RESEARCH ARTICLE

A Genetic Algorithm with Local Search Strategy for Improved Detection of Community Structure

SHUZHUO LI,[1] YINGHUI CHEN,[1] HAIFENG DU,[1] AND MARCUS W. FELDMAN[1,2]

[1]Institute for Population and Development Studies; School of Management; Center for Administration and Complexity Science, Xi'an Jiaotong University, Xi'an, Shaanxi Province 710049, China; and [2]Morrison Institute for Population and Resource Studies, Stanford University, Stanford, California 94305

Received February 27, 2009; accepted June 30, 2009

On the basis of modularity optimization, a genetic algorithm is proposed to detect community structure in networks by defining a local search operator. The local search operator emphasizes two features: one is that the connected nodes in a network should be located in the same community, while the other is "local selection" inspired by the mechanisms of efficient message delivery underlying the small-world phenomenon. The results of community detection for some classic networks, such as Ucinet and Pajek networks, indicate that our algorithm achieves better community structure than other methodologies based on modularity optimization, such as the algorithms based on betweenness analysis, simulated annealing, or Tasgin and Bingol's genetic algorithm. © *2009 Wiley Periodicals, Inc. Complexity 15: 53–60, 2010*

Key Words: network structure; modularity; genetic algorithm; small-world phenomena

1. INTRODUCTION

Many real life networks are inhomogeneous; they do not consist of an undifferentiated mass of nodes but have some subgroup structure. Within these subgroups, the network connections are denser while between them the connections are sparser [1]. This characteristic of networks is called community structure, and its quantification has been one of the key objectives of network analysis in various domains such as sociology, biology, and computer science [1–8]. There also has been growing interest in algorithms for finding structures in networks. Detecting community structure in networks is an NP hard problem [9]. A network with n nodes has $\sum_{k=1}^{n} \frac{1}{k!} \sum_{j=1}^{k} \binom{k}{j} j^n$ different possible community structures [10], which makes it very difficult theoretically to identify an optimal assessment of structure.

Corresponding author: Marcus W. Feldman, Morrison Institute for Population and Resource Studies, Stanford University, Stanford, California 94305; e-mail: mfeldman@stanford.edu

DOI 10.1002/cplx.20300
Published online 25 November 2009 in Wiley InterScience (www.interscience.wiley.com)

Methods for detecting community structure can be classified as historical or current approaches [11]. Historical approaches include spectral partitioning and hierarchical clustering, while most of the current approaches are based on modularity. Modularity, defined by Girvan and Newman [2], has facilitated the development of current approaches and has become a cornerstone of research into community structure. In current approaches, two basic strategies have been used in community detection. One is a "top-down dividing" strategy, which takes the whole network as a single community and then divides this community into two smaller subgroups iteratively. The other is a "bottom-up merging" strategy, which treats each node as a community and then combines two small communities to form a bigger one. These two strategies take the detection of community structure based on modularity to be an optimization process for the objective function that defines modularity [8]. On the basis of modularity optimization, some artificial intelligence algorithms have been introduced to detect community structure in networks, such as simulated annealing suggested by physical phenomena [12], and genetic algorithms, inspired by biological phenomena [13]. These algorithms involve both "dividing" and "merging," so they are "mixed optimizing" strategies.

Detection of community structure may be a new application of genetic algorithms, which have many advantages for this problem such as parallel implementation, adaptive heuristic search, they use only the value of the objective function to optimize, and they do not require auxiliary knowledge of the problem. All of these advantages are very useful for complex or loosely defined problems. Of course, genetic algorithms have some shortcomings, such as premature convergence or becoming trapped at a local optimum [14], but these may be overcome by implementing local search methods.

We suggest a genetic algorithm to detect community structure in networks by defining a local search operator, which is inspired by the mechanisms of efficient message delivery underlying the small-world phenomenon. The article is organized as follows. In section 2, we review the definition of modularity in community structure. After defining the crossover, mutation, and local search operators, we represent the logic of our algorithm in section 3. In section 4, we demonstrate experimentally the effectiveness of our algorithm. Our conclusions are presented in section 5.

2. MODULARITY AND COMMUNITY STRUCTURE DETECTING

A network can be denoted as $G(V, \boldsymbol{A})$, where $V = \{v_1, v_2, \ldots v_n\}$ represents the set of n nodes in the network, and the connections between nodes are represented by an $n \times n$ adjacency matrix $\boldsymbol{A} = (a_{i,j})$ for $i, j = 1, 2, \ldots, n$. If i and j are connected, then the entry $a_{i,j} = 1$, otherwise $a_{i,j} = 0$. We assume there are no-self connections; that is, $a_{i,i} = 0$ for $i = 1, 2, \ldots, n$.

Community structure is defined by dividing V, the set of nodes, into m mutually exclusive subsets through an operator τ, i.e., $\tau(V) = \{V_1, V_2, \cdots, V_m\}, V_i \subset V, V_i \neq \phi$ for $i = 1, 2, \ldots, m$, and $\bigcup_{i=1}^{m} V_i = V$, $V_i \cap V_j = \phi$, $i \neq j$, where V_i represents community i. The connections within each community are relatively dense, while those between them are relatively sparse. For a given adjacency matrix $\boldsymbol{A}$, $\boldsymbol{A}_{p,q} = (a_{i,j})$, $i \in V_p$, $j \in V_q$, denotes the connection matrix between communities V_p and V_q. Let $N = \|\boldsymbol{A}\| = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{i,j}$ and $\|\boldsymbol{A}_{p,q}\| = \sum_{i\in V_p}\sum_{j\in V_q} a_{i,j}$, $a_{i,\,j} \in A$. If $p = q$, $\|\boldsymbol{A}_{p,q}\|$ denotes a number of equal to two times the number of connections within V_p or V_q. Otherwise, $\|\boldsymbol{A}_{p,q}\|$ equals the number of connections between nodes in V_p and nodes in V_q. Let $e_{p,q} = \frac{\|\boldsymbol{A}_{p,q}\|}{\|\boldsymbol{A}\|}$, $e_{p,p} = \frac{\|\boldsymbol{A}_{p,p}\|}{\|\boldsymbol{A}\|}$. Then detection of community structure can be considered as a process of maximizing $e_{p,p}$ and minimizing $e_{p,q}$ (where $p \neq q$) by adjusting the order of rows and columns simultaneously in the adjacency matrix $\boldsymbol{A}$. Thus, community detection does not change the network; the final result of detection is just another representation of the original network $G(V, \boldsymbol{A})$.

Girvan and Newman [2] defined the modularity Q as:

$$Q = \sum_{p=1}^{m}\left[e_{p,p} - \left(\sum_{q=1}^{m} e_{p,q}\right)^2\right], \tag{1}$$

where the first term $\sum_{p=1}^{m} e_{p,p}$ reflects the connections within communities and the second term $\sum_{p=1}^{m}\left(\sum_{q=1}^{m} e_{p,q}\right)^2$ describes the connections between any two communities. Apparently, the larger the value of $\sum_{p=1}^{m} e_{p,p}$ and the smaller the value of $\sum_{p=1}^{m}\left(\sum_{q=1}^{m} e_{p,q}\right)^2$, the stronger the community structure of the whole network, indicating more connections within communities and less connections between them. Q is one possible form for $\tau(V)$, defined earlier.

Let I be the space of all possible community structures of the network $G(V, \boldsymbol{A})$, and write $\tau(V) = \{l_1, l_2, \ldots l_n\} \in I$, be a point in space I. Here, the integer $l_i \in [1, n]$ is the label of the community to which v_i belongs. If v_i and v_j are located in the same community, then $l_i = l_j$, $\tau(V) = \{l_1, l_2, \ldots l_n\}$ and $\tau(V) = \{V_1, V_2, \ldots, V_m\}$ are two different but equivalent expressions for the community structure.

Our problem is to find a proper $\tau^*(V) \in I$ that maximizes modularity Q, i.e.,

$$\max Q(\tau(V)) \quad s.t. \quad \tau(V) = \{l_1, l_2, \cdots l_n\} \in I \tag{2}$$

DOI 10.1002/cplx

As the entry l_i in $\tau^*(V) = \{l_1, l_2, \ldots l_n\}$ is an integer, I is a discrete space. Therefore, detection of community structure in the sense of (2) is a typical combinatorial optimization problem.

3. A GENETIC ALGORITHM INSPIRED BY SMALL-WORLD PHENOMENA (GAS)

To detect community structure in network G using a genetic algorithm, we need first to define the fitness function and randomly generate g independent community structures to form an initial population, which is denoted by $\mathcal{T}$ with $\mathcal{T} = \{\tau_1(V), \tau_2(V), \cdots, \tau_g(V)\}$. Then the search for the optimum of the fitness function is carried out in space I through crossover and mutation, which, together with selection, are the three basic operations in genetic algorithms. As in Ref. [14], roulette wheel selection[1] is used here, and we take the modularity shown in formula (1) as the fitness function. We redefine the crossover operator using Ref. [13]. We introduce a new mutation operator and apply a local search operator using small-world properties to arrive at our algorithm.

3.1. Basic Operators

3.1.1. Crossover Operator

We randomly select two individuals, $\tau_i(V) = \{l_1^i, l_2^i, \cdots l_n^i\}$ and $\tau_j(V) = \{l_1^j, l_2^j, \cdots l_n^j\}$, from the population $\mathcal{T} = \{\tau_1(V), \tau_2(V), \cdots, \tau_g(V)\}$. Then, we randomly choose a community label as $l_r^i \in \tau_i(V)$ from individual i. Here l_r^i is the community label of node r in individual i. We iteratively search the nodes that have the same community label l_r^i and transfer the label l_r^i to the same nodes in individual j with crossover probability p_c, namely,

$$l_k^j = l_r^i \quad if\ k = r,\ k = 1, 2, \cdots, n \tag{3}$$

Figure 1 shows the implementation of crossover, in which the corresponding labels in $\tau_j(V)$ are replaced by "4" in $\tau_i(V)$.

FIGURE 1

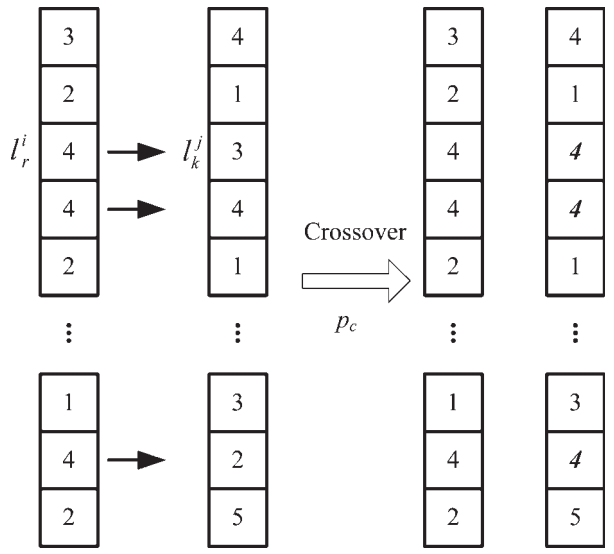


Illustration of crossover implementation.

3.1.2. Mutation Operator

The mutation operation $\mathcal{M}$ is defined as:

$$\mathcal{M}(\mathcal{T}) = \{\mathcal{M}(\tau_1(V)), \mathcal{M}(\tau_2(V)), \cdots, \mathcal{M}(\tau_g(V))\} \tag{4}$$

With probability p_m, $\tau_i(V) = \{l_1^i, l_2^i, \cdots l_n^i\}$ is randomly selected from $\mathcal{T} = \{\tau_1(V), \tau_2(V), \cdots, \tau_g(V)\}$. Then we randomly replace p's community label $l_p^i \in \tau_i(V)$ by q's community label $l_q^i \in \tau_i(V)$; namely, we combine community p and q to form a new community, where $l_p^i \neq l_q^i$.

3.1.3. Local Search Operator

Both crossover and mutation are random operations based on the relative probability in space I for community structure $\tau(V)$. Crossover, the defining operation for GAs, combines two individuals (parents) to produce a new individual (offspring). Generally, crossover changes the individuals much more than mutation and can be regarded as random long-distance search in Kleinberg's terminology [15, 16]. None of these take advantage of any prior knowledge about relations between nodes defined by the adjacency matrix $\boldsymbol{A}$. Usually the connected nodes in a network tend to be from the same community. For any community structure $\tau_i(V) \in \mathcal{T}$, we define a local search operator $\mathcal{L}^{n_l}(\mathcal{T})$ that makes n_l copies of $\tau_i(V)$. For each copy of $\tau_i(V)$, the directly connected nodes are randomly selected and placed into the same community based on the adjacency matrix $\boldsymbol{A}$. Then $\mathcal{L}^{n_l}(\mathcal{T})$ executes "local selection,"

[1] *Roulette-wheel selection is also called stochastic sampling with replacement. It involves the following technique: The individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness. A random number is generated and the individual whose segment spans the random number is selected. The process is repeated until the desired number of individuals (called the mating population) is obtained. (http://www.geatbx.com/docu/algindex-02.html#P363_18910).*

DOI 10.1002/cplx

which means that $\tau_i(V)$ is substituted by the best among those copies. The process of local search can be viewed as a message delivered from an initial candidate to a single acquaintance. It is described in the "six degrees of separation" (or "small world phenomena") experiments performed by the social psychologist Stanley Milgram in the 1960s [17].

The specific implementation of the local search operator $\mathcal{L}^{n_l}(\mathcal{T})$ is as follows:

Algorithm 1: local search $\mathcal{L}^{n_l}(\mathcal{T})$

Step 1: initiate the algorithm by $k = 1$, $\bar{\mathcal{T}} = \mathcal{T}$;

Step 2: if $k > g$, then the algorithm stops. Otherwise:

Step 3: $i = 1$; $\bar{\tau}_k(V) = \tau_k(V)$; $\tau'(V) = \tau_k(V)$;

Step 4: if $i > n_l$, go to Step 9, otherwise:

Step 5: randomly select row j from the adjacency matrix A, $a_j = \{a_{j,1}, a_{j,2}, \cdots, a_{j,n}\}$;

Step 6: if $a_{j,q} = 1$, for $q = 1, 2, \ldots, n$, let $\bar{l}_q^k = l_j^k$, where $\bar{l}_q^k \in \bar{\tau}_k(V)$, $l_j^k \in \tau_k(V)$, $q = 1, 2, \ldots, n$;

Step 7: if $Q(\bar{\tau}_k(V)) \geq Q(\tau'(V))$, let $\tau'(V) = \bar{\tau}_k(V)$;

Step 8: $i = i + 1$, return to step 4;

Step 9: if $Q(\tau'(V)) \geq Q(\tau_k(V))$, let $\tau_k(V) = \tau'(V)$;

Step 10: $k = k + 1$, return to step 2.

3.2. The General Steps

Incorporating the local search operator into the genetic algorithm produces a new algorithm for detecting optimal community structure. We denote the algorithm by "GAS."

Algorithm 2: GAS

Step 1: initialize parameters, such as the population size g, the length of the local search n_l, mutation probability p_m, crossover probability p_c, the number of optimal individuals retained at each iteration; randomly generate community structures to form the initial population $\mathcal{T} = \{\tau_1(V), \tau_2(V), \cdots, \tau_g(V)\}$;

Step 2: while the termination condition has not been met:

Step 3: save the best individual τ^* selected from $\mathcal{T} = \{\tau_1(V), \tau_2(V), \cdots, \tau_g(V)\}$;

Step 4: according to the value of Q, select g pairs of individuals from $\mathcal{T} = \{\tau_1(V), \tau_2(V), \cdots, \tau_g(V)\}$ and store them as a temporary population $\mathcal{T}'$ by roulette wheel selection; and use crossover with probability p_c for pairs of individuals in $\mathcal{T}'$ to form the new population $\mathcal{T}''$;

Step 5: implement mutation with probability p_m for individuals in $\mathcal{T}''$, i.e., $\mathcal{T}''' \leftarrow \mathcal{M}(\mathcal{T}'')$;

Step 6: generate the new population $\mathcal{T}$ using the local search operation for every individual in population $\mathcal{T}'''$, i.e., $\mathcal{T} \leftarrow \mathcal{L}^{n_l}(\mathcal{T}''')$;

Step 7: select the best individual τ^{**} from the new population $\mathcal{T}$. If the value of modularity for τ^* is greater than that for τ^{**}, then use τ^* to replace τ^{**}. Otherwise retain τ^{**} in the new population $\mathcal{T}'$;

Step 8: return to Step 2.

Generally, the termination condition can be defined by the maximum number of iterations N_i, or achieving no progress in terms of the best value found for N_c consecutive generations, which is set by the investigator. Finally we pick the solution with the best fitness from the final population:

$$Q_{\max}(\mathcal{T}) = Q(\tau_i(V)) \geq Q(\tau_j(V)) \quad \tau_i(V) \in \mathcal{T}, \tau_i(V) \in \mathcal{T}, \quad j = 1, 2, \cdots, g \qquad (5)$$

and $\tau_i(V)$ in (5) is the final community structure detected.

The multidimensional solution space I can be thought of as a network with each point $\tau(V) \in I$ acting as a node and a potential solution. The process by which our algorithm finds the optimal solution $\tau^*(V) \in I$ can be viewed as a message delivered from the initial candidate solutions $\mathcal{T} = \{\tau_1(V), \tau_2(V), \cdots, \tau_g(V)\}$ to $\tau^*(V) \in I$. Our algorithm is analogous to the small-world phenomenon, because the crossover and the mutation operator can be thought as the random long-range connections and the local search operator as finding local short connections in the solution space I.

4. APPLICATIONS

All the following experiments are run in Matlab 7.0 on a PC with IntelR CoreTM 2 2.00 GHz. The basic parameters are $g = 20$, $n_l = 4$, $p_m = 0.1$, $p_c = 0.8$, $N_i = 3000$, and $N_c = 100$. In this section, four other algorithms are compared with GAS: the first is Girvan and Newman's algorithm (denoted as G-N) based on betweenness analysis [2], the second is Medus et al.'s algorithm based on simulated annealing [12], the third the simple genetic algorithm based on [13] (denoted as TGA) and the fourth is Blondel et al.'s heuristic method without recursive computation [18] (denoted as BHM).

We compare the results obtained for Zachary Network, Les Misérables Network, and Bottlenose Dolphin Network by G-N, GAS, and the simulated annealing by Medus et al., where we also report the results of some Ucinet and Pajek networks [19,20] in which the algorithms GAS, TGA, G-N, and BHM are compared.

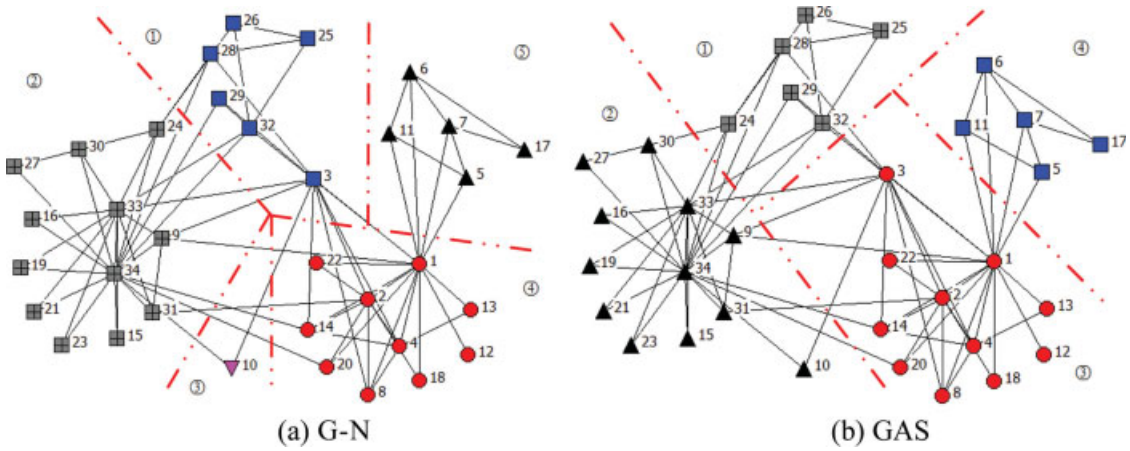
Results reported for our GAS and TGA are the best of the 40 runs for each since they are the probabilistic algorithms while G-N and BHM in each case are only for one run since they are the deterministic algorithms. All of the results for Medus et al.'s algorithm come from Ref. [12].

4.1. Zachary Network

Zachary karate club network is a symmetrical 0–1 network supplied by Ucinet software. Its size is 34, average degree is 4.588, and density is 0.139. Figure 2(a) represents the result of G-N algorithm with modularity $Q = 0.401$; Figure

DOI 10.1002/cplx

FIGURE 2



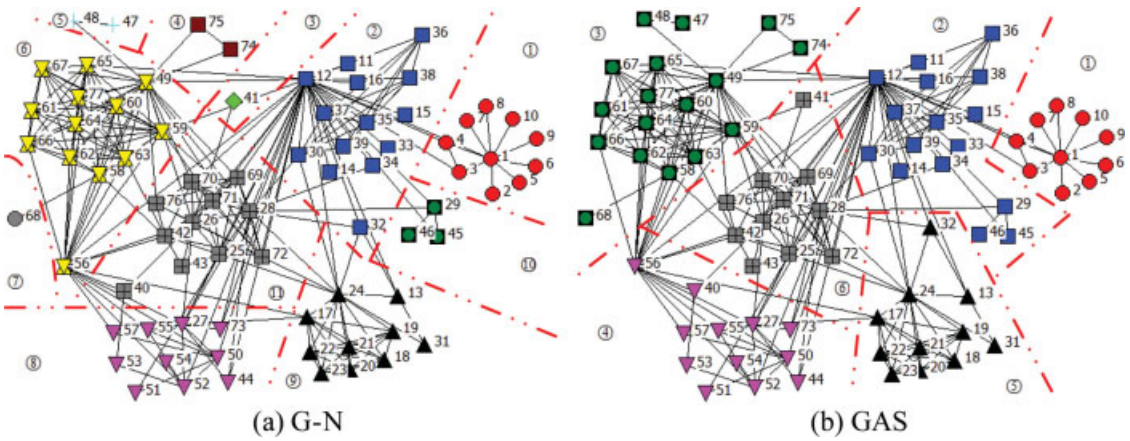
Community structures for Zachary karate club network. The dashdotted lines denote the partitions obtained the algorithms. The circled numbers are the labels of the communities. The numbers without circles are the labels of the nodes in the network. The different shapes with different colors, such as blue squares, red circles and so on, denote the members of each community.

2(b) shows the result of GAS with $Q = 0.420$, which is the same as that obtained with simulated annealing by Medus et al. [12]. Compared with the actual subgroups, community 1 and 2, community 3 and 4, respectively, correspond to two subgroups in (b), while in (a) the node 3 is incorrectly assigned to community 1. Moreover, it is unreasonable that node 10 alone is one community, because $e_{p,p} = 0$.

4.2. Les Misérables Network

Nodes in Les Misérables network are the main characters in the famous novel; those who appear in the same chapter or in more than one chapter are supposed to connect with each other in the network. Its size is 77, average degree is 6.597, and density is 0.087. The G-N algorithm finds eleven communities with modularity $Q = 0.538$ as shown in Figure 3(a), while with GAS we find six commun-

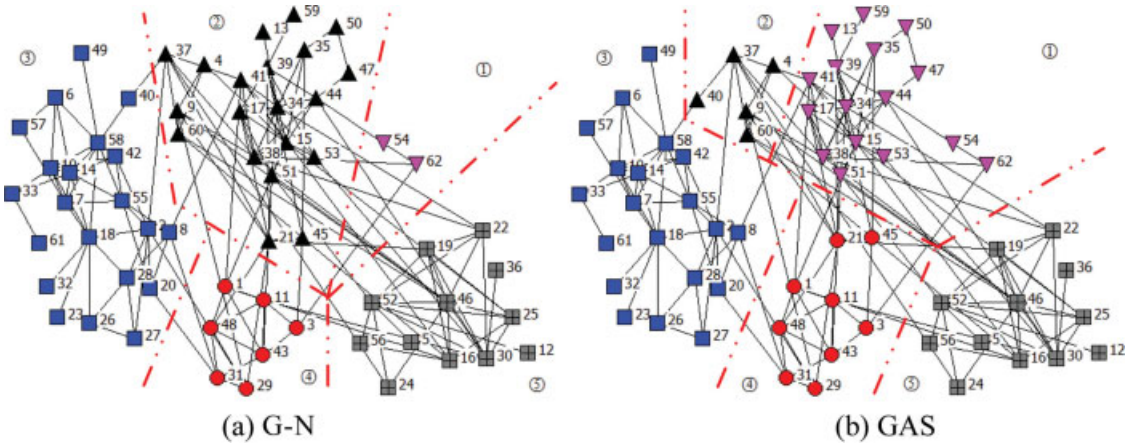
FIGURE 3



Community structures for Les Misérables network.

DOI 10.1002/cplx

FIGURE 4



Community structures for Bottlenose Dolphins network.

ities with $Q = 0.560$ as shown in Figure 3(b). We find nodes 41 and 68 each comprise one community with G-N, while with GAS they are merged into a bigger community. In Figure 3, except for nodes 32 and 56 which belong to different communities, we can obtain the result shown in Figure 3(b) by merging some individual communities in (a). The method of Medus et al. found five communities by simulated annealing, with $Q = 0.546$.

4.3. Bottlenose Dolphin Network

Bottlenose dolphin network records matings among 62 bottlenose dolphins. Its average degree is 5.129, and density is 0.084. As shown in Figure 4, although five communities are found by both G-N and GAS (see Figure 4), communities in Figure 4(a) are different from those in Figure 4(b) except for community 5. Modularity is $Q = 0.529$ by GAS and $Q = 0.519$ for G-N.

4.4. Pajec and Unient Networks

To further analyze the applicability of our GAS, we tested some standard networks available in the social network analysis software, Pajek and Ucinet. The parameters for these networks are shown in Table 1, where the original asymmetric networks are symmetrized.

Genetic algorithms have not commonly been used to detect community structure, and TGA is the earliest one [13]. Our GAS can be regarded as a modification of TGA, and we use TGA to compare with our GAS. For GAS and TGA based on 40 independent experiments we present four statistics, the maximum value, the minimum value, the mean, and the corresponding standard deviation of modularity Q, which are indicated by Q_{max}, Q_{min}, Q_{mean}, and Q_{std}, respectively, in GAS, and by Q'_{max}, Q'_{min}, Q'_{mean}, and Q'_{std} in the TGA. The termination condition for TGA is the maximum number of iterations N_i. We choose $N_i = 3000$ and the population size is $g = 60$ while the other parameters used in TGA, such as mutation and crossover probabilities, are the same as GAS.

Table 2 shows the results for three algorithms. For Ucinet, the highest modularity is obtained by using the G-N algorithm and BHM algorithm. The speed of BHM is the

TABLE 1

Parameters for the Networks in Ucinet and Pajek

Networks	Size	Average Degree	Density	Source
Drugnet	293	1.939	0.007	Ucinet
1crn	327	2.061	0.006	Pajek
ADF073	262	2.046	0.008	Pajek
B	111	3.478	0.032	Pajek
BKHAM	44	4.046	0.094	Pajek
BKOFF	40	6.150	0.158	Pajek
c	65	3.846	0.060	Pajek
cc	62	4.645	0.076	Pajek
CENPROD	131	9.603	0.074	Pajek
Dnet	180	2.533	0.014	Pajek
GR3_53	144	5.000	0.035	Pajek
GR3_60	120	3.000	0.025	Pajek
KAPTAIL	39	8.103	0.213	Pajek
MREZA3	144	3.667	0.026	Pajek
nooy	85	1.906	0.023	Pajek

DOI 10.1002/cplx

TABLE 2

The Detecting Results of Three Algorithms

	G-N	BHM	GAS				TGA			
Networks	$Q_{\text{G-N}}$	Q_{BHM}	$Q_{\min}$	Q_{mean}	$Q_{\max}$	Q_{std}	$Q'_{\min}$	Q'_{mean}	$Q'_{\max}$	Q'_{std}
Drugnet	0.737	0.545	0.717	0.726	0.742	7.862×10^{-3}	0.486	0.532	0.576	1.918×10^{-2}
1crn	0.874	0.527	0.866	0.872	0.881	6.289×10^{-3}	0.485	0.548	0.587	2.183×10^{-2}
ADF073	0.883	0.490	0.865	0.873	0.882	5.263×10^{-3}	0.489	0.543	0.583	1.944×10^{-2}
B	0.625	0.561	0.610	0.624	0.630	5.786×10^{-3}	4.967×10^{-4}	0.132	0.585	2.410×10^{-1}
BKHAM	−0.002	0.179	0.180	0.199	0.207	1.103×10^{-2}	6.123×10^{-3}	0.135	0.203	8.312×10^{-2}
BKOFF	0.338	0.327	0.353	0.365	0.375	7.924×10^{-3}	7.304×10^{-3}	0.330	0.370	$9{,}349 \times 10^{-2}$
C	0.547	0.376	0.580	0.584	0.585	1.559×10^{-3}	4.896×10^{-3}	0.241	0.525	2.370×10^{-1}
Cc	0.547	0.376	0.578	0.584	0.585	2.141×10^{-3}	5.312×10^{-3}	0.276	0.510	2.350×10^{-1}
CENPROD	0.041	0.271	0.264	0.277	0.295	1.070×10^{-2}	9.099×10^{-3}	0.043	0.296	1.010×10^{-1}
Dnet	0.600	0.468	0.610	0.641	0.651	1.258×10^{-2}	0.440	0.465	0.486	1.090×10^{-2}
GR3_53	0.649	0.508	0.700	0.709	0.717	6.521×10^{-3}	9.066×10^{-4}	2.293×10^{-3}	6.269×10^{-3}	1.197×10^{-3}
GR3_60	0.658	0.317	0.679	0.692	0.696	4.894×10^{-3}	2.083×10^{-3}	3.601×10^{-3}	7.778×10^{-3}	2.080×10^{-3}
KAPTAIL	0.227	0.316	0.315	0.320	0.321	1.918×10^{-3}	5.408×10^{-3}	0.264	0.321	1.090×10^{-1}
MREZA3	0.665	0.259	0.685	0.693	0.701	5.416×10^{-2}	1.722×10^{-4}	1.176×10^{-3}	3.917×10^{-3}	1.429×10^{-3}
Nooy	0.808	0.798	0.808	0.808	0.808	1.655×10^{-16}	0.788	0.797	0.798	1.662×10^{-3}

All of the modularity values $Q_{\text{G-N}}$ with G-N are obtained by the Ucinet software.

fastest among these four algorithms. For 14 of 15 networks, we see $Q_{\max} > Q_{\text{G-N}}$. Only for ADF073 is the modularity with GAS smaller than that with G-N, but they are very close. All the modularity values for $Q_{\max}$ with GAS are significantly higher than those of $Q'_{\max}$ with TGA. Thus, to some extent GAS avoids the premature convergence and trapping in local optima of TGA. Irrespective of the randomness of the algorithms, we have $Q_{\min} > Q_{\text{G-N}}$ for 11 networks. We are currently working on different schemes (such as inspired from BHM) to improve the efficiency of the calculation of our algorithm.

5. CONCLUSIONS

The algorithm we propose for finding community structure, GAS, is based on crossover, mutation, and local search and extends the work of Kleinberg [15, 16], which, as he explained, was inspired by small world phenomena. The effectiveness of our algorithm is demonstrated by experiments with networks, and compared TGA, higher values for modularity can be obtained in GAS. In comparison with G-N, our algorithm performs better except in large networks.

If we ignore the context in which the specific problems arise and do not properly incorporate prior knowledge, TGA will not perform very well, although it has the advantages of problem independence and strong flexibility. GAS combines local and long distance search and achieves better performance in the experiments tested.

We have compared the accuracy among the GAS, TGA, G-N and the simulated annealing method of Medus et al. for the other issue relevant to this kind of analysis, namely speed. As with simulated annealing approaches, the genetic algorithms are random search strategies. Hence, they are intrinsically slow. As pointed in Ref. [13], TGA has $O(n^2)$ time-complexity, where n is the size of the network. The time-complexity of our GAS is a little greater than TGA, but no more than $O(n^2 n_l)$, because the local search operator $\mathcal{L}^{n_l}(\mathcal{T})$ copies the population $\mathcal{T} = \{\tau_1(V), \tau_2(V), \cdots, \tau_g(V)\}$ n_l times. For the time being, we are restricted to networks of size less than 400, and it remains to improve the search ability of GAS for application to large networks.

Acknowledgments

This work is jointly supported by the National Natural Science Foundation of China (70671083), Changjiang Scholar Awarding Program by Ministry of Education, Program for Changjiang Scholars and Innovative Research Team in University (IRT0855), New Century Excellent Talents (NCET-07-0668, NCET-08-0451), Morrison Institute for Population and Resource Studies at Stanford University, Santa Fe Institute International Program, and "985 Project" National Social Science Research Base of The State Education Committee.

DOI 10.1002/cplx

REFERENCES

1. Newman, M.E.J.; Barabàsi, A.L.; Watts, D.J. The Structure and Dynamic of Networks; Princeton University Press: Princeton, NJ, 2006.
2. Girvan, M.; Newman, M.E.J. Community structure in social and biological networks. Proc Natl Acad Sci USA 2002, 99, 7821–7826.
3. Eckmann, J.; Moses, P.E. Curvature of co-links uncover hidden thematic layers in the World Wide Web. Proc Natl Acad Sci USA 2002, 99, 5825–5829.
4. Eriksen, K.A.; Simonsen, I.; Maslov, S.; Sneppen, K. Modularity and extreme edges of the internet. Phys Rev Lett 2003, 90, 148701.
5. Holme, P.; Huss, M.; Jeong, H. Subnetwork hierarchies of biochemical pathways. Bioinformatics 2003, 19, 532–538.
6. Arenas, A.; Danon, L.; Diaz-Guilera, A.; Gleiser, P.M.; Guimerà, R. Community analysis in social networks. Eur Phys J B 2004, 38, 373–380.
7. Massen, C.P.; Doye, J.P.K. Identifying communities within energy landscapes. Phys Rev E 2005, 71, 046101.
8. Newman, M.E.J. Modularity and community structure in networks. Proc Natl Acad Sci USA 2006, 103, 8577–8582.
9. Brandes, U.; Delling, D.; Gaertler, M.; Goerke, R.; Hoefer, M.; Nikoloski, Z.; Wagner, D. Maximizing modularity is hard. 2006. arXiv:physics/0608255v2
10. Richard, O.D.; Peter, E.H.; David, G.S. Pattern Classification, 2nd ed.; Wiley: New York, 2001.
11. Newman, M.E.J. Detecting community structure in networks. Eur Phys J B 2004a, 38, 321–330.
12. Medus, A.; Acuna, G.; Dorso, C.O. Detection of community structures in networks via global optimization. Physic A 2005, 358, 593–604.
13. Tasgin, M.; Bingol, H. Community Detection in Complex Networks using Genetic Algorithm. 2006. arXiv:cond-mat/0604419v1.
14. Michalewicz, Z. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag Berlin Heidelberg: Berlin, 1996
15. Kleinberg, J. Navigation in a small world. Nature 2000, 406, 845.
16. Kleinberg, J. The small-world phenomenon and decentralized search. SIAM News, 2004, 37(3).
17. Watts, D.J.; Strogatz, S.H. Collective dynamics of small-world networks. Nature 1998, 393, 440–442.
18. Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte and Etienne Lefebvre. Fast unfolding of communities in large networks. 2008. arXiv:0803.0476v2.
19. Download UCINET for Windows, Version 6 Software for Social Network Analysis. Available at: http://www.analytictech.com/downloaduc6.htm. Accessed April 26, 2005.
20. Pajek Program for Large Network Analysis. Available at: http://vlado.fmf.uni-lj.si/pub/networks/pajek/default.htm. Accessed April 26, 2005.

DOI 10.1002/cplx