J. Cent. South Univ. (2013) 20: 1269−1276
DOI: 10.1007/s11771-013-1611-y

Springer

A genetic algorithm for community detection in complex networks

LI Yun(李赟), LIU Gang(刘钢), LAO Song-yang(老松杨)

College of Information System and Management, National University of Defense Technology, Changsha 410073, China

Abstract: A new genetic algorithm for community detection in complex networks was proposed. It adopts matrix encoding that enables traditional crossover between individuals. Initial populations are generated using nodes similarity, which enhances the diversity of initial individuals while retaining an acceptable level of accuracy, and improves the efficiency of optimal solution search. Individual crossover is based on the quality of individuals' genes; all nodes unassigned to any community are grouped into a new community, while ambiguously placed nodes are assigned to the community to which most of their neighbors belong. Individual mutation, which splits a gene into two new genes or randomly fuses it into other genes, is non-uniform. The simplicity and effectiveness of the algorithm are revealed in experimental tests using artificial random networks and real networks. The accuracy of the algorithm is superior to that of some classic algorithms, and is comparable to that of some recent high-precision algorithms.

Key words: complex networks; community detection; genetic algorithm; matrix encoding; nodes similarity

1 Introduction

With recent advances in complex network research, many real networks have been shown to possess community structure; in other words, the whole network is composed of several communities. Community structure emerges as the third important characteristic of complex networks, the other properties being small-world and scale-free. In complex networks, the nodes in a community connect closely with each other, while communities connect sparsely with other communities. Community structure acquires different meanings in different application fields. For example, in a social network, the community represents a group of people closely connected or possessing similar characteristics, while in world-wide-web terminology, a webpage community comprises a set of webpages linked by a common theme. In 2002, NEWMAN and GIRVAN [1] initiated a new development in complex networks research, i.e., community detection in complex networks.

Community detection attempts to gain a meaningful community partition of complex networks by using the information hidden in network topology. Assuming that all nodes in complex networks can be assigned to groups, these nodes are divided into several mutually exclusive communities. Therefore, community detection in complex networks is a typical NP combinatorial optimization problem. Owing to its high performance in solving NP-hard problems, genetic algorithm (GA) has been widely applied by scholars across the world to detect the community structure of complex networks.

We develop a simple and effective GA based on matrix encoding and nodes similarity (abbreviated to MENSGA), for community detection in complex networks. In MENSGA, network modularity function is set as target and fitness function, matrix encoding is adopted, and nodes similarity is used to generate an initial population. The details of the algorithm are described in the next section.

2 MENSGA

2.1 Network modularity function

Let $G(V, E)$ represent a complex network, where V and E represent the node set and edge set of the network, respectively, and

$$V=\{v_i\,|\,i=1,\ 2,\ \cdots,\ n\},\ E=\{e_i\,|\,i=1,\ 2,\ \cdots,\ m\}$$

where n and m are the numbers of nodes and edges, respectively.

Network modularity function, also called Q-function, is widely used to quantitatively evaluate the community partition of complex networks. It is expressed as follows [1]:

$$Q=\frac{1}{2m}\sum_{ij}\left[a_{ij}-\frac{k_i k_j}{2m}\right]\delta(c_i,c_j) \tag{1}$$

where a_{ij} is an element of the network adjacency matrix $\boldsymbol{A}=(a_{ij})_{n\times n}$. If v_i and v_j are connected by an edge, then

Received date: 2012−05−10; **Accepted date:** 2012−10−18
Corresponding author: LI Yun, PhD Candidate; Tel: +86−13787092573; E-mail: lee2003100@163.com

a_{ij}=1, or a_{ij}=0; c_i and c_j represent the communities to which v_i and v_j belong, respectively; if c_i=c_j, then $\delta(c_i, c_j)$=1, or $\delta(c_i, c_j)$=0; k_i and k_j are respectively the degrees of v_i and v_j, with $k_i=\sum_{j=1}^{n} a_{ij}$, $k_j=\sum_{i=1}^{n} a_{ij}$, i, j=1, 2, ⋯, n. Note that $|Q|\leq 1$. Higher Q values (close to 1) correspond to stronger community partition of G.

2.2 Individual encoding

Present research on community detection in complex networks widely employs string encoding [2−7] and graph-based encoding [8−11]. The former probably does not admit traditional crossover operations, while the latter requires additional decoding [2, 7]. To avoid these shortcomings, individuals are encoded into a binary matrix, as described in Ref. [12]. Then, regardless of how G is partitioned, the community partition of G is always represented by a binary matrix $\boldsymbol{M}$:

$$\boldsymbol{M}=\begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1t} \\ m_{21} & m_{22} & \cdots & m_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nt} \end{pmatrix}$$

where $\boldsymbol{M}$ is an $n\times t$ matrix, t ($1<t<n$) is the number of communities after partitioning G. Row i ($1\leq i\leq n$) of $\boldsymbol{M}$ corresponds to the assigning result of v_i, and column j ($1\leq j\leq t$) corresponds to community c_j. If v_i belongs to c_j, then m_{ij}=1, or m_{ij}=0.

Since any node of G must and can belong to a single community, $\boldsymbol{M}$ must follow the constraints defined in Eqs. (2) and (3):

$$\sum_{j=1}^{t} m_{ij}=1 \tag{2}$$

$$\sum_{i=1}^{n} m_{ij}>0 \tag{3}$$

A simple example of the encoding process is shown in Fig. 1. The network comprises 6 nodes divided into two communities v_1, v_3, v_6 and v_2, v_4, v_5, respectively (delineated by dotted lines). The community partition of the network is encoded by the matrix $\boldsymbol{M}$ in the right section of Fig. 1.

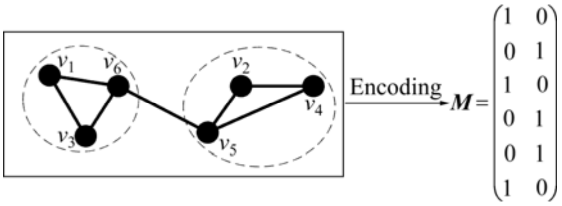

Fig. 1 Example of matrix encoding

Different community structures can be generated by different partitions of G, thus the number of columns of $\boldsymbol{M}$ is variable. Regardless of the ordering of the columns, $\boldsymbol{M}$ always represents the same partition of G, unless any entry of the columns changes.

2.3 Population initialization

Inspired by the quantitative description of nodes similarity of a complex network presented in Ref. [13], we propose a new population initialization method based on traditional clustering.

LEICHT et al [13] took into account the similar relation between nodes in both long and short paths of network topology, and derived Eq. (4) to compute the nodes similarity of a complex network.

$$\boldsymbol{DSD}=\frac{\alpha}{\lambda_{\max}}\boldsymbol{A}(\boldsymbol{DSD})+\boldsymbol{I} \tag{4}$$

where $\lambda_{\max}$ is the maximum eigenvalue of $\boldsymbol{A}$; $\boldsymbol{D}$ is an n-order diagonal square matrix whose diagonal entries d_{ii}=$\sum_{j=1}^{n} a_{ij}$, i, j=1, 2, ⋯, n, represent degrees of nodes; $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{S}$ is the nodes similarity matrix. For optimal $\boldsymbol{S}$, α ($0<\alpha<1$) is generally set to 0.97 [13].

We propose the algorithm for population initialization based on nodes similarity (abbreviated to PINS) as follows:

Step 1: Obtain the nodes similarity matrix $\boldsymbol{S}$ by iterating Eq. (4) until the left and right hand sides of the equation converge.

Step 2: Randomly select v_1, v_2, ⋯, v_t as centers of t communities, $1<t<n$.

Step 3: By following the max-similarity principle that a node has greater similarity with one community center than with the others, use $\boldsymbol{S}$ to assign each non-community-center node to a selected community.

Repeat **Steps 2** and **3** for all members of the population (up to and including P_n).

The time complexity of **Step 1** is $O(ln^3)$, where l is the iteration time. In **Step 2**, the community centers chosen are different for each time, which provides high diversity to the initial individuals, and the time complexity can be ignored. **Step 3** greatly reduces the possibility that node pairs with low similarity or no link between each other are divided into the same community, thus assigning initial individuals with enhanced accuracy, lessening the algorithm's search space and speeding up the convergence. The time complexity of **Step 3** is $O[P_n(n-\bar{t})\bar{t}]$, where $\bar{t}$ is the average number of communities after each partition. Therefore, the time complexity of PINS is $O[n^3+(n-\bar{t})\bar{t}]$.

2.4 Crossover operator

Having quantitatively described the quality of individuals' genes, a traditional single-point crossover operator is adopted for crossing individuals. Referring to Section 2.2, a column of $\boldsymbol{M}$ corresponds to one gene of an individual, as well as to a single community of G. Generally, as the average nodes similarity in a community enlarges, the community structure improves. Crossover operation based on the quality of individuals' genes is then implemented as follows:

Step 1: Use Eq. (1) to calculate the fitness of all individuals, and sort the individuals in descending order according to their fitness. Next, select the top $P_n\times P_{\mathrm{c}}$ individuals possessing optimal fitness and pair them to cross. P_{c} ($0<P_{\mathrm{c}}<1$) is a fixed constant, and $(P_n\times P_{\mathrm{c}})\bmod 2=0$. As an illustrative example, let the top six individuals with optimal fitness sorted in descending order be represented by I_1, I_2, I_3, I_4, I_5 and I_6. Pair them to cross; then successively cross I_1 and I_6, I_2 and I_5, I_3 and I_4.

Step 2: Measure the genetic quality of the crossover individuals. Suppose that in $\boldsymbol{M}$, column $\boldsymbol{M}_i=(m_{1i}, m_{2i}, \cdots, m_{ni})^{\mathrm{T}}$ contains r nonzero elements, m_{u_1i}, m_{u_2i}, $\cdots$, m_{u_ri}, where $1\le u_p\le n$, $1\le p\le r$, $1\le r<n$, then use $\boldsymbol{S}$ and Eq. (5) to calculate the average similarity $\bar{s}_i$ of v_{u_1}, v_{u_2}, $\cdots$, v_{u_r}, which also indicates the quality of gene i of the individual. In Eq. (5), $s_{u_pu_q}$ is the similarity between v_{u_p} and v_{u_q}.

$$\bar{s}_i=\frac{\sum\limits_{p=1}^{r}\sum\limits_{p<q}^{r}s_{u_pu_q}}{r(r-1)/2} \tag{5}$$

Sort each crossover individual's genes in descending order according to $\bar{s}_i$, whose value is 0 when a community possesses only one node.

Step 3: Implement traditional single-point crossover. Exchange the best genes of two crossover individuals (corresponding to the first two columns of their encoding matrices) to generate two new individuals. As shown in Fig. 2, if ($\boldsymbol{X}_1$, $\boldsymbol{X}_2$, $\cdots$, $\boldsymbol{X}_p$) and ($\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$, $\cdots$, $\boldsymbol{Y}_q$) respectively represent the encoding matrices' column vectors of individuals $\boldsymbol{X}$ and $\boldsymbol{Y}$, the exchange of $\boldsymbol{X}_1$ and $\boldsymbol{Y}_1$ generates two new individuals $\boldsymbol{X}'=(\boldsymbol{Y}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_p)$ and $\boldsymbol{Y}'=(\boldsymbol{X}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_q)$.

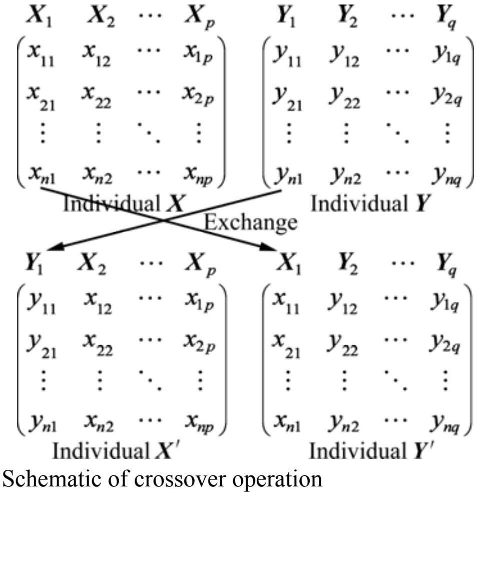

Fig. 2 Schematic of crossover operation

It is worth noting that the new individuals generated by **Step 3** may be illegal, which means that their encoding matrixes may violate Eqs. (2) and (3). In this case, some nodes may not belong to a community, or may belong to more than one community, which contradicts the presupposition. These invalid solutions are revised as follows:

1) Form a new community composed of nodes belonging to none of the communities. Thus, if $\boldsymbol{M}$ contains rows whose entries are all 0 (called 0-rows for brevity), then add a new all-zero column to $\boldsymbol{M}$, and set the elements intercepting both the new column and the 0-rows to 1;

2) Assign nodes belonging to more than one community into the communities holding most of their neighbors, which is a process known as neighbor-most principle [7]. Thus, if $\boldsymbol{M}$ contains a row with several 1s (called 1s-row for brevity), identify the column with the highest number of neighbors of the node represented by the 1s-row, retain that column element of the 1s-row at 1, and set all other elements in the 1s-row to 0.

In Fig. 3, suppose that v_3 has neighbors v_1 and v_6, and an invalid solution is reached in which v_2 and v_4 belong to none of the existing communities, and v_3 belongs to both c_2 and c_3. To revise this solution, add a new all-zeros column to the invalid solution, then set its 2nd and 4th elements to 1. Additionally, set all elements in the 3rd row to 0 except the first, which is retained as 1. Delete all columns containing zeros only.

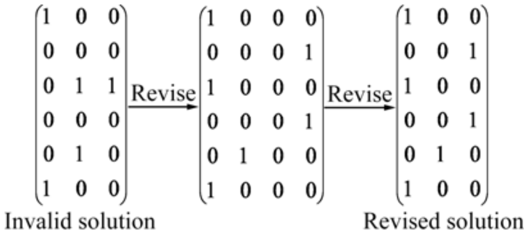

Fig. 3 Example of revision of an invalid crossover solution

In the above crossover operation, all sortings are made using bubble sort. Denote the average numbers of nodes in each community and those which are divided into more than one community after each partition, by αn and βn, respectively, where $0<\alpha<1$, $0<\beta\le1$. Then the respective time complexities of the three steps are $O(n^2+P_n^2)$, $O\{(P_n\times P_{\mathrm{c}})[(\alpha n)^2\bar{t}+\bar{t}^2]\}$ and $O[(P_n\times P_{\mathrm{c}})\cdot\beta n\bar{t}]$, yielding a time complexity for the crossover operation of $O(n^2\bar{t}+\bar{t}^2)$.

2.5 Mutation operator

In the matrix encoding, non-uniform mutation based on individuals' genes sorted in order of descending fitness is used to change the gene through split or fusion operations. In $\boldsymbol{M}$, the split operation is achieved by randomly splitting the last column (whose elements sum to more than 1) into two columns that replace the original one; whereas the fusion operation moves the none-zero elements of the last column to other columns according to the neighbor-most principle, and deletes the last column. The details of the mutation operation are as follows:

Step 1: Select the bottom-ranked $P_n \times P_m$ individuals which have minimum fitness, then measure and sort their genes according to the method applied to the 2nd step of crossover operation. Here, P_m ($0<P_m<1$) is a fixed constant and $P_n \times P_m$ is an integer.

Step 2: If ***M*** contains just two columns, implement the split operation; if ***M*** contains three or more columns, implement split or fusion operation randomly. During the fusion operation, if most of the neighbors of a node belonging to more than one community reside in the least fit community (corresponding to the last column), the node is assigned to the community in which the second most neighbors reside.

In Fig. 4, suppose that v_3 has neighbors v_1 and v_6, v_5 has neighbors v_2 and v_4, then in the mutation individual, a split operation splits the last column into two columns, enabling v_3 and v_5 to each constitute a new community; while a fusion operation sets the 3rd element in the 1st column and the 5th in the 2nd column to 1, and deletes the last column.

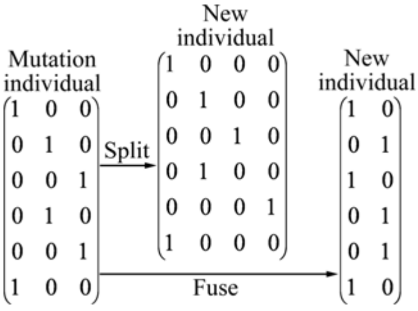
Mutation individual
$\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\\0&1&0\\0&0&1\\1&0&0\end{pmatrix}$
Split
New individual
$\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\\1&0&0&0\end{pmatrix}$
Fuse
New individual
$\begin{pmatrix}1&0\\0&1\\1&0\\0&1\\0&1\\1&0\end{pmatrix}$

Fig. 4 Example of a mutation operation

Denote the average number of nodes involved in fusion operations after each partition as γn, with $0<\gamma<1$. The respective time complexities of the two mutation steps are then $O\{(P_n \times P_m)((\alpha n)^2\bar{t}+\bar{t}^2)\}$ and $O[(P_n\times P_m)\gamma n\bar{t}]$, yielding an overall time complexity for the mutation operation of $O(n^2\bar{t}+\bar{t}^2)$.

2.6 Selection operator and description of MENSGA

By using $\mu+\lambda$ strategy, the top P_n individuals with optimal fitness are selected from the parent generation and the new population generated by crossover and mutation, as the progeny generation [7]. In MENSGA, these procedures are implanted as follows:

Algorithm input: This comprises the network adjacency matrix ***A*** and the parameters of MENSGA given in Table 1.

Table 1 Parameters of MENSGA

Parameter	Value	Description
α	0.97	Parameter used to calculate nodes similarity
P_n	100	Number of individuals in population
P_c	0.8	Ratio of crossover individuals to all individuals of population
P_m	0.2	Ratio of mutation individuals to all individuals of population
N_{max}	100	Maximum number of iterations

Algorithm output: Encoding matrix ***M*** representing the community partition of a complex network.

Terminal condition to end iteration: The algorithm runs through N_{max} iterations.

Algorithm pseudo-code:

```
1) Use PINS to generate initial population $P_{original}$;
2) for $i=1:N_{max}$
  3) Calculate the fitness of each individual using Eq. (1), and sort individuals in descending fitness order;
  Generate $P_n(P_c+P_m)$ new individuals through crossover and mutation to form new population $P_{new}$;
  4) Select the fittest $P_n \times P_c$ individuals to cross, and calculate the quality of their genes using Eq. (5), then sort the genes of each crossover individual in order of descending quality;
  5) Pair crossover individuals and exchange their best genes;
  6) Revise invalid individuals generated by the crossover operation;
  7) Select the least fit $P_n \times P_m$ individuals to mutate, and calculate the quality of their genes using Eq. (5), then sort the genes of each mutation individual in order of descending quality;
  8) Allow mutation individuals to mutate non-uniformly;
  Following completion of Steps 4 to 8, a new population is obtained $P_{new}$;
  9) Use Eq. (1) to compute the fitness of all individuals in $P_{new}$, integrate $P_{new}$ and $P_{original}$, and select the top $P_n$ individuals possessing optimal fitness as the progeny generation;
  end
10) Select the maximally fitted individual as the community partition result of the complex network.
```

Referring to Sections 2.3−2.5, and noting that $\bar{t}$ is usually far less than n, the time complexity of MENSGA is $O(n^3)$.

3 Experimental analysis

3.1 Experimental description

Similarly to Refs. [2] and [7], the performance of MENSGA is tested on artificial random networks and five real networks, and is compared with the classical algorithms GN [14] and FN [15], as well as contemporary high-accuracy algorithms TGA [5], CCGA [7] and LGA [2]. To avoid repetition, some of the performance data are replicated from Ref. [2]. It is worth noting that, to achieve high accuracy, TGA, CCGA and LGA require many optimizing steps.

The experiment was run on a Microsoft Windows Server 2003 (X64) operation system using a Matlab 7 programming platform; Intel (R) Core (TM)2 Duo CPU T8300 @ 2.40GHz processor, 4.00 GB memory and 350 GB hard disk.

MENSGA parameter values were set as given in Table 1. Specially, P_n, P_c, P_m and N_{max} could be altered as appropriate for the situation. In practice, for the networks used in the following experiments, we have found that it is optimum to set P_c as 0.8 and P_m as 0.2 by experimenting repeatedly.

3.2 Experimental results and analysis

3.2.1 Artificial random network

RN (N, C, D, Z_{out}) [14] represents an artificial random network. N=32 is the number of nodes in each community, C=4 is the number of communities, D=16 is the degree of each node, Z_{out} is the number of links between each node and nodes in other communities. When Z_{out} is small, the community structure of the network is well-defined; otherwise, it is poorly characterized, especially when $Z_{out}\geq 8$. Each algorithm is tested 50 times on this artificial random network, and the results are shown in Fig. 5.

In Fig. 5, the y-axis indicates average NMI similarity [16] between the 50 independently-generated community partitions of the network and its true community structure. Note that this measure assesses the algorithm's accuracy. From Fig. 5, it is seen that MENSGA outperforms GN, FN and TGA in the artificial random network test. In the case of unclear community structure, as occurs particularly when Z_{out}=8, MENSGA is inferior to CCGA and LGA. However, the overall accuracy of MENSGA exceeds that of the other algorithms.

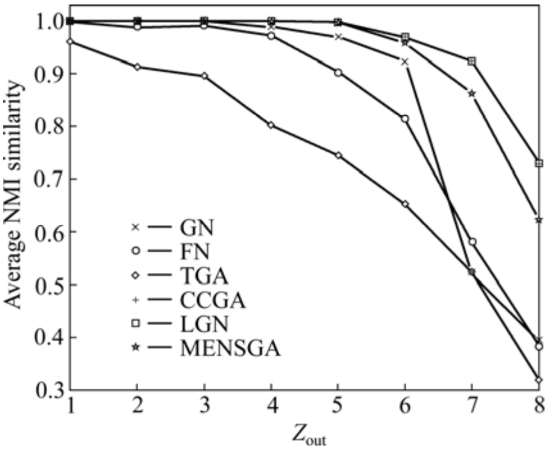

Fig. 5 Accuracies of GN, FN, TGA, CCGA, LGA and MENSGA running on artificial random networks

In an additional test, setting Z_{out}=7, 10 individuals were randomly generated by PINS. Table 2 gives the accuracy of the individuals, represented by the NMI similarity between their community partition generated by PINS and the true community structure of the network. In Table 3, the upper and lower triangular data are the NMI and Jaccard similarities between the individuals, respectively [17]. Collectively, Tables 2 and 3 reveal an average accuracy of the 10 individuals of 39.08%, with large differences between them, indicating that PINS-generated initial individuals are diverse yet retain a respectable level of accuracy. It is worth noting that, since the calculation methods of NMI and Jaccard similarity differ, the NMI and Jaccard similarities tend to differ by an order of magnitude (Table 3).

3.2.2 Real network

The performance of MENSGA was further tested on five widely used real networks [18]. The specifications of these networks are listed in Table 4.

MENSGA was run 50 times on each real network. The average Q-function values obtained for each network were compared against those of GN, FN, TGA, CCGA and LGA reported in Ref. [2]. The results are summarized in Table 5.

For each real network, MENSGA returns approximately the same result each time and converges rapidly. For example, the convergence of MENSGA running on American college football network is shown in Fig. 6.

In Table 4, the community structures of the first three real networks are known in advance. Thus, the performance of MENSGA was analyzed further by

Table 2 Accuracies of 10 individuals randomly generated by PINS running on *RN*(32, 4, 16, 17)

Individual	1	2	3	4	5	6	7	8	9	10	Average
Accuracy/%	44.79	45.66	30.66	32.60	40.66	46.81	38.11	35.13	45.29	31.14	39.08

Table 3 NMI similarity and Jaccard similarity between 10 individuals in Table 2

Individual	1	2	3	4	5	6	7	8	9	10	Average similarity
1	0	0.919 5	0.648 5	0.708 5	0.766 8	0.850 2	0.650 9	0.701 3	0.917 7	0.455 6	Average NMI similarity: 0.6200
2	0.087 0	0	0.678 8	0.735 3	0.812 7	0.871 9	0.671 3	0.716 0	0.942 3	0.480 1	
3	0.028 1	0.020 7	0	0.483 1	0.571 5	0.574 7	0.402 0	0.422 1	0.676 0	0.273 2	
4	0.031 9	0.016 8	0.083 6	0	0.604 4	0.668 6	0.4974	0.489 6	0.718 0	0.376 4	
5	0.038 1	0.053 2	0.115 6	0.083 3	0	0.755 8	0.544 8	0.598 2	0.807 7	0.428 2	
6	0.080 4	0.063 2	0.029 1	0.059 1	0.101 6	0	0.604 2	0.651 7	0.863 1	0.417 1	Average Jaccard similarity: 0.0544
7	0.026 9	0.013 9	0.072 8	0.099 1	0.074 2	0.044 7	0	0.437 2	0.669 1	0.298 1	
8	0.036 8	0.012 7	0.068 4	0.050 6	0.081 4	0.053 4	0.061 9	0	0.728 6	0.329 9	
9	0.092 8	0.085 7	0.027 4	0.007 0	0.063 2	0.043 7	0.016 9	0.032 1	0	0.480 1	
10	0.014 6	0.008 7	0.091 3	0.085 3	0.066 6	0.029 0	0.110 8	0.073 6	0.011 5	0	

Table 4 Specifications of real networks used in experiments

Network name	Number of nodes	Number of edges
Zachary's karate club	34	78
Dolphin sociality	62	159
Books on US politics	105	441
American college football	115	613
Jazz musicians	198	5484

Table 5 Average Q-function values of GN, FN, TGA, CCGA, LGA and MENSGA running on 5 real networks

Network name	GN	FN	TGA
Zachary's karate club	0.401 3	0.252 8	0.403 9
Dolphin sociality	0.470 6	0.371 5	0.524 1
Books on US politics	0.516 8	0.502 0	0.524 5
American college football	0.599 6	0.454 9	0.593 7
Jazz musicians	0.405 1	0.403 0	0.440 6

Network name	CCGA	LGA	MENSGA
Zachary's karate club	0.419 8	0.419 8	0.419 8
Dolphin sociality	0.527 3	0.528 0	0.527 2
Books on US politics	0.526 9	0.527 2	0.526 2
American college football	0.600 5	0.604 6	0.604 4
Jazz musicians	0.444 5	0.444 9	0.444 7

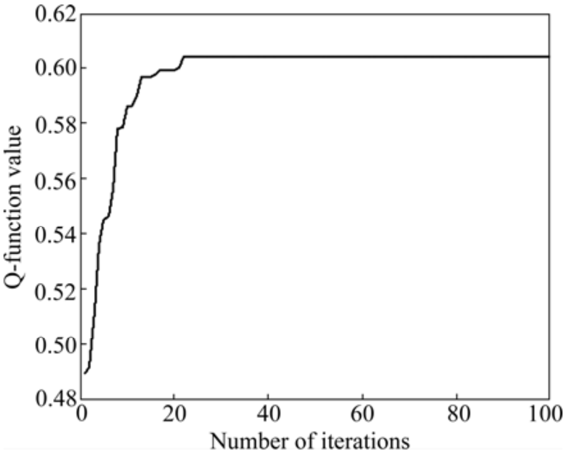

Fig. 6 Convergence of MENSGA running on American college football network

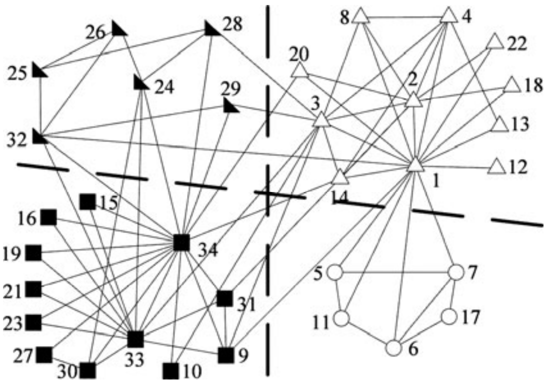

Fig. 7 Community partition result of Zachary's karate club network using MENSGA

applying the algorithm to these networks.

1) Zachary's karate club network

This network reflects the social relations of a karate club in an American university. Its nodes represent club members, and an edge indicates social communication between two club members. The club splits into two independent clubs due to internal divergence between the original coach and director. Figure 7 shows the community partitions generated by a single random run of MENSGA on this network. The black and white nodes delineate two independent communities following the split, while the four node shapes represent four small communities of different sizes. Nodes 1 and 34 represent the coach and director, respectively. Figure 7 reveals that MENSGA not only accurately discovers the network's real community structure, but identifies small communities nested within the known ones. When MENSGA is run 50 times on this network, the average value of the Q-function is 0.419 8, which exceeds that corresponding to its real community structure (0.371 5).

2) Dolphin sociality network

This network reflects the contact of dolphins within male and female communities. Its nodes represent dolphins, in which an edge indicates that two dolphins contact frequently. Figure 8 illustrates the output of a single random run of MENSGA on this network. The square nodes represent female dolphins, and different square nodes delineate the small communities within the female dolphin population. Triangular nodes represent male dolphins. Figure 8 shows that MENSGA not only accurately discovers the network's real community structure, but identifies smaller communities among female dolphins. Running MENSGA 50 times on this network yields an average Q-function value of 0.527 2, which again exceeds that of the real community structure (0.372 2).

3) American college football network

This network is a match network based on an American college football regular season plan in the autumn of 2000. The nodes represent teams, and an edge indicates that matches are played between the two connected teams. In the regular season, 115 teams attend 12 conferences of different sizes. The majority of matches are played between teams within the same conference, thus the 12 conferences constitute the network's 12 real communities. When MENSGA is randomly run on this network a single time, ten communities emerge as shown in Fig. 9. The nodes with different gray scales and shapes represent teams from different conferences. Figure 9 shows that MENSGA can correctly assign most of the teams belonging to the same

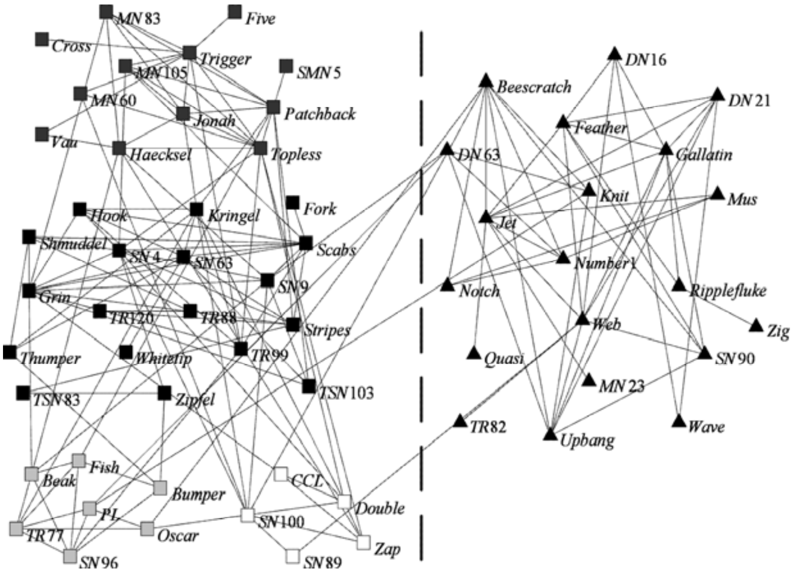

Fig. 8 Community partition result of Dolphin social network using MENSGA

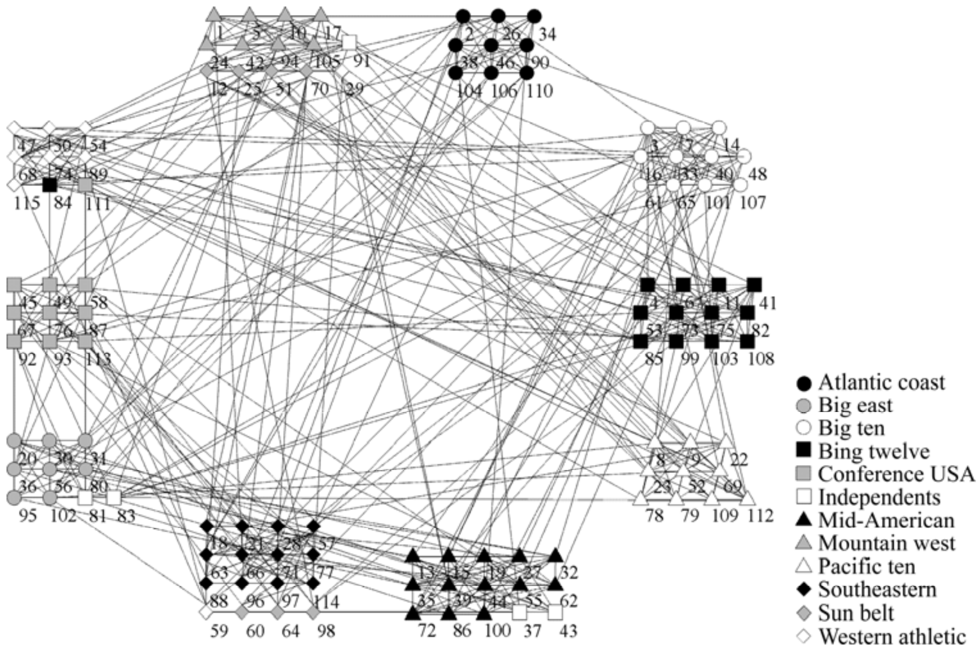

Fig. 9 Community partition result of American college football network using MENSGA

conference into the equivalent community; however, teams from Independents and Sun Belt are placed into other communities. This is reasonable since more inter-community than intra-community matches are played by these teams. In addition, teams from Independents are independent [14]. The average Q-function value obtained from 50 runs of MENSGA on this network is 0.604 4, which, like the above networks, exceeds that corresponding to its real community structure (0.551 8).

4 Conclusions

1) In the presented generic algorithm for community detection in complex networks, matrix encoding enables traditional individual crossover and requires no additional decoding.

2) Initial individuals generated by nodes similarity approaches are diverse yet retain an acceptable level of accuracy.

3) Overall, MENSGA outperforms GN, FN and TGA.

4) MENSGA is inferior to CCGA and LGA when processing a network with unclear community structure, but can detect the community structures of real networks to the same level of accuracy.

5) MENSGA requires no additional optimizing steps. It adopts matrix encoding, uses nodes similarity to initialize the population and conducts simple crossover and mutation operations, yet achieves high accuracy. Therefore, it is a simple and effective genetic algorithm for community detection in complex networks.

Acknowledgements

We'd like to express our sincere appreciation to Dr. JIN Di from Computer Science and Technology College, Jilin University, for his sincere guidance and help.

References

[1] NEWMAN M E J, GIRVAN M. Finding and evaluating community structure in networks [J]. Physical Review E, 2004, 69(2): 026113.

[2] JIN Di, LIU Jie, YANG Bo, HE Dong-xiao, LIU Da-you. Genetic algorithm with local search for community detection in large-scale complex networks [J]. Acta Automatic Sinica, 2011, 37(7): 873−882. (in Chinese)

[3] LI Shu-zhuo, CHEN Ying-hui, DU Hai-feng, FELDMAN M W. A genetic algorithm with local search strategy for improved detection of community structure [J]. Complexity, 2010, 15(4): 53−60.

[4] LIU Xin, LI De-yi, WANG Shu-liang, TAO Zhi-wei. Effective algorithm for detecting community structure in complex networks based on GA and clustering [C]// SHI Yong, ALBADA G D V, DONGARRA J, SLOOT P M A. Computational Science – ICCS 2007 – 7th International Conference. Berlin Heidelberg: Springer, 2007: 657−664.

[5] GOG A, DUMITRESCU D, HIRSBRUNNER B. Community detection in complex networks using collaborative evolutionary algorithms [C]// COSTA F A E, ROCHA L M, COSTA E, HARVEY I, COUTINHO A. Advances in Artificial Life – 9th European Conference, ECAL 2007. Berlin Heidelberg: Springer, 2007: 886−894.

[6] TASGIN M, HERDAGDELEN A, BINGOL H. Community detection in complex networks using genetic algorithms [J/OL]. 2007−11−04. http://arxiv.org/PS_cache/arxiv/pdf/0711/0711.0491v1.pdf.

[7] HE Dong-xiao, ZHOU Xu, WANG Zuo, ZHOU Chun-guang, WANG Zhe, JIN Di. Community mining in complex networks clustering combination based genetic algorithm [J]. Acta Automatica Sinica, 2010, 36(8): 1160−1170. (in Chinese)

[8] PIZZUTI C. GA-net: A genetic algorithm for community detection in social networks [C]// RUDOLPH G, JANSEN T, LUCAS S, POLONI C, BEUME N. Parallel Problem Solving from Nature – PPSN X – 10th International Conference. Berlin Heidelberg: Springer, 2008: 1081−1090.

[9] PIZZUTI C. Community detection in social networks with genetic algorithms [C]// KEIJZER M. GECCO'08: Preceedings of 10th Annual Conference on Genetic and Evolutionary Computation 2008. New York, NY, USA: ACM, 2008: 1137−1138.

[10] PIZZUTI C. A multi-objective genetic algorithm for community detection in networks [C]// ICTAI 2009 – 21st IEEE International Conference on Tools with Artificial Intelligence. Washington DC, USA: IEEE Computer Society, 2009: 379−386.

[11] SHI Chuan, YAN Zhen-yu, WANG Yi, CAI Ya-nan, WU Bin. A genetic algorithm for detecting communities in large-scale complex networks [J]. Advances in Complex Systems, 2010, 13(1): 3−17.

[12] CAI Yi-chao. Research on force aggregation technology in situation assessment [D]. Changsha: National University of Defense Technology, Information System and Management College, 2006. (in Chinese)

[13] LEICHT E A, HOLME P, NEWMAN M E J. Vertex similarity in networks [J]. Physical Review E, 2006, 73(2): 026120.

[14] GIRVAN M, NEWMAN M E J. Community structure in social and biological networks [J]. Proceedings of the National Academy of Sciences of the United States of America, 2002, 99(12): 7821−7826.

[15] NEWMAN M E J. Fast algorithm for detecting community structure in networks [J]. Physical Review E, 2004, 69(6): 066133.

[16] DANON L, DÍAZ-GUILERA A, DUCH J, ARENAS A. Comparing community structure identification [J]. Journal of Statistical Mechanics, 2005(9): P09008.

[17] RAGHAVAN U N, ALBERT R, KUMARA S. Near linear time algorithm to detect community structures in large-scale networks [J]. Physical Review E, 2007, 76(3): 036106.

[18] NEWMAN M E J. Network data [DB/OL]. 2011−06−14. http://www-personal.umich.edu/~mejn/netdata/.

(Edited by HE Yun-bin)