computer-generated network and four real-world networks: Zachary's karate club, the dolphin social network, American college football and books about US politics.

3.1.1. Computer-generated network [41]

This network is a benchmark network put forward by Lancichinetti et al. 128 nodes are divided into four communities and there are 32 nodes in each community. The network brings in a mixing parameter μ, and by adjusting μ one can analyze the performance of algorithm. More details are given in the literature [41].

3.1.2. Four real-world networks

(1) Zachary's karate club [34].

Zachary's karate club is one of most the widely used networks in community detection. The 34 members of the club constitute the 34 nodes of the network. The relationships between members constitute the 78 edges of the network. The network is shown in Fig. 3.

(2) Dolphin social network [35].

By the observation of dolphin behavior for seven years, Lusseau proposed the network. The connection of any two dolphins represents that they have tighter connection. The dolphin social network consists of 62 dolphins as the nodes and 159 connections as the edges in the network. The network can be detected as two communities, as shown in Fig. 4.

(3) American college football [18].

The network was proposed by Girvan and Newman. The nodes represent different football teams and the edges represent the matches between two teams. The network consists of 115 nodes and 616 edges. The network consists of 12 communities, which are 12 football teams. The network is shown in Fig. 5.

(4) Books about US politics [5,36].

This network consists of 105 books about US politics published in 2004, and sold by Amazon.com. Based on the descriptions and reviews of the books posted on Amazon [38], Newman divided the network into three communities. The network is shown in Fig. 6.

3.2. Performance measure

Normalized mutual information (NMI) is taken as the performance measure. NMI reflects the similarity between the true community structure and the detected community structure. NMI is used to estimate the performance of MIGA, the MA and the GA. Given two parts A and B of a network, C is the confusion matrix. In C, C_{ij} is the number of nodes of community i of part A that are also in community j of part B [32]. NMI $I(A, B)$ is defined as follows [42]:

$$I(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\log(C_{ij}N/C_i\cdot C_{\cdot j})}{\sum_{i=1}^{C_A} C_i\cdot\log(C_i/N) + \sum_{j=1}^{C_B} C_{\cdot j}\log(C_{\cdot j}/N)}. \tag{3}$$

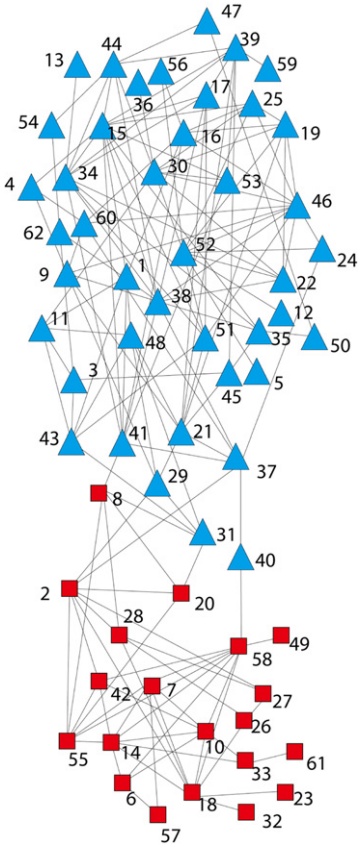

Fig. 4. Dolphin social network.

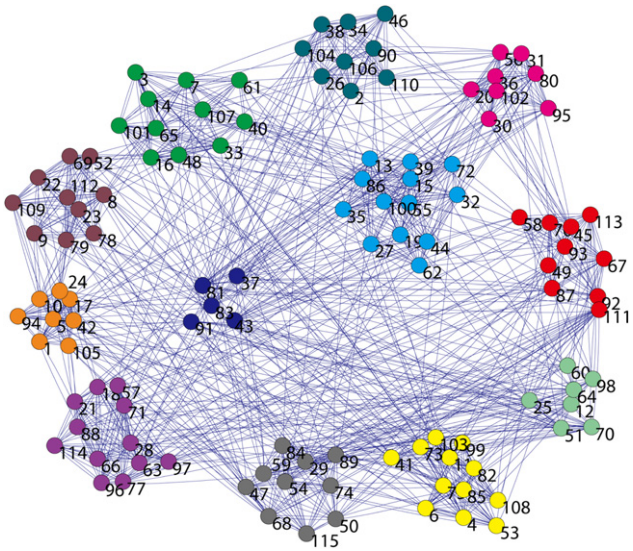

Fig. 5. American college football.

Here, $c_A(c_B)$ is the number of classes in part $A(B)$, $C_i \cdot (C_j)$ is the number of elements of C in row i (column j), and N is the total number of nodes. If $A = B$, $I(A, B) = 1$; if A and B are totally different, $I(A, B) = 0$. The higher the value of NMI, the more approximate to the true communities are the detected communities.

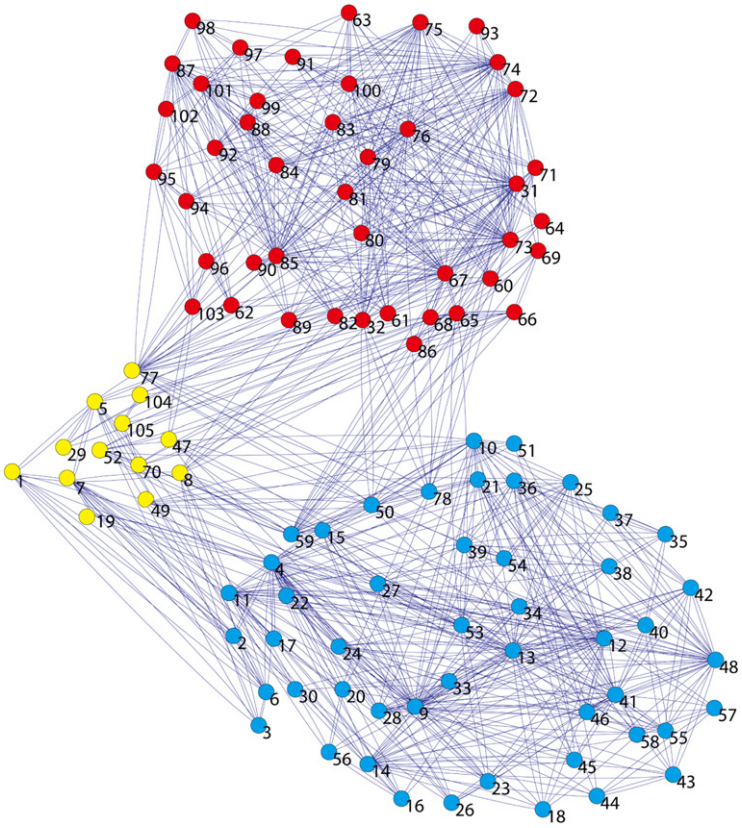

Fig. 6. Books about US politics.

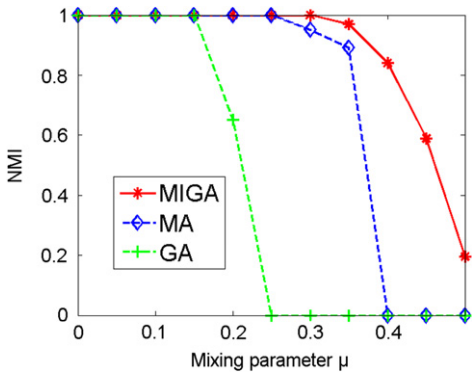

Fig. 7. NMI versus mixing parameter μ.

3.3. Simulation results and discussion

3.3.1. Computer-generated network

In the experimental part, we will analyze the performance of MIGA, the MA and the GA through adjusting the parameter μ, as illustrated in Fig. 7.

Fig. 7 shows that the value of NMI obtained by MIGA is 1 when μ changes from 0 to 0.3, which means that the algorithm can find the true community structure correctly. When μ equals 0.4, the value of NMI obtained by MIGA is larger than 0.8, which indicates that the detected part is close to the true community structure. However, it is difficult for the MA to detect the true community structure when μ is larger than 0.25. And when μ is greater than 0.15, the GA also cannot detect the true community structure. This means that MIGA could discover smaller communities.

Table 4
Parameter settings.

Parameter	Value
The number of iterations G_{max}	50
Population size S_{pop}	450
Size of the mating pool S_{pool}	$S_{pop}/2$
Tournament size S_{tour}	2
Crossover probability P_c	0.8
Mutation probability P_m	0.2

Table 5
Parameter settings.

Parameter	Value
Initial temperature T	800 000
Constant k	0.99
Loop count l	10

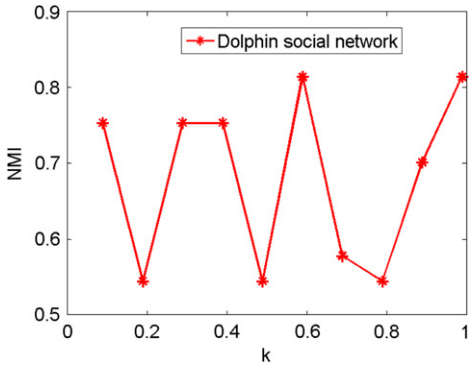

Fig. 8. The selection process of k (from 0 to 1) in the simulated annealing operation.

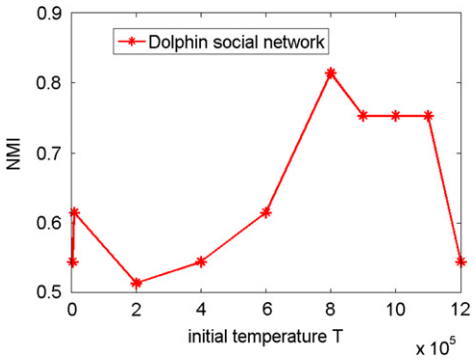

Fig. 9. The selection process of T in the simulated annealing operation.

3.3.2. Real-world networks

This part gives the simulation results of MIGA, the MA and the GA on real-world networks. From the experimental results, the effectiveness of MIGA in community detection is analyzed. The parameters of MIGA are the same as those in Ref. [19], which are shown in Table 4.

The settings of the basic parameters in the simulated annealing operation are shown in Table 5. In Table 5, l is the cycle number in each T, and l has little effect on the results of community detection. Taking the dolphin social network as an example, the selection process of the parameters k (from 0 to 1) and T are introduced, as shown in Figs. 8 and 9.

Given an initial temperature T which is large enough, first set $T = 100\,000$ and take $l = 10$. Adjust k from 0 to 1, and find the corresponding NMI, as shown in Fig. 8.

As can be seen from Fig. 8, when k approaches 0.6 or 1, the value of NMI is the maximum. So the value of k can either approach 0.6 or approach 1. However, when k approaches 0.6, the temperature T will fall very quickly, which will lead to a small number of iterations in the local search. Hence, set $k = 0.99$.

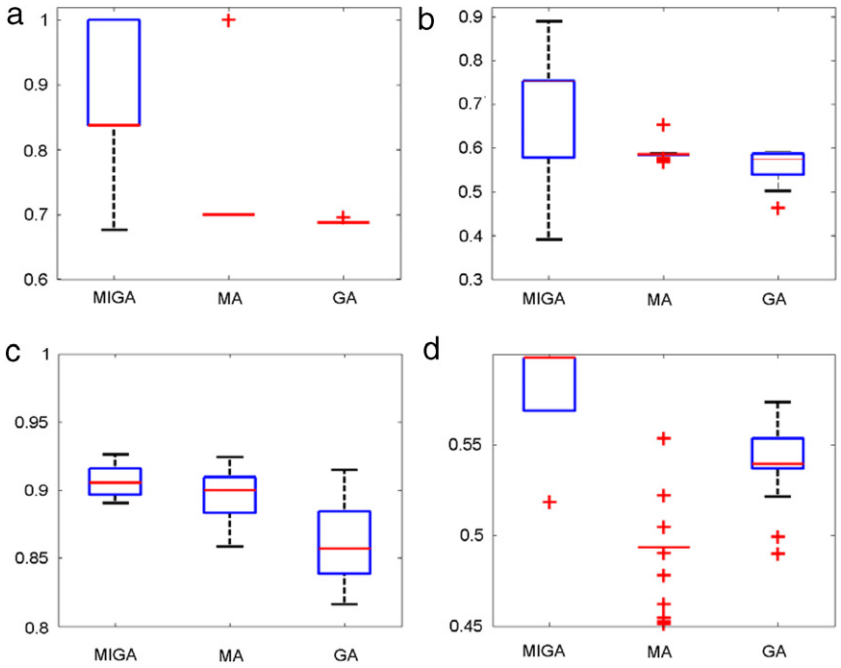

Fig. 10. Values of NMI over the 30 runs on (a) Zachary's karate club, (b) the dolphin social network, (c) American college football, (d) books about US politics.

Table 6
The values of NMI at the 30-th run.

Network	MIGA	MA	GA
Zachary's karate club	**1**	0.699	0.687
Dolphin social network	**0.814**	0.589	0.532
American college football	**0.916**	0.910	0.819
Books about US politics	**0.585**	0.455	0.521

Set the parameters $k = 0.99$ and $l = 10$. As the local search method requires T large enough, T ranges from 5 thousand to 1.2 million, as shown in Fig. 9. The maximum value of NMI is about 0.81 when T equals to 0.8 million. In summary, T is set to 0.8 million.

The box plots of the values of NMI over the 30 runs on Zachary's karate club, the dolphin social network, American college football and books about US politics for MIGA, the MA and the GA are shown in Fig. 10.

Fig. 10(a) shows the box plot over 30 runs on Zachary's karate club. It can be seen from the figure that the values of NMI obtained by the algorithm MIGA are larger than 0.8; the values of NMI obtained by MIGA are greater than those by the MA and the GA. This shows that the detected communities of MIGA are more approximate to the true communities compared with the MA and the GA on Zachary's karate club. Fig. 10(b) shows the box plot over 30 runs on the dolphin social network. It can be seen from the figure that the values of the 25-th percentile, median and the 75-th percentile obtained by MIGA are larger than those by the MA and the GA. This suggests that the detected communities of MIGA are more approximate to true communities than those of the MA and the GA on the dolphin social network. Fig. 10(c) shows the box plot over 30 runs on the American college football network. It can be seen from the figure that the values of the 25-th percentile, median and the 75-th percentile obtained by MIGA are larger than those obtained by the MA. This means that the detected communities of MIGA are more approximate to true communities than those of the MA and the GA on the American college football network. Fig. 10(d) shows the box plot over 30 runs on books about US politics. From this figure, it can be seen that the values of the 25-th percentile, median and the 75-th percentile obtained by MIGA are more approximate to the true communities than those of the MA and the GA on books about US politics.

Table 6 shows the values of NMI at the 30-th run on the four real-world networks for MIGA, the MA and the GA. The total number of detected communities at the 30-th run on the four real-world networks is shown in Table 7.

As can be seen from Tables 6 and 7, MIGA can effectively detect the community structure on Zachary's karate club, the dolphin social network, American college football and books about US politics. MIGA is able to detect 100% of the community structure information on Zachary's karate club. The number of detected communities by MIGA is equal to the correct community structures.

Fig. 11 shows the detected results on Zachary's karate club [23] of MIGA, the MA and the GA.

Table 7
The total number of detected communities at the 30-th run.

Network	MIGA	MA	GA
Zachary's karate club	**2**	3	4
Dolphin social network	**2**	5	5
American college football	**12**	11	8
Books about US politics	**3**	7	5

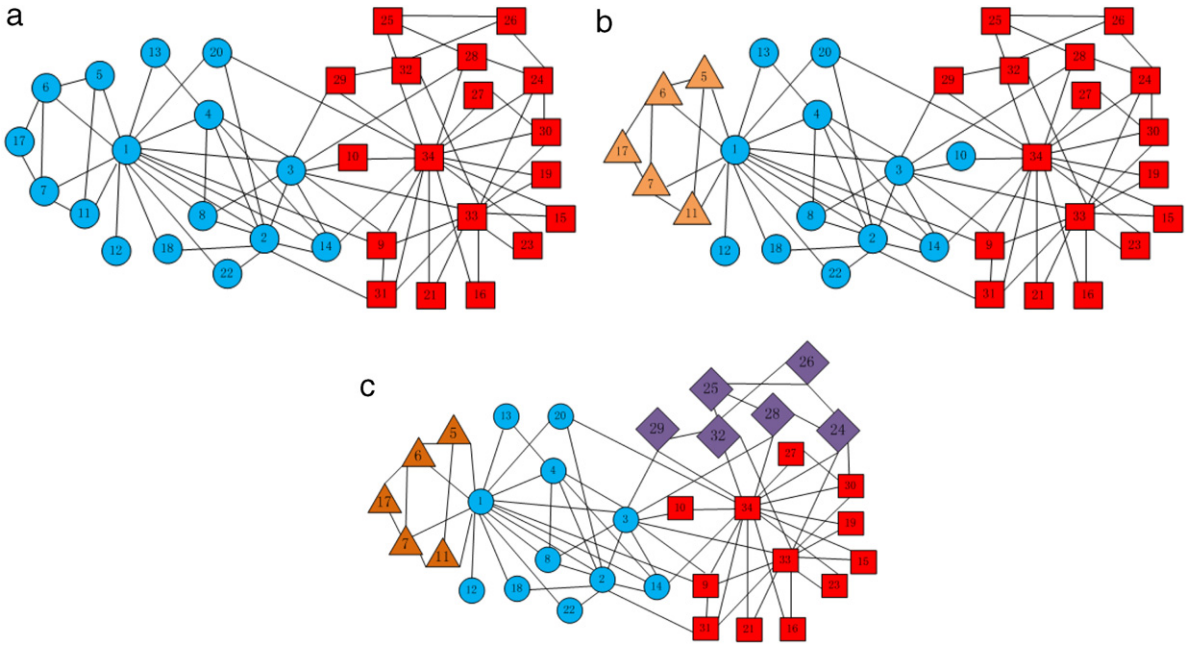

Fig. 11. (a) The detected result of MIGA on Zachary's karate club, (b) the detected result of the MA on Zachary's karate club, (c) the detected result of the GA on Zachary's karate club.

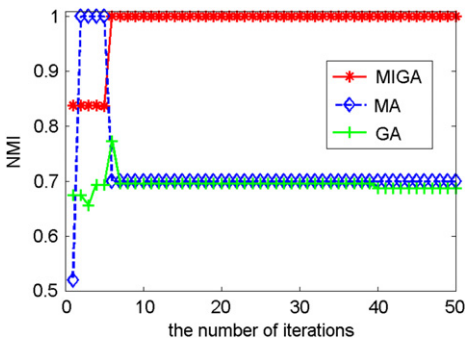

Fig. 12. The values of NMI at the 30-th run on Zachary's karate club.

It can be seen from Figs. 3 and 11 that the detected community structure of MIGA is the true community structure on Zachary's karate club. And the detected community structure information of the MA is different from the true one. This is due to one of the two large communities being divided into two smaller ones. The GA put nodes in four communities, and it cannot detect the true community structure.

Fig. 12 shows the values of NMI according to the generations ranging from 1 to 50 at the 30-th run on Zachary's karate club for MIGA, the MA and the GA. The value of NMI has reached 1 in the fifth generation. This can illustrate that MIGA can detect the true community structure. However, without the number of the given classes and simulated annealing method, the values of NMI obtained by the MA and the GA are approximately 0.7 after the fifth generation. So, for this network, MIGA performs best.

Fig. 13 shows the detected results on the dolphin social network of MIGA, the MA and the GA. It can be seen from Figs. 4 and 13 that only the 31-th node and the 40-th node are misplaced. And the detected community structure of MIGA is close

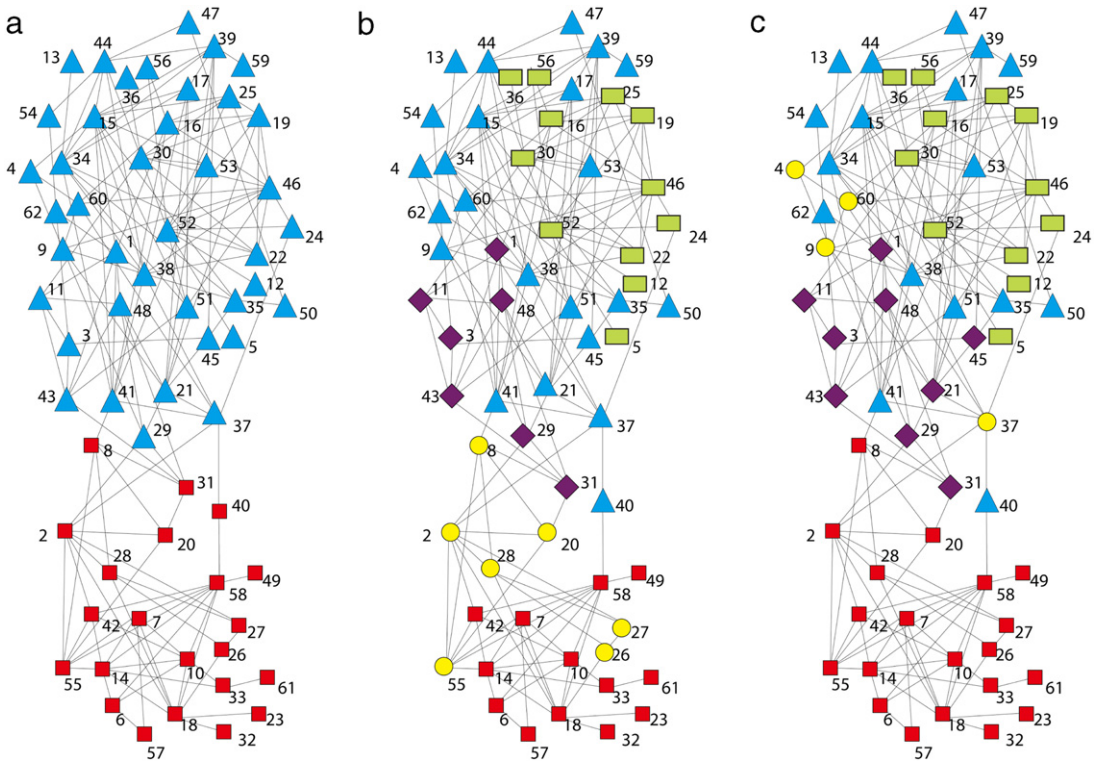

Fig. 13. (a) The detected result of MIGA on the dolphin social network, (b) the detected result of the MA on the dolphin social network, (c) the detected result of the GA on the dolphin social network.

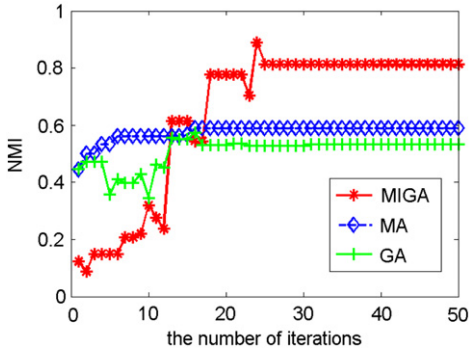

Fig. 14. The values of NMI at the 30-th run on the dolphin social network.

to the true community structure on the dolphin social network. The MA and the GA put nodes in five parts, which means that the detected community structure is different from the true one.

Fig. 14 displays the values of NMI according to the generations ranging from 1 to 50 at the 30-th run on the dolphin social network for MIGA, the MA and the GA.

It can be seen from Fig. 14 that the values of NMI obtained by MIGA are larger than those obtained by the MA and the GA after the fifth generation. The value of NMI obtained by MIGA is 0.814 at the end of the generations, and the values of NMI obtained by MA and GA are less than 0.6. This means that the detected community structure of MIGA is close to the true one (the value of NMI is about 0.8).

Fig. 15 shows the detected results on the American college football network of MIGA, the MA and the GA. It can be seen from Fig. 3 that the true community structure consists of 12 parts. Fig. 15(a)–(e) show the 9 parts, the 10 parts, the 11 parts, the 12 parts and the 13 parts detected by MIGA on the American college football network, respectively. Fig. 15(f) is the detected result of 11 parts detected by the MA on the American college football network and Fig. 15(g) gives the 7 parts detected by the GA on the American college football network.

It can be seen from Fig. 15 that 11 parts are found by the MA. Between 9 and 13 are always found by MIGA. It can be seen from Fig. 15(d) that 12 parts are detected by the proposed MIGA on the American college football network, which is

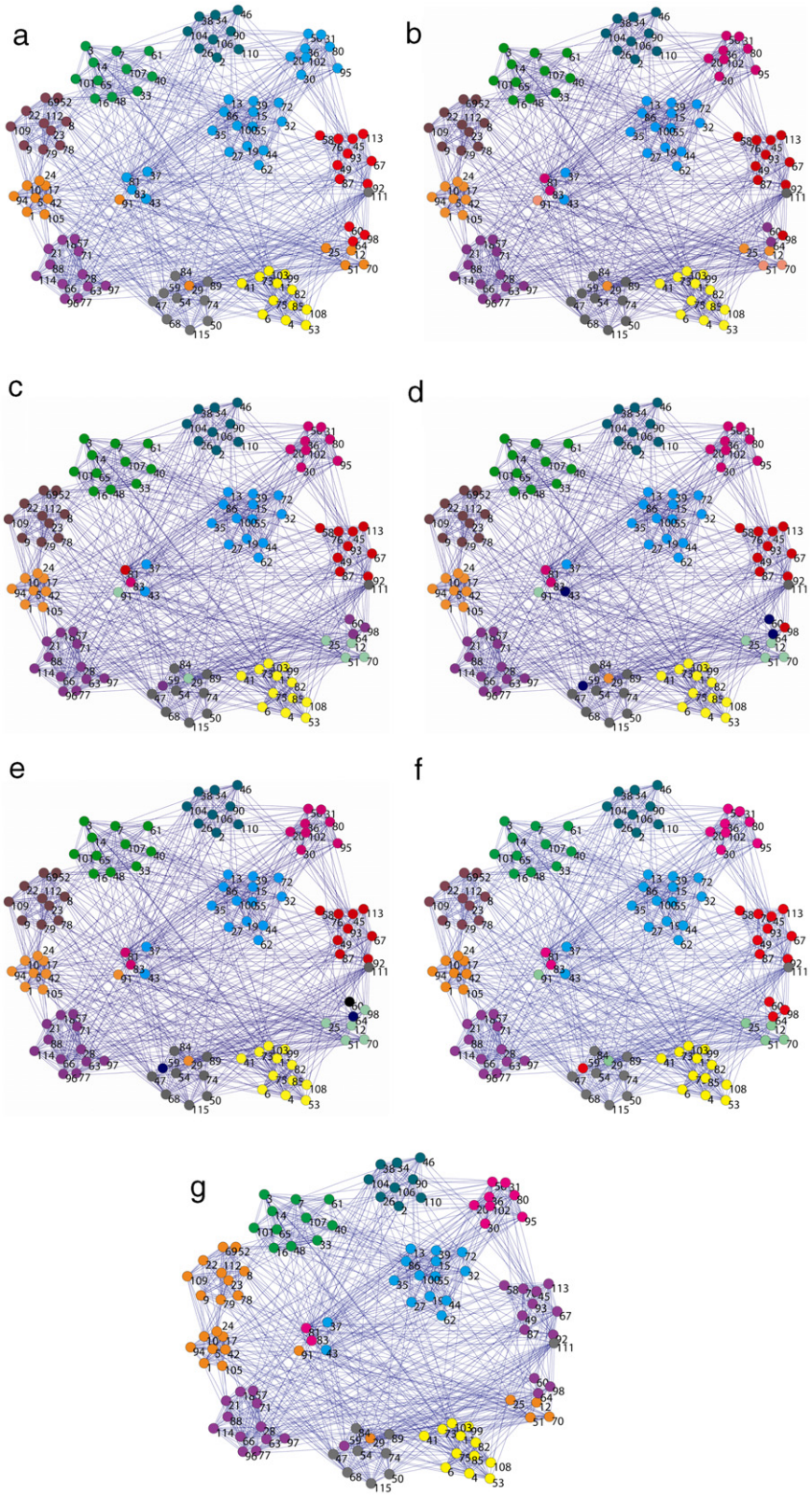

Fig. 15. (a) The 9 parts detected result of MIGA on the American college football network, (b) the 10 parts detected result of MIGA on the American college football network, (c) the 11 parts detected result of MIGA on the American college football network, (d) the 12 parts detected result of MIGA on the American college football network, (e) the 13 parts detected result of MIGA on the American college football network, (f) the detected result of the MA on the American college football network, (g) the detected result of the GA on the American college football network.

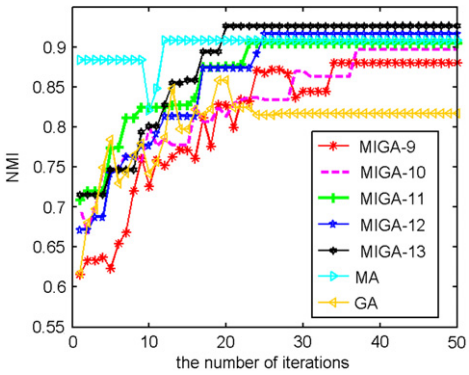

Fig. 16. The values of NMI at the 30-th run on the American college football network.

Table 8
The computation time.

Network	MIGA (s)	MA (s)
Computer-generated network	**12.82**	32.56
Zachary's karate club	**3.27**	4.01
Dolphin social network	**4.84**	11.29
American college football	**9.68**	88.36
Books about US politics	**7.69**	37.71

same as the true community structure on the American college football network, and only a few nodes are misplaced. Seven parts are found by the GA, which is different from the true one. For the American college football network, only MIGA can obtain the right classes and put most of the nodes in the right places. So, the proposed MIGA has the ability to detect a large number of communities.

More details are given in Fig. 16, which gives the values of NMI according to the generations ranging from 1 to 50 at the 30-th run on the American college football network for MIGA which are based on different prior information, the MA and the GA.

It can be seen from Fig. 16 that the values of NMI obtained by MIGA have five different results. The results of 9 parts to 13 parts by MIGA are 0.88, 0.897, 0.904, 0.916 and 0.926, respectively. The NMI value of the MA is 0.91 at the end of the generations. The value of NMI obtained by MIGA depends on the prior information (the number of community structures). When the number of community structures is selected incorrectly, MIGA can only find 9, 10 or 11 parts, all of which are different from the true community structure. The values of NMI obtained by MIGA are larger than that by the MA when the right detected community structure is selected, such as MIGA-12 and MIGA-13. The correct prior information (the class number of community structures) and the simulated annealing method make the proposed algorithm MIGA more targeted and improve the stability and accuracy of community detection. So MIGA also performs well on the American college football network.

Fig. 17 displaces the detected results on books about US politics of MIGA, the MA and the GA. It can be seen from Figs. 4 and 17 that the results obtained by MIGA put the nodes in three parts, and only a few nodes are misplaced. The MA put nodes in seven parts and the GA put nodes in five communities; these cannot detect the true community structure.

Fig. 18 shows the values of NMI according to the generations ranging from 1 to 50 at the 30-th run on books about US politics for MIGA, the MA and the GA.

It can be seen from Fig. 18 that the accuracy of all three algorithms falls in community detection. The reason is that the more complex the network, the less accurate MIGA becomes. But the value of NMI obtained by MIGA is better than those obtained by the MA and the GA at the end of the generations, which goes to show that MIGA is able to detect more than half of the true parts with the prior information and simulated annealing method; however, without the prior information and simulated annealing method, the performance of the MA and GA algorithms become worse.

3.3.3. The computation time

The average computation time of MIGA and the MA over the 10 runs is shown in Table 8.

As can be seen from Table 8, MIGA has lower computational time in both the computer-generated network and real-world networks.

4. Conclusion

In this paper, community detection based on the modularity and an improved genetic algorithm has been proposed. The first step is to take modularity Q as the objective function. The second step is to use the prior information and the last step

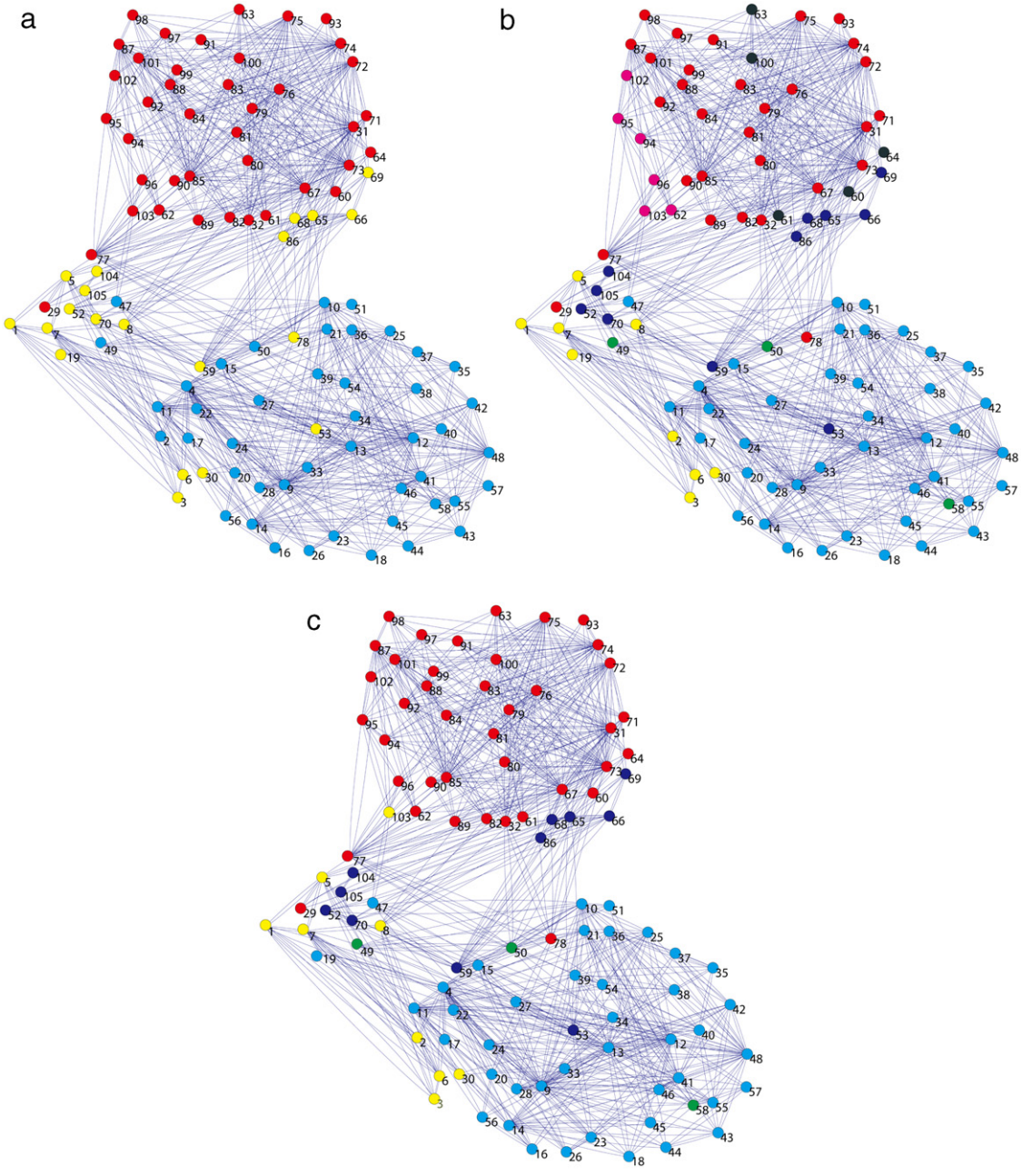

Fig. 17. (a) The detected result of MIGA on books about US politics, (b) the detected result of the MA on books about US politics, (c) the detected result of the GA on books about US politics.

is to take the simulated annealing method as the local search method. Comparing MIGA with the MA and the GA, MIGA is more targeted and has lower computational complexity in both a computer-generated network and real-world networks. However, the performance of MIGA on the dolphin social network, the American college football network and books about US politics still has room for improvement. Aiming at the above problem, our future work will improve MIGA's performance on the following two points: first, as multi-objective optimization problems can increase the search space and improve the diversity of solutions, the single objective optimization problem will be converted into two objectives optimization problem for community detection; second, the immune clonal operator will be used in the algorithm. The clonal operator can find solutions with high quality and this can benefit the search for the best solutions, which would make algorithm obtain high quality solutions and avoid falling into the local optimal solution of large-scale problems for community detection.

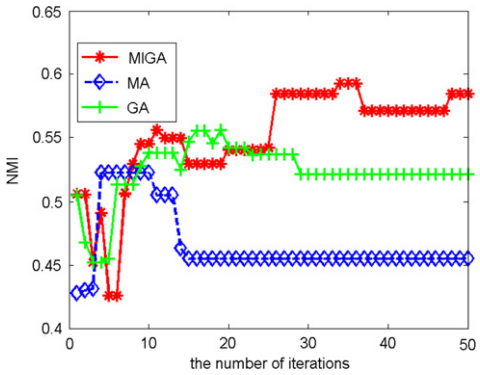

Fig. 18. The values of NMI at the 30-th run on books about US politics.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China, under Grants 61001202, 61072139 and 61003199, the China Post-Doctoral Science Foundation, under Grants 201104658 and 20090451369, and the National Research Foundation for the Doctoral Program of Higher Education of China, under Grants 20100203120008, 200807010003 and 20090203120016.

References

[1] R. Halalai, C. Lemnaru, R. Potolea, Distributed community detection in social networks with genetic algorithms, in: Proceeding Conference Intelligent Computer Communication, 2010, pp. 35–41.
[2] A. Pothen, H. Simon, K.P. Liou, Partitioning sparse matrices with eigenvectors of graphs, SIAM Journal on Matrix Analysis and Applications 11 (1990) 430–452.
[3] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proceedings of the National Academy of Sciences of the United States of America 99 (2002) 7821–7826.
[4] M.E.J. Newman, Fast algorithm for detecting community structure in networks, Physical Review E 69 (2004) 066133.
[5] M.E.J. Newman, Modularity and community structure in networks, Proceedings of the National Academy of Sciences of the United States of America 103 (2006) 8577–8582.
[6] M.E.J. Newman, E.A. Leicht, Mixture models and exploratory analysis in networks, Proceedings of the National Academy of Sciences of the United States of America 104 (2007) 9564–9569.
[7] M. Rosvall, C.T. Bergstrom, An information theoretic framework for resolving community structure in complex networks, Proceedings of the National Academy of Sciences of the United States of America 104 (2007) 7327–7331.
[8] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large scale networks, Physical Review E 76 (2007) 036106.
[9] G. Palla, I. Dereny, I. Farkas, et al., Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814–818.
[10] J.M. Kumpula, M. Kivea, K. Kaski, et al., Sequential algorithm for fast clique percolation, Physical Review E 78 (2008) 026109.
[11] J. Duch, A. Arenas, Community detection in complex networks using external optimization, Physical Review E 72 (2005) 027104.
[12] F.D. Xie, M. Ji, Y. Zhang, D. Huang, The detection of community structure in network via an improved spectral method, Physica A 388 (2009) 3268–3272.
[13] Z.H. Wu, Y.F. Lin, H.Y. Wan, S.F. Tian, K.Y. Hu, Efficient overlapping community detection in huge real-world networks, method, Physica A 391 (2012) 2475–2490.
[14] D.W. Zhang, F.D. Xie, Y. Zhang, F.Y. Dong, K.R. Hirota, Fuzzy analysis of community detection in complex networks, Physica A 389 (2010) 5319–5327.
[15] A. Faqeeh, K.A. Samani, Community detection based on the "clumpiness" matrix in complex networks, Physica A 391 (2012) 2463–2474.
[16] Y. Pan, D.H. Li, J.G. Liu, J.Z. Liang, Detecting community structure in complex networks via node similarity, Physica A 389 (2010) 2849–2857.
[17] X.H. Wang, L.C. Jiao, J.S. Wu, Adjusting from disjoint to overlapping community detection of complex networks, Physica A 388 (2009) 5045–5056.
[18] J.Q. Jiang, L.J. McQuay, Modularity functions maximization with nonnegative relaxation facilitates community detection in networks, Physica A 391 (2012) 854–865.
[19] J.H. Holland, Adaptation in Natural and Artificial Systems, Univ. of Michigan Press, Ann Arbor, MI, 1975.
[20] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[21] J.R. Koza, Genetic Programming, MIT Press, 1992;
J.R. Koza, Genetic Programming II, MIT Press, 1994.
[22] Daene C. McKinney, Genetic algorithm solution of groundwater management models, Water Resources Research 30 (1994) 1897–1906.
[23] H. Schwefel, Evolution and Optimum Seeking, John Wiley & Sons, 1995.
[24] Colin R. Reeves, A genetic algorithm for flowshop sequencing, Computers & Operations Research 22 (1995) 5–13.
[25] J. Yang, Feature subset selection using a genetic algorithm, IEEE Intelligent Systems and their Applications 13 (1998) 44–49.
[26] Gareth Jones, Peter Willett, Robert C. Glen, Andrew R. Leach, Robin Taylor, Development and validation of a genetic algorithm for flexible docking, Journal of Molecular Biology 267 (1997) 727–748.
[27] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.
[28] César O. Stoico, Danilo G. Renzi, Fernando Vericat, A genetic algorithm for the 1D electron gas, Physica A: Statistical Mechanics and its Applications 387 (2008) 159–166.
[29] Jieyu Wu, Xinyu Shao, Jinhang Li, Gang Huang, Scale-free properties of information flux networks in genetic algorithms, Physica A: Statistical Mechanics and its Applications 391 (2012) 1692–1701.
[30] D. Whitley, T. Starkweather, C. Bogart, Genetic algorithms and neural networks: optimizing connections and connectivity, Physica A 14 (1990) 347–361.
[31] Paul S. Heckerling, Ben S. Gerber, Thomas G. Tape, Robert S. Wigton, Use of Genetic algorithms for neural networks to predict community-acquired pneumonia, Physica A 30 (2004) 71–84.

[32] M.G. Gong, B. Fu, L.C. Jiao, H.F. Du, Memetic algorithm for community detection in networks, Physical Review E 00 (2011) 006100.
[33] Z. Li, S. Zhang, R.-S. Zhang, L. Chen, Quantitative function for community detection, Physical Review E 77 (2008) 036109.
[34] W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33 (1977) 452–473.
[35] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, S.M. Dawson, The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations, Behavioral Ecology and Sociobiology 54 (2003) 396–405.
[36] V. Krebs, http://www.orgnet.com (unpublished).
[37] M. Tasgin, A. Herdagdelen, H. Bingol, e-print. arXiv:0711.0491.
[38] S.W. Mahfoud, D.E. Goldberg, A genetic algorithm for parallel simulated annealing, in: Proceeding of the Conference on Parallel Problem Solving from Nature, 1992.
[39] J.S. Wu, X.H. Wang, L.C. Jiao, Synchronization on overlapping community network, Physica A 391 (2012) 508–514.
[40] T. Boseniuk, W. Ebeling, Boltzmann, Darwin- and Haeckel-strategies in optimization problems, in: Proceeding of the Conference on Parallel Problem Solving from Nature, vol. 496, 1991, pp. 429–444.
[41] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Physical Review E 78 (2008) 046110.
[42] L. Danon, A. Díaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, Journal of Statistical Mechanics: Theory and Experiment (2005) P09008.