Complex Network Community Detection based on Genetic Algorithm using K-cliques

Jian Ma1,2, Jianping Fan1,3
1School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China
2Research Center for High-Speed Railway Network Management of Ministry of Education, Beijing Jiaotong University, Beijing 100044, China
3Shenzhen Institutes of Advanced Technology, Shenzhen 518055, China

13112083@bjtu.edu.cn

Abstract. Inspired by the genetic algorithm, this paper proposes a complex network community detection algorithm, which searches for complete sub-graphs in a network, the numbers of nodes are greater than or equal to k, also referred to maximum k-cliques. K-cliques are the most connected communities in a network. Using the maximum k-clique initializes the population, which can improve the accuracy and efficiency of population initialization. K-clique-based population initialization, crossover, mutation, μ+λ selection strategy, and evaluation function (Q function) are adopted to select the next generation of population, the superior traits formed by parents during the process of evolution cannot be destroyed and can also be effectively inherited by offspring individuals. Finally, community partition is optimized using the fast Newman algorithm. This step performs further clustering on the communities. The algorithm can reduce the search space of community partition and improve the search efficiency of the algorithm. It is tested on benchmark networks and real-world networks. The algorithm has an acceptable time complexity. The experimental results show that the algorithm can effectively divide the communities.

1. Introduction
In a sense, complex networks are made up of different communities. The problem of complex network clustering (community mining) is how to reveal the real communities in which the internal nodes are densely connected, and the connections between different community nodes are sparse. In recent years, many researchers have proposed many classic community structure detection algorithms from different perspectives.

The genetic algorithm is a method for simulating natural evolution process and searching for optimal solution. As an adaptive global optimization probability search algorithm, the genetic algorithm is applied to community detection problems and is an effective means for solving complex optimization problems. Therefore, a complex network community detection method based on the genetic algorithm has attracted the attention of scholars. Tasgin et al. [1] first introduced the genetic algorithm for solving complex network community detection problems. The advantage was that the algorithm did not require prior knowledge, such as the number of communities. The disadvantage was that the algorithm searched entire solution space, rather than the neighboring nodes, when searching for community labels, it would lead to a meaningless search space with slow convergence. Yang et al. [2]...
proposed an improved Genetic Algorithm based on local modularity (IGALM). Hafez et al. [3] proposed a novel multi-objective optimization problem for the genetic algorithm in solving community detection problems. Li et al. [4] proposed MAGA-Net, which was based on the genetic algorithm, each agent represented a community partition, and different operators for crossover and mutation were proposed, the purpose was to increase community partition evaluation values. The k-path algorithm was used to obtain the topology of a network, applying the k-path algorithm in initialization phase could speed up the convergence of the algorithm and increase the modularity function for community structure detection [5]. Guerrero et al. [6] proposed a novel genetic algorithm called GGA+. The algorithm analyzed a network from different perspectives, optimized the operator of the genetic algorithm based on modularity function, and obtained relatively good experimental results compared with those by other algorithms. Due to the limitation of the algorithm itself. Liu et al. [7] proposed a genetic method using node-to-community function mutation. Inspired by the genetic algorithm, this paper proposes a complex network community detection algorithm based on maximum k-cliques (GAC algorithm). The algorithm, referred to the GAC algorithm, uses maximum k-cliques to improve the genetic representation of the genetic algorithm and continues to use the maximum k-cliques during crossover and mutation phases. Finally, the test is carried out on benchmark networks and real-world networks. The experimental results show that the algorithm can effectively divide communities.

2. Genetic algorithm

2.1. Gene expression

A genetic algorithm (GA) evolves to the optimal solution by simulating selection, crossover, and mutation of biological inheritance. The genetic algorithm begins with a population that represents a possible solution set of problems, and the population consists of a certain number of genetically encoded individuals. By encoding, the solution to the problem can be transformed into a chromosome or an individual.

The encoding methods currently applied to GA-based community mining algorithms include string-based encoding and locus-based adjacency representation. This paper uses the former. Let the two-tuple $G=(N,E)$ denote a complex network, where $V=\{v_1,v_2,...,v_m\}$ represents a node set, and $E=\{e_1,e_2,...,e_n\}$ represents an edge set. $A=(a_{ij})_{mn}$ represents the adjacency matrix of the network; if node $v_i$ and $v_j$ are adjacent, $a_{ij}=1$; otherwise, $a_{ij}=0$.

$$\kappa(t)=[\kappa_1,\kappa_2,\ldots,\kappa_n]$$

(1)

where $\kappa_i$ indicates the community label to which node $i$ belongs; if $\kappa_i=\kappa_j$, node $i$ and node $j$ are in the same community; otherwise, node $i$ and node $j$ belong to different communities.

**Figure 1.** An example of community structures
This paper adopts string-based encoding. Each individual (chromosome) in the population contains \( n \) gene positions, 1...\( n \), where \( n \) is the number of nodes in the network. In Figure 1, the community label of each node is represented as in Equation (1). Let there be 7 nodes in the network, and the chromosome is initialized as shown in Table 1.

**Table 1.** Gene expression

| \( v \) | individual (chromosome) |
|-------|-------------------------|
| \( v_1 \) | 1                       |
| \( v_2 \) | 2                       |
| \( v_3 \) | 3                       |
| \( v_4 \) | 4                       |
| \( v_5 \) | 5                       |
| \( v_6 \) | 6                       |
| \( v_7 \) | 7                       |

The \( k \)-clique is the most closely connected community in a network. The algorithm searches for all complete sub-graphs which the numbers of all nodes are greater than or equal to \( k \), that is, maximal \( k \)-cliques. In this paper, the maximal \( k \)-cliques generated by the above steps are to initialize the population. The algorithm generates a chromosome using the following steps: Initialize the community label of each node in the network as a unique label. When the algorithm generates an individual, we randomly select a node and seek any maximal clique containing this node, If all the nodes in this selected clique are not assigned to any community, that is, the community labels of the nodes are still at their initial labels, then update the community labels corresponding to all nodes in the clique with the minimum initialized community label among all the nodes in this clique. The reason for this is that nodes in the same clique are more likely to be in the same community. If the node does not belong to any maximal clique, the algorithm does nothing. We use the same operation \( k \times n \) times. The process forms many small local communities, but it is still far from optimal solution.

The maximal \( k \)-clique initializes the population, which can improve the accuracy and efficiency of population initialization, nodes belonging to the same community are assigned together. For example, the network in Table 2 has 7 nodes. A maximal 3-clique is a complete sub-graph which the number of nodes is greater than or equal to 3. The nodes are randomly arranged, for example, in the order \( v_5, v_2, v_6, v_1, v_3, v_7 \). We select the node \( v_5 \), there are three cliques containing node \( v_5 \), which are \{\( v_4, v_5, v_6 \}\}, \{\( v_2, v_6, v_7 \}\}, \{\( v_4, v_5, v_6, v_7 \}\}, the algorithm arbitrarily selects one, for example, \{\( v_4, v_5, v_6 \}\}, then update the community labels of nodes \( v_4, v_5, v_6 \), and \( v_6 \) to the minimum label of the nodes in clique \{\( v_4, v_5, v_6 \}\}, which is 4.

**Table 2.** Population initialization

| \( v \) | maximal 3-clique |
|-------|-----------------|
| \( v_1 \) | \( \{v_1,v_2,v_3,v_4\} \) |
| \( v_2 \) | \( \{v_1,v_2,v_3,v_4\} \) |
| \( v_3 \) | \( \{v_1,v_2,v_3,v_4\} \) |
| \( v_4 \) | \( \{v_1,v_2,v_3,v_4\},\{v_4,v_5,v_6,v_7\} \) |
| \( v_5 \) | \( \{v_2,v_6,v_7\},\{v_4,v_5,v_6,v_7\} \) |
| \( v_6 \) | \( \{v_4,v_5,v_6\},\{v_4,v_5,v_6,v_7\} \) |
| \( v_7 \) | \( \{v_5,v_6,v_7\},\{v_4,v_5,v_6,v_7\} \) |

### 2.2. Selection operator

The algorithm adopts a traversal sampling method to implement the survival of individuals in the population. In this paper, the selection strategy \( \mu+\lambda \) is adopted, which the size of the parent population is \( \mu \). After crossover and mutation, the sub-population size is \( \mu+\lambda \), from which \( \mu \) individuals with the greatest fitness function (Q function) [8] are selected to be the next-generation parents.
2.3. Crossover operation
The crossover operation of the genetic algorithm in this paper does not simply exchange some gene positions of the parents, this operation could not only produce excellent individuals but destroy the good community structures that has been formed. Therefore, in the process of crossover, the algorithm adopts the single-way crossover strategy. Firstly, select two individuals, one as the source individual called \( s_{\text{src}} \), and the other as the target individual named \( s_{\text{dest}} \). Randomly select node \( v_j \) belonging to two or more maximal k-cliques, find the nodes that are in the same community as \( v_j \) in the source individual \( s_{\text{src}} \), then update the corresponding node community labels in the target individual \( s_{\text{dest}} \) to the community label of node \( v_j \), thereby generating an individual \( s_{\text{new}} \). If the above node \( v_j \) does not exist, the algorithm does nothing.

For example, Table 3 shows the single-way cross operation, node \( v_4 \) which belongs to three maximal k-cliques is selected. In the source individual, nodes \( v_1, v_4 \) are in the same community. After one crossover operation, the community labels of the corresponding nodes in the new individual are all updated to 1.

| \( v \) | \( s_{\text{src}} \) | \( s_{\text{dest}} \) | \( s_{\text{new}} \) |
|---|---|---|---|
| \( v_1 \) | 1 | 1 | 1 |
| \( v_2 \) | 3 | 2 | 2 |
| \( v_3 \) | 3 | 2 | 2 |
| \( v_4 \) | 1 | 4 | 1 |
| \( v_5 \) | 7 | 5 | 5 |
| \( v_6 \) | 2 | 5 | 5 |
| \( v_7 \) | 3 | 2 | 2 |

2.4. Mutation operator
The mutation operation adopts basic position mutation. The source individual \( s_{\text{src}} \) randomly assigns a certain gene position or a certain number of genes to the mutation operation. Firstly, find the nodes that do not belong to any maximum k-clique, randomly pick node \( v_j \) from these nodes, obtain the collection of nodes adjacent to this node, then randomly select a neighboring node, and update the community label of node \( v_j \) to the label of this node, the rest remain unchanged. If the above node \( v_j \) does not exist, the algorithm does nothing. The purpose is to solve the isolated node that does not belong to any maximal k-cliques.

2.5. Optimization operation
This paper combines the communities formed by the above steps and finds the maximum Q value to increase. Starting with an origin network which each community is formed by the above steps as a sole member and no edge, then repeatedly join communities together in pairs, at each step choose the greatest increase value in Q. “dendrogram”, tree-like diagram that shows the order of the nodes join. Cut through this dendrogram at different levels to give divisions of the network, the network will be split into several communities and the partition corresponding to the largest Q value is the most accurate partition. The optimization operation is the further clustering operation of the communities.

2.6. Algorithm Description
In summary, the algorithm is described as follows:

Algorithm 1: Initiation
Input: maximum k-cliques /* maximum k-cliques is a set of all maximum k-cliques*/
Output: \( P \) /* population \( P \)*/
1. for \( i=1: \alpha \times n \)
2. randomly select a node \( v_j \);
3. if exists maximal k-clique mc contains this node vj and all the nodes in mc are their initial labels;
4. update the gene positions corresponding to all nodes in mc with the minimum community label of all nodes in mc;
5. end if
6. end for

Algorithm 2: crossover
Input: P
Output: P
1. randomly select a chromosome g1 from P;
2. randomly select a node vi which belongs to more than one maximal k-cliques, find the node set vs in the same community as node vi in g1;
3. randomly select a chromosome g2 from P;
4. update vs’s community labels in g2 to the community label of node vi;

Algorithm 3: mutation
Input: P
Output: P
1. randomly select a chromosome g1 from P;
2. randomly select a node vi which do not belong to any maximal k-cliques, obtain the community labels of node set vs linking to vi;
3. update vi’s labels to any of the community label of node set vs;

Algorithm 4:
1. P = Initiation(maximum k-cliques) /*population initialization, the number is μ*/
2. for i = 1:NG /*NG is the number of iterations*/
3. Pnew = ∅
4. for j = 1:λ
5. g = choose two individuals from P; /*crossover*/
6. g’ = choose one individual from P; /*mutation*/
7. Pnew = Pnew ∪ g ∪ g’;
8. end for
9. P = P ∪ Pnew;
10. the greatest fitness μ are selected from P to P, the next-generation is P; /*μ+λ*/
11. end for
12. bestv = max(P); /*bestv is the maximal fitness individual*/
13. cluster = fast newman(bestv); /*optimization operation, cluster is the community with maximum Q value */

Figure 2 is a flow chart of the GAC algorithm, it reveals the whole process of the GAC algorithm and outlines Algorithm 1 to Algorithm 4. This gives an intuitive idea of our algorithm. The GAC algorithm starts from population initialization, then calculate Q value, selection operation, crossover operation, mutation operation, finally, if it is greater than the number of iterations, the optimization operation is performed, otherwise the iteration continues.
Figure 2. A flow chart of the GAC algorithm

2.7. Algorithm complexity analysis
The recursive algorithm is adopted to solve maximum k-cliques. Let \( t \) is the number of maximum k-cliques in a network, the network has \( n \) nodes and \( m \) edges, and the average degree of all nodes in the network is \( d \). The time complexity of algorithm is \( O(n^*d) \).

The time complexity of population initialization is \( O(\alpha^*n^*\mu) \), \( n \) is the number of nodes. The number of operations in the genetic algorithm is \( \lambda \) and the number of iterations is \( NG \), the time complexity of the whole iterative process is \( O(NG^*\lambda) \). If there are a total of \( c \) communities after decoding, the time complexity of the optimization operation scales as \( O(c^2) \).

3. Experiment and analysis

3.1. GN benchmark network
The GN artificial network was proposed by Girven and Newman. According to the artificial data sets generated by computer program, there are 128 nodes in the network; each community has 32 nodes, and there are 4 communities in total. Each node in the network is connected to \( z_{in} \) nodes in the same community and to \( z_{out} \) nodes of other communities. The sum of \( z_{in} \) and \( z_{out} \) is 16. The larger \( z_{in} \), the more obvious the community structures. Otherwise, there will be more obstacles in the community structures.

Figure 3 shows a comparison of GAC with IGALM [2], Ref. [7] and MAGA-Net [4] algorithms; the x-axis represents \( z_{out} \), the y-axis represents Normalized Mutual Information (NMI), and the experimental data are the average of 20 runs of the algorithm. The experimental results show that good experimental results can be achieved. Figure 3 shows that the algorithm has good clustering accuracy.
Figure 3. Comparison of clustering results of GN networks
Figure 3 shows that when \( z_{out} \) is less than or equal to 0.4, the algorithm can correctly detect the real community structures in the network. When \( z_{out} \) is greater than 0.4, the community structures of the network becomes an obstacle, and the GAC algorithm still has a higher community detection accuracy than those of other algorithms, which shows that the algorithm is feasible and can well detect community structures.

3.2. LFR benchmark network
The LFR benchmark network model [9] was proposed by Lancichinetti et al. It can simulate the statistical characteristics of real complex networks and has been widely used in testing network community detection algorithms. The parameter settings of the benchmark network are: LFR(\( N, k, k_{max}, c_{min}, c_{max}, t_1, t_2, \mu, \sigma, \alpha \)). Table 4 shows the meanings of the parameters.

Table 4. The meaning of the parameters

| parameters   | meaning                        |
|--------------|--------------------------------|
| \( N \)      | the number of nodes            |
| \( k \)      | the average degree of the nodes|
| \( k_{max} \)| nodes’ maximum degree         |
| \( c_{min} \)| nodes’ minimum cluster size   |
| \( c_{max} \)| nodes’ maximum cluster size   |
| \( t_1 \)    | the exponent of the nodes degree distribution |
| \( t_2 \)    | the exponent of the cluster distribution |
| \( \mu \)    | the mixed parameter           |
| \( \sigma \) | the number of overlapping nodes’ communities |
| \( \alpha \) | the number of multiple communities’ nodes |

Figure 4. Comparison of clustering results of LFR benchmark networks
In the experiment, we used the network parameters LFR \((n = 1000, k = 15, k_{max} = 50, c_{min} = 20, c_{max} = 50, t_1 = 2, t_2 = 1, mu = 0.1–0.6)\).

Figure 4 shows the experimental results of the comparison of GAC and IGALM, Ref. [7] and MAGA-Net algorithms. The x-axis represents \(mu\), the y-axis represents NMI, and the experimental data are the average of 20 runs of the algorithm. It can be seen that the algorithm has good clustering accuracy. It can be seen from Fig. 4 that when \(mu\) is less than or equal to 0.5, the algorithm can correctly detect the real community structures in the network, and when \(mu\) is greater than 0.5, the community structures of the network become blurred, but the GAC algorithm can still correctly divide 95.69% of the nodes.

![Figure 4](image)

**Figure 5.** Comparison of NMI of LFR benchmark networks

Figure 5 (a) and (b) are the changes of NMI of four overlapping community detection algorithms with the increase of mixed parameter \(mu\) in the LFR \((N=1000/5000, k = 20, k_{max} = 50, t_1 = 2, t_2 =1, om= 0, om = 1, c_{min}=10, c_{max}=50)\). The network size \(N\) and the distribution interval of community \([c_{min}, c_{max}]\) also affect the community discovery algorithm results, Figure 5 (c) and (d) are the experimental results of parameters LFR \((N=1000/5000, k = 40, k_{max} = 100, t_1 = 2, t_2 =1, om= 0, om = 1, c_{min}=20, c_{max}=100)\). It can be concluded from the experimental results that GAC achieves higher detection accuracy than the other three algorithms with different network sizes and distribution intervals.
3.3. Running time

![Figure 6. GAC algorithm runtime](image)

This test uses the network parameter LFR \((n=1000-5000, k=15, k_{\text{max}}=50, c_{\text{min}}=20, c_{\text{max}}=50, t_1=2, t_2=1, \mu=0.6, \omega=0, \phi=1)\). Figure 6 shows the algorithm runtime as the number of nodes increases, and the running time of the GAC algorithm is within an acceptable range.

3.4. Parameter analysis

![Figure 7. Comparison of clustering results of LFR network with different k values](image)

The parameters of the algorithm are \(\alpha\), the population size \(\mu\), the number of iterations \(NG\), and the \(k\) in k-clique. The first four parameters are easy to set. In the text, \(\alpha\) is 0.2, \(NG\) is taken as 20, \(\mu\) and \(\lambda\) are 100. For the parameter \(k\), we use the networks LFR \((n=1000, k=15, k_{\text{max}}=50, c_{\text{min}}=20, c_{\text{max}}=50, t_1=2, t_2=1, \mu=0.6, \omega=0, \phi=1)\). Figure 7 shows that when the value of \(k\) is 3 or 4, you can get better results.

3.5. Real world network

We test the algorithm on the real-world networks. The data used in the experiment are as follows: the Karate club network provided by Newman [10], Dolphin social network [11], American College football, Word adjacencies, Books about US politics, Jazz musicians network, and Email network of human interactions. Figures 8 and 9 show the clustering results for the real networks Karate club and Dolphins, respectively, with Fig. 10 listing the different shapes and representing different community divisions. Table 5 shows seven real network datasets.
Table 5. Real networks used in the experiment

| NO. | networks   | node | edge | communities |
|-----|------------|------|------|-------------|
| 1   | karate     | 34   | 78   | 2           |
| 2   | Dolphins   | 62   | 159  | 2           |
| 3   | Football   | 115  | 613  | 12          |
| 4   | adjnoun    | 112  | 425  | 2           |
| 5   | Political books | 105 | 441  | 3           |
| 6   | jazz       | 198  | 5484 | 4           |
| 7   | email      | 1133 | 5451 | 11          |

Table 6. Comparison of Q value of real world networks

| Q-value      | MAGA-Net | Ref.[7] | IGALM | GAC |
|--------------|----------|---------|-------|-----|
| Zachary’s club | 0.4194  | 0.4183  | 0.4179 | 0.4174 |
| Dolphins      | 0.5286  | 0.5264  | 0.5163 | 0.5268 |
| Football      | 0.6046  | 0.6046  | 0.5986 | 0.6046 |
| adjnoun       | 0.2940  | 0.2851  | 0.2851 | 0.2945 |
| Political books | 0.5273  | 0.5264  | 0.5231 | 0.5268 |
| jazz          | 0.4448  | 0.4431  | 0.4438 | 0.4433 |
| email         | 0.5115  | 0.5246  | 0.5168 | 0.5210 |

Table 6 shows the Q function values for different real-world networks. With the Karate club, the Dolphin social network, and the Jazz musician network, the GAC algorithm is higher-performing than the Ref.[7]. With American College football and Word adjacencies, the GAC algorithm has the highest Q value. Through analysis and comparison, the GAC algorithm can find small communities that exist in the network and can effectively divide the network.

Figure 8 shows the network partition obtained by the GAC algorithm applied to the Karate club network. The algorithm divides the network into four communities. The circular nodes and square nodes is one part, and the triangle nodes and diamond nodes is the other parts. The combination of the circular nodes and square nodes, and the combination of the triangle nodes and diamond nodes are the correct division of this network.

The phenomenon suggests that the GAC algorithm can uncover smaller, more compact communities in communities.
Figure 9. Communities structure of Dolphins obtained by the GAC

Figure 9 shows the network partition obtained by applying the GAC algorithm to the Dolphins network. The algorithm divides the network into four communities. The nodes represented by the triangles are consistent with the correct division of the network, and the combination of the rest corresponds to the proper partition of the network. The results of the partitioning also show that the algorithm can find a tightly connected community structure.

Figure 10. Communities structure of football obtained by the GAC

Figure 10 shows the network partition obtained by applying the GAC algorithm to the football network. The algorithm divides the network into 10 communities. Different communities are represented by different node markers, and the GAC algorithm correctly divides 87% of the community structures.
4. Conclusions
This paper proposes the GAC algorithm, a k-clique complex network community algorithm based on the genetic algorithm which uses maximum k-cliques for genetic evolution selection, crossover, and mutation operations. It does not require prior knowledge and reduces the search space for community partition. The algorithm improves the search efficiency and is tested on benchmark networks and real-world networks. The experimental structure shows that the algorithm can effectively divide the community. This algorithm can also be applied broadly to clustering problems in many others domains.

Acknowledgments
The authors would like to acknowledge the funding support of the National Key R&D Program of China under Grant 2016YFB1200100.

References
[1] M TASGIN, A HERDAGDELEN, H BINGOL. Community Detection in Complex Networks Using Genetic Algorithms. eprint, arXiv:0711.0491, 2007.
[2] X Yang, R Li. An Improved Genetic Algorithm Based on Local Modularity for Community Detection in Complex Network, International Conference on Information Science & Cloud Computing Companion. 2014.
[3] A I HAFEZ, E T AISHAMMARI, A E HASSANIEN, et al. Genetic Algorithms for Multi-Objective Community Detection in Complex Networks, In Social Networks: A Framework of Computational Intelligence: Springer, 2014: 145-171.
[4] Z LI, J LIU. A Multi-agent Genetic Algorithm for Community Detection in Complex Networks. Physica A Statistical Mechanics & Its Applications, vol. 449, no. 5, pp. 336-347, 2016.
[5] X MENG, L DONG, Y LI, et al. A Genetic Algorithm Using K-path Initialization for Community Detection in Complex networks. Cluster Computing, vol. 20, no. 1, pp. 1-10, 2016.
[6] M GUERRERO, F G MONTOYA, et al. Adaptive Community Detection in Complex Networks Using Genetic Algorithms. Neurocomputing, vol. 226, no. Suppl. C, pp. 101-113, 2017.
[7] H LIU, F YANG, D LIU. Genetic Algorithm Optimizing Modularity for Community Detection in Complex Networks. Proceedings of the 35th Chinese Control Conference. July 27-29, 2016, Chengdu, China.
[8] M E J NEWMAN. Fast Algorithm for Detecting Community Structure in Networks. Physical Review E Statistical Nonlinear & Soft Matter Physics, vol. 69, no. 6, pp. 066133, 2004.
[9] A. Lancichinetti, S. Fortunato, János Kertész, Detecting the overlapping and hierarchical community structure in complex networks, New J. Phys. vol. 11, no. 3, 033015, 2009.
[10] W W Zachary. An information flow model for conflict and fission in small groups. Journal of Anthropological Research, vol. 33, no. 4, pp. 452-473, 1997.
[11] 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, vol. 54, no. 4, pp. 396-405, 2003.