Information Sciences 614 (2022) 123–137

Contents lists available at ScienceDirect

Information Sciences

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

Hiding ourselves from community detection through genetic algorithms

Dong Liu[a,b,c,*], Zhengchao Chang[a], Guoliang Yang[a], Enhong Chen[d,*]

[a] College of Computer and Information Engineering, Henan Normal University, Xinxiang 453000, Henan, China
[b] Key Laboratory of Artificial Intelligence and Personalized Learning in Education of Henan Province, Xinxiang 453000, Henan, China
[c] Big Data Engineering Lab of Teaching Resources & Assessment of Education Quality, Henan Province, China
[d] School of Computer Science, University of Science and Technology of China, Hefei 230000, Anhui, China

A R T I C L E I N F O

Article history:
Received 19 May 2022
Received in revised form 13 August 2022
Accepted 3 October 2022
Available online 7 October 2022

Keywords:
Community detection
Community hiding
Genetic algorithm
Privacy protection
Normalized mutual information

A B S T R A C T

Community structure plays an important role in social networks, which can reveal potential social relationships and deliver vast economic benefits to enterprises and organizations. Many efficient community detection algorithms have been proposed by researchers. However, effective community detection algorithms are accompanied by a growing problem of privacy disclosure. People have started to worry that their private information will be overexposed by community detection algorithms, so determining how to hide the community structure in the network to resist community detection algorithms has become an important issues. In view of this, we develop effective strategies to attack community detection algorithms through invisible disturbances to the network, namely, adding and removing a small number of connections, thus achieving privacy protection. In particular, a hiding strategy named "community hiding based on genetic algorithms using NMI (CGN)" is proposed in this paper. The algorithm uses normalized mutual information (NMI) as the fitness function and achieves an efficient global hiding effect by introducing a gene pool with prior information. We launched attacks based on CGN against four community detection algorithms on multiple real-world networks. By comparing with several state-of-the-art baseline algorithms, our CGN achieved the optimal results in NMI reduction. By visualizing the attack effect, it is proven that our CGN can achieve the community division error of nodes irrelevant to the connection changes by changing a very small number of connections, which fully reflects the concealment of community hiding. In addition, we further test the transferability and find that the modified network obtained by CGN on a specific community detection algorithm also shows extraordinary hiding effects when extended to other algorithms.

1. Introduction

A community, with respect to graphs, can be defined as a subset of nodes that are densely connected to each other and loosely connected to the nodes in the other communities in the same graph. Social media platforms such as Facebook, Instagram, and Twitter are places where people are constantly making friends with each other. Eventually, after a while, they end up being connected with people belonging to different social circles. These social circles can be a group of relatives, school-

* Corresponding author at: College of Computer and Information Engineering, Henan Normal University, Xinxiang 453000, Henan, China.
E-mail addresses: liudong@htu.edu.cn (D. Liu), cheneh@ustc.edu.cn (E. Chen).

https://doi.org/10.1016/j.ins.2022.10.027
0020-0255/© 2022 Elsevier Inc. All rights reserved.

mates, colleagues, etc. These social circles are nothing but communities. Detecting communities is important for many reasons in network analysis. It has the capability of revealing hierarchies that exist in real networks [1] and allows the existence of a nontrivial internal network organization to be revealed at a coarse-grained level. This allows further inference of special relationships between nodes that may not be easily obtained from direct empirical tests. As paradigmatic examples, communities in social networks represent groups of individuals with common interests and backgrounds, and imply patterns of real social groups. Moreover, community members with similar interests are likely to have the same purchasing tendency, which is conducive to the establishment of an efficient recommendation system [2].

However, in recent years, with the rapid development of community detection algorithms [3–11], information overmining, a new and challenging problem, has gradually emerged. People realize that some information, even their own privacy, will be overmined by these community detection algorithms. For example, people may not want their identity, interests, occupation, and some similar information to be discovered, but if the information about other members of the same community is disclosed, their information will be discovered as well. Therefore, a remedy to avoid the recognition of mainstream community discovery algorithms namely community hiding [12–14] has become an increasing concern.

Community hiding is a kind of hiding algorithm against community detection algorithms, that can reduce detection accuracy of the community detection algorithms by slightly modifying some network structures. It can be used as a guideline that individuals or organizations can implement to artificially manage their social circles or change the way of communication or channels to protect their privacy from being disclosed [12,15–21]. Community hiding algorithms can help the public protect their privacy from corporate interests. For example, people may not want their identity, interests, occupation, and some similar information to be discovered, but if the information about other members of the same community is disclosed, their information will be discovered as well, and community hiding algorithms can prevent that from happening. Reversely, it may also help establishment understand how industrial espionage escape from detection, especially given the increasing reliance on social networks. However, at present, many community hiding algorithms only focus on local information, which leads to the failure of causing large-scale damage to the community detection algorithms. In addition, the budget design of these algorithms is not very reasonable, because they do not achieve a good hiding effect unless very high budgets are employed.

This paper therefore focuses on the problem of allowing people to hide themselves to the maximum extent from community detection algorithms within very small budgets. Mapping this problem to the network data seeks to accomplish the maximum change in the community structure by reconnecting the minimum number of edges. It is a typical combinatorial optimization problem to find the attack scheme that can minimize the performance of community detection algorithms under given budgets.

Genetic algorithms (GAs) [22–26] and other evolutionary algorithms have been widely applied in various disciplines due to their good performance in solving complex optimization problems such as complex networks [27–29]. In addition, network data such as social networks are generally represented by graph data, which are generally discrete values, while genetic algorithms are naturally suitable for discrete data that do not require continuous and differentiable data. Therefore, we propose a heuristic attack strategy to solve the above problems for community hiding. This strategy is a genetic algorithm-based hiding approach that uses normalized mutual information (NMI) [30] as a fitness function. The advantage of using NMI as a fitness function is that the minimum value of NMI indicates that the community structure of the network after attack is quite different from that of the original network. Therefore, the reduction of NMI must make the distribution of nodes as different as possible before and after the attack. In addition, we introduced gene pools with prior information added. The gene pool marks links within the community as edges to be deleted and links outside the community as edges to be added. By introducing gene pools with prior information, the size of knowledge space is effectively reduced. In this way, an efficient global attack can be achieved through genetic algorithm iteration. The effectiveness of CGN is verified in different community detection algorithms and multiple real networks. The main contributions of our work are summarized as follows:

- First, a CGN method is proposed, which substantially reduces the performance of the community detection algorithms by introducing the prior information of the attack group, adding invisible interference to the network, and using NMI as the fitness function;
- Second, comprehensive experiments are carried out on several real networks to verify the hiding effect of CGN on four community detection algorithms, and compared with several excellent baseline algorithms, the optimal results are obtained;
- Finally, the transferability of CGN is verified, and community hiding can also be used as a robust evaluation index to compare the anti-attack capability of community detection algorithms.

The rest of this article is organized as follows. In the second part, we review the traditional community detection algorithms and the related works of community detection algorithms the adversarial attack networks. Then, in Section 3, we describe our approach in detail. After that, we present a number of experiments and a series of discussions in Section 4. Finally, the fifth part summarizes the thesis and looks forward to future work.

2. Related work

This section will introduce the current mainstream community detection algorithms and advanced community hiding algorithms in detail, and describe some of the imperfections of the current community hiding algorithms.

2.1. Community detection

Since Newman proposed community structure [31], an important statistical feature of complex networks, community detection has been a highly concerning area of research. The community detection method [32–35] based on network topology assumes that the community structure conforms to the network topology characteristics of relatively dense node connections within the community and relatively sparse connections between the communities. The simplest method is to minimize the delete link at the expense of the realization of the aim of community division [31,36,37], such as minimum cut, normalized cut and the number of edge interfaces. One of the most representative approaches is the Girvan–Newman algorithm [31]. Modularity [38] is currently the most commonly used method to measure the strength of network community structure, which essentially represents the numerical difference between the observation network and the internal edge of the community in the equivalent random graph. To reduce the computational cost, adapting to the actual need, found in a large social networking community label propagation algorithm (LPA) [35,5] can quickly detect a large social network of community structures, and its essence follows the characteristics of network homogeneity. In addition, community detection algorithms that measure network topology distance from different aspects are also emerging. These include random walks [39], graph coloring [40], game theory [41], etc. The aim of this method is to construct a community detection algorithm with low time complexity and adapt it to the community detection task of large-scale networks. In Bayesian generation models, community structure is modeled from the perspective of network generation mechanisms [42]. The random block model, which assumes that the generation of edges between two nodes is entirely determined by the relationship between the communities (blocks) to which they belong, is widely used in community detection tasks. To overcome the influence of the node degree on network partitioning, Newman's research group proposed a degree-corrected stochastic block model(DC-SBM) [43]. Sun et al. [11] proposed a graph neural network encoding method for multiobjective evolutionary algorithm to handle the community detection problem in complex attribute networks. Yin et al. [10] proposed an efficient and effective multi-objective method, namely DYN-MODPSO, and where the traditional evolutionary clustering framework and the particle swarm algorithm are modified and enhanced, respectively. Su et al. [9] proposed a new taxonomy covering different state-of-the-art methods, including deep learning-based models upon deep neural networks, deep nonnegative matrix factorization and deep sparse filtering.

2.2. Community hiding

As a classic research problem in network science, community detection has attracted many researchers from different disciplines. In addition, mining social network data can deliver vast benefits to enterprises and organizations. However, with the emergence of a large number of efficient and practical community detection methods and tools, many negative effects also occur, such as user information leakage, network information fraud, malicious information dissemination, etc. Therefore, the community hiding problem has increasingly highlighted the importance and urgency of network mining research.

Community hiding refers to attacking network data by some strategy to avoid detection by mainstream community detection algorithms to realize the hiding of the target node, target community or global community structure. Community hiding is a related issue in community detection. With the emergence of efficient and practical community detection methods and tools, the study of community hiding has become increasingly important and urgent. In essence, community hiding conceals the network structure or node attributes through some strategy to interfere with the recognition accuracy of the mainstream community detection algorithms and realize the hiding of the target community structure. Fig. 1 gives an overview of community hiding. An attack algorithm attacks some nodes (edge addition or edge deletion) in the network, so that the community structure detected by the original community detection algorithm is very different. For example, members of one community are dispersed to other communities, or even the number of communities in the entire network changes dramatically.

A related problem was introduced by Nagaraja [15]. The author proposed the community hiding problem for the first time and determined whether to add edges according to the node centrality to realize target community hiding. Waniek et al. [16] proposed a heuristic network reconnection edge algorithm, which essentially adopts the naive idea of removing the internal edges of the community and increasing the edges between communities, to conceal the target community structure. Due to the use of the edge random selection strategy, the time consumption of the algorithm is large. Chen et al. [20] constructed a formal representation based on node attacks and proposed the evolutionary perturbation attack (EPA) algorithm to realize the microcommunity hiding target. Fionda et al. [17] proposed the deception score model for evaluating the hidden effects of target communities. The essence of this model is to anticipate that the target community nodes can be uniformly distributed in increasingly more other communities. Chen et al. [18] proposed the Q-Attack algorithm, which constructed the fitness function based on modularity and then optimized it using the genetic algorithm to realize global community hiding. Liu et al. [19] constructed information entropy based on the community to depict the community structure problem and found

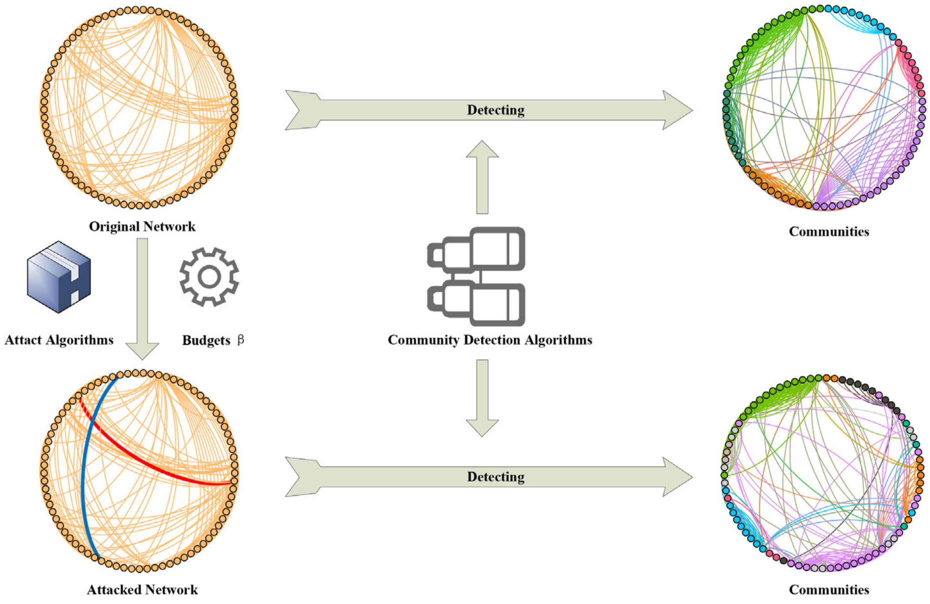


Fig. 1. Given a original network, the community structure in the network is detected by the community detection algorithm. Then, the original network is attacked by some operation such as edge deletions and edge additions, and further analyzed by community detection algorithms, which will make the resulted network structure greatly changed.

the edge that needed to be increased the most through the entropy residual error minimization (REM) algorithm to realize the global community hiding of the network. Li et al. [21] built a community discovery graph learning model on a graph neural network and a minimum cut, combined local and global similarities to design an edge modification strategy, and finally realized community hiding of target nodes. Mittal et al. [13] introduced network deception using permanence loss (NEURAL), a novel method that greedily optimizes a node-centric objective function to determine the rewiring strategy. Liu et al. [14] proposed a probabilistic framework named as ProHiCo to hide communities. The key idea of ProHiCo is to first allocate the resource of perturbations randomly and fairly and then choose the appropriate edges to perturb via likelihood minimization.

However, at present, there is still not enough attention given to protecting personal privacy through community hiding. The design of the budget is not very reasonable. For example, Q-Attack [18], the best current algorithm, which also uses a genetic algorithm, attacks specific nodes by artificially adding an edge and deleting an edge, which doubles the budgets and fails to give full play to the maximum efficiency of genetic algorithm. In this paper, a gene pool with a priori information is creatively established. Under the constraint of budgets, the selection of adding or deleting edges is completely determined by the algorithm without human intervention, which achieves the best attack effect under an the invisible disturbance.

3. The proposed methods

In this section, we define and formulate community attack problems, and then describe how "Community hiding based on Genetic algorithms using NMI (CGN)" can be used to attack and degrade the performance of community detection algorithms. The symbols in this paper are listed in Table 1.

3.1. Problem definition

We represent a set of nodes $V = \{v_1, v_2, \dots, v_n\}$, representing real-world entities, for example, friends in a social network and papers in a citation network. We use a set of edges $E = \{e_1, e_2, \dots, e_m\}$ to describe the connections between nodes in V. e_i indicates whether there is an undirected edge between a pair of nodes, such as the establishment of a friend relationship between the two accounts and the mutual reference of the two papers. In this study, we focus on undirected graphs; however, our approach also extends to directed graphs.

Definition 1 (Community Detection). Let $G = (V, E)$ be an undirected graph including a set of nodes V and set of links E. The goal of the community detection problem is to divide a graph $G = (V, E)$ into K nonintersecting graphs $G_i = (V_i, E_i), i = 1, \dots, K$, where $V = \cup_{i=1}^{K} V_i$ and $V_i \cap V_j = \varnothing$ for $i \neq j$. Given a set of communities discovered by some community detection algorithm A_D is represented by $\overline{C} = \{C_1, \dots, C_K\}$, with $C_i \subseteq V$ and $C_i \cap C_j = \varnothing$, for all $i,j \in \{1, \dots, K\}$ and $i \neq j$.

Definition 2 (Perturbation Network). Given the original net network $G = (V, E)$, adversarial network $\widehat{G} = \left(\widehat{V}, \widehat{E}\right)$ is made by adding disturbance $E_\beta = \left\{e_1^+, e_2^-, \dots, e_t^+\right\}$ on G,where β is the budget and +, − indicates adding and deleting edges, respec-

Table 1
Symbols used in this paper.

Symbol	Definition	Symbol	Definition
$G = (V, E)$	original network with nodes V, links E.	$\widehat{G} = \left(\widehat{V}, \widehat{E}\right)$	adversarial network with nodes $\widehat{V}$, updated links $\widehat{E}$.
$\overline{C}, \widehat{C}$	the communities discovered by some community detection algorithms of G and adversarial network $\widehat{G}$.	A_D	the community detection algorithm
β	the budget	β_+	the number of edges added
β_-	the number of edges removed	E_β	the set of edges that are changed under budget constraints
$\overline{E}^+_\beta, \overline{E}^-_\beta$	stands for adding edges between communities and removal within the same community.	P_m	mutation probability.
P_c	crossover probability.	f	fitness function.

tively. A_D is a community detection algorithm that discovers the set of communities $\widehat{C} = \left\{\dot{C}_1, \ldots, \dot{C}_L\right\}$ in $\widehat{G}$. $\widehat{C}$ is as different as possible from $\overline{C} = \{C_1, \ldots, C_K\}$, where $L = K$ or $L \neq K$.

Definition 3 (Community Hiding). The community hiding problem is defined as finding a strategy to make small budgets on $G = (V, E)$ change to $\widehat{G} = \left(\widehat{V}, \widehat{E}\right)$, formulated as

$$\begin{cases} \hat{V} = V \\ \hat{E} = E + E_\beta \\ E_\beta = \overline{E}^+_{\beta_+} - \overline{E}^-_{\beta_-} \\ \beta = \beta_+ + \beta_- \end{cases} \tag{1}$$

In this paper, we focus on adding and removing edges(i.e., $\widehat{G} = \left(\widehat{V}, \widehat{E}\right)$=$\left(V, \widehat{E}\right)$). Here, $\overline{E}^+_\beta$ stands for adding edges between communities, and $\overline{E}^-_\beta$ stands for edge removal within the community. The above Eq. (1) means that no changes are made to the nodes in the original network. Within the limitation of budget, the community structure in the network is changed by adding edges between communities and deleting edges within communities.

3.2. Framework

There are three main steps to achieving community hiding: (1) Community detection. Community detection algorithm A_D is used to detect the graph. (2) Rewiring connections. Changes are made to the edges in the graph. (3) Community redetection and results comparison. This problem can be regarded as a minimum-maximization problem. Its greatest challenge is to find the smallest set of edges that can be added (or removed) to have the greatest impact on the structure of the graph. The evolutionary algorithm is famous for its good effect on optimization problems.

An evolutionary algorithm is an intelligent optimization technology designed by simulating natural phenomena or utilizing various mechanisms of natural organisms [44–46]. The genetic algorithm is a typical evolutionary algorithm. Its basic model was first introduced and studied by Holland [44], mainly using a natural selection mechanism to solve optimization problems. At the beginning of the algorithm, the chromosome groups representing individuals with different characteristics are initialized, and then descendants are generated according to the designed crossover, mutation and other genetic operators. Better-adapted individuals have a greater chance of surviving and reproducing, so populations can evolve. In practice, direct confrontational attacks can be difficult. Therefore, we further designed a more feasible hiding strategy that can implement the attacker's behavior more effectively. Fig. 2 gives an overview of the evolution process, and the pseudocode of this attack strategy is shown in Algorithm 1. The gene pool was created according to the community structure of the original graph G discovered by the community detection algorithm, and genes were extracted from the gene pool to form chromosomes and form the population. After selecting, crossover and mutation of several generations, the optimal attack strategy was obtained. The same community detection algorithm was used on the attacked graph $G\prime$, and the community structure was found to be quite different. In Algorithm 1, we introduce the concept of gene pool to reduce the size of knowledge space and improve the efficiency of the algorithm. Edges with markers (added or removed) are randomly selected from the gene pool to form chromosomes. The length of the chromosome is the budget value. Multiple chromosomes are grouped into populations according to hyperparameters. The fitness value of each chromosome was calculated according to the fitness function. The chromosome with the highest fitness value was selected by wheel selection. Cross and mutate chromosomes according to hyperparameters. The evolution generation is set to be a constant, and when this condition is met, the algorithm stops. The flow-chat of Genetic Algorithm for community hiding as shown in Fig. 3. Our strategy is discussed in detail below.

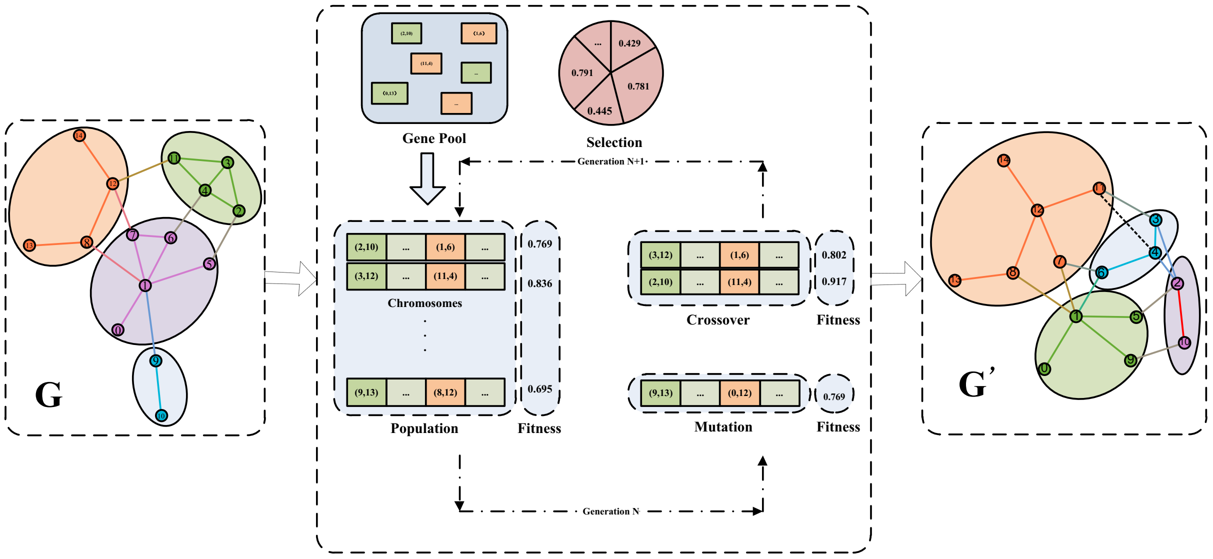


Fig. 2. The overview of CGN strategy.

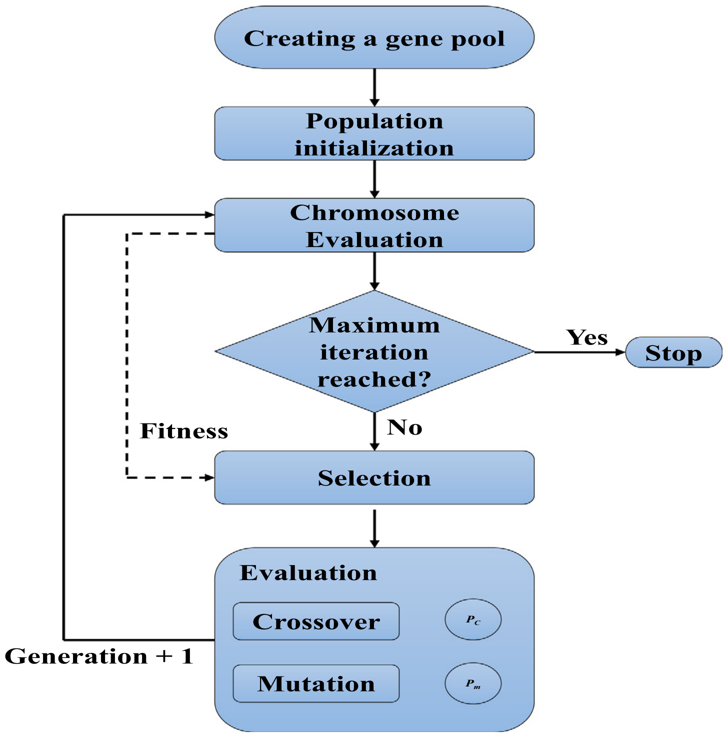


Fig. 3. The flow-chat of Genetic Algorithm for community hiding.

3.3. The Genetic Algorithm for community hiding

Gene pool containing prior information. Inspired by Waniek et al. [16], according to the community structure discovered by the community detection algorithm, edges within the community (edges that may be deleted) and edges between communities (edges that may be added) are mixed into a set. Each edge in the set is represented as a gene. The gene markers formed by the edges within the community are deleted, and the gene markers formed by the edges between the communities are added. All genes constitute a gene pool.

Encoding. Under budget constraints, a chromosome is made up of randomly selected genes that are not put back into the gene pool. The length of the chromosome is the budget value. As shown in Fig. 4, according to the community structure in the original network, edges inside the community were selected as edges to be deleted and edges outside the community were selected as edges to be added to form a gene pool. According to the budget value, the edges in the gene pool are selected to form chromosomes. Perform the chromosome to complete community hiding operation.

Fitness function. There are many standards to measure the quality of community detection results, and the most common ones are modularity and normalized mutual information. Modularity [38] mainly describes the closeness of connections within and between communities, and does not directly reflect the community belonging of to nodes. Normalized mutual information(NMI) [30] mainly describes the similarity between two clustering results. Therefore, the reduction of NMI must make the distribution of nodes as different as possible before and after the attack, so the fitness function is designed as follows:

$$f = 1 - NMI \tag{2}$$

indicates that the larger the fitness function value is, the better the attack strategy produced by the selected chromosome is.

Population. This is the set of chromosomes in which the number of individuals is called population size. Each chromosome in the population can cross and mutate, and the optimal chromosome is the attack strategy with the maximum fitness function value that should be adopted.In this paper, a chromosome is composed of several edges. Some of these edges are marked as deleted or added. Perform these edge-adding and edge-removing operations in the network, and each chromosome will get its own f value.

Selection. In nature, the more adaptable individuals are, the more likely they are to reproduce. However, you cannot say that the more fit you are, the more offspring you have, only in terms of probability. The link between nature and mathematical problems is roulette wheel selection. The fitness values of chromosomes in a population form a wheel. The larger the fitness value, the larger the wheel area occupied by the chromosome. In random selection on the wheel, the chromosome with a higher fitness value has a higher probability of being selected, which is similar to the survival of the fittest principle in nature. The formula is as follows:

$$p_i = \frac{f(i)}{\sum_{j=1}^{n} f(j)} \tag{3}$$

Crossover. Two chromosomes are cut at the same place, and the two strands are crossed to form two new chromosomes. Additionally, this is known as gene recombination or hybridization. Here, a method of randomly generating a cut point is adopted to change the gene fragments between the parents. The probability of a pair of individuals producing offspring through crossover operations is represented by P_c.

Mutation. It is possible (with a small probability) to make some kind of replication error, to mutate into new chromosomes and to exhibit new traits. This is reflected in the genetic algorithm to prevent falling into the local optimal solution. In this paper, a random gene was selected from the gene pool to replace a gene randomly in the chromosome. The mutation probability is expressed by P_m.

Termination conditions. The evolution generation is set to be a constant, and when this condition is met, the algorithm stops.

In summary, CGN mainly involves gene pool construction, encoding, fitness function design and a crossover mutation strategy.

Algorithm 1: Method of CGN

```
   Input: $G,\beta,P_c,P_m,PopSize,A_D,Generation$
   Output: The adversarial network $\widehat{G}$;
1  GenPoll ← CrateGenPoll($G,A_D$);
2  Chrom ← RandomChoice($GenPoll,\beta$);
3  InitPop ← InitPop($Chrom,PopSize$);
4  for $i$ in $Generation$ do
5      FitPop ← Fitness($InitPop$);
6      SelectedPop ← Selection($FitPop$);
7      CrossPop ← Cross($SelectedPop,P_c$);
8      MutaPop ← Mutation($CrossPop,P_m$);
9      InitPop ← $MutaPop$;
10 BestChrom ← ChoiceBestChrom($InitPop$);
11 AddDelEdges($BestChrom$);
12 return $\widehat{G}$;
```

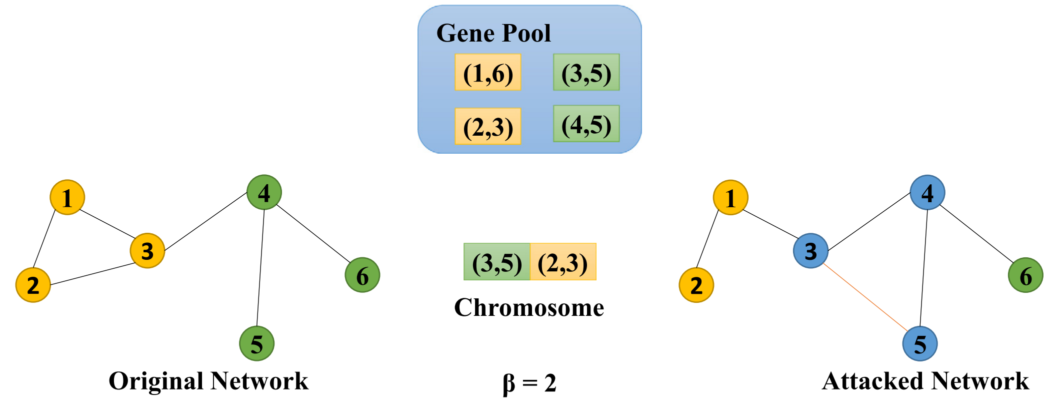


Fig. 4. The example for encoding step.

4. Experiments

We now report an experimental evaluation of an extensive community hiding algorithm. This section describes the datasets, baselines, and parameter settings associated with the experiment and reports the results. We verify the validity of the model on four real datasets and four community detection algorithms and obtain state-of-the-art results compared with the five baseline algorithms. The code of the CGN algorithm is available online.[1]

Datasets.

Karate:[2] The Zachary Network is a social network built by observing a karate club at an American university. The network consists of 34 nodes and 78 edges, with individuals representing members of the club and edges representing friendship relationships among members. The karate club network has become a classic problem in the detection of complex network community structures.

Miserables:[3] This is a coappearance network of characters in the novel Les Miserables. The network consists of 77 nodes and 254 edges.

DBLP:[4] Digital Bibliography Project (DBLP) is a computer science bibliography. In this dataset, authors are considered users, the paper titles of the authors are the text of users and the coauthorship relationship forms the links of users. The network consists of 5304 nodes and 14227 edges.

PubMed:[5] PubMed comprises more than 26 million citations for biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full-text content from PubMed Central and publisher websites. The network consists of 19717 nodes and 44324 edges.

Community Detection Algorithms. As the enemy of the spoofing algorithm, we consider four detection algorithms as our attack targets.

1. **Multilevel** [32]**:** The algorithm regards the community detection problem as a multilevel modularity optimization problem. It starts with isolated nodes and repeats the process of moving nodes to communities to maximize modularity gains until further improvements are impossible. Then, a new network is built by merging communities into a single supernode. These two steps are repeated until the algorithm stabilizes.
2. **Fastgreedy** [33]**:** This is based on a greedy modularity maximization strategy.
3. **Label_propagation** [35,5]**:** The algorithm starts by assigning each node a label, which is updated to the most frequent label among the node's neighbors as the information is propagated. The algorithm terminates when no node changes its label.
4. **Infomap** [34]**:** It returns a community structure that provides the shortest description length for a random walk.

Baseline. We compare CGN with five baseline methods as follows:

[1] https://github.com/HNU-CCIE-AI-Lab/CGN.
[2] http://www-personal.umich.edu/mejn/netdata/.
[3] http://www-personal.umich.edu/mejn/netdata/.
[4] https://dblp.uni-trier.de/.
[5] http://snap.stanford.edu/data.

1. **Random Attack (RAT):** In the original network, a link B is randomly disconnected and then is B randomly connected to nodes that were not originally connected. This is the easiest way to attack.
2. **Internally disconnected, externally connected (DICE):** DICE was originally a heuristic algorithm used to disguise the community. For the target community in the network, DICE first deletes the internal links of A and connects B links in the community to distant nodes, making the target invisible [16].
3. **Safeness-Based Attacks (SBA):** These attacks weaken the community structure by removing edges within the community and inserting edges between communities. They are based on safeness [17].
4. **Q-Attack:** Based on the genetic algorithm, the attack method of specific nodes is selected with modularity as the fitness function [18].
5. **NEURAL:** It introduces network deception using permanence loss (NEURAL), a method that greedily optimizes a node-centric objective function to determine the rewiring strategy [13].

Metrics.

1. **Normalized Mutual Information (NMI):** Normalized mutual information is commonly used in clustering to measure the degree of similarity between two clustering results. It was proposed by Danon et al. [30]. It is an important measurement index of community detection, which can objectively evaluate the accuracy of a community division compared with the standard division. The range of NMI is 0 to 1, and the higher the value is, the more accurate it is when performing community detection. Let the joint distribution of two random variables (X, Y) be $p(x, y)$, and let the edge distribution be $p(x), p(y)$. Mutual information $I(X;Y)$ is the relative entropy of the joint distribution $p(x, y)$ and the product distribution $p(x)p(y)$, namely, the formula is:

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{4}$$

An obvious problem with mutual information as a similarity measure alone is that subpartitions obtained by splitting some of *Y*'s clusters into smaller clusters have the same mutual information as *Y*. Therefore, NMI is proposed to address this problem, which is defined as:

$$NMI(X;Y) = \frac{2I(X;Y)}{H(X) + H(Y)} \tag{5}$$

where $H(X) = -\sum_i p(x_i) \log p(x_i)$ is the information entropy. In contrast to the community detection results, the closer NMI is to zero in the community hiding problem, the better. The advantage of using NMI as a fitness function is that the minimum value of NMI indicates that the community structure of the network after attack is quite different from that of the original network. Therefore, the reduction of NMI must make the distribution of nodes as different as possible before and after the attack.
2. **β times NMI(BN):** It shows the cost of an attack. Attack cost is a measure of how many edges are modified while lowering the NMI value. This value will be greater than 1 and less than β, which usually means that the attack was successful despite the cost of the comeback. A smaller value indicates a more successful attack. Therefore, BN is proposed to solve this problem, defined as:

$$BN = \beta * NMI \tag{6}$$
3. **Modularity(Q):** It is a commonly used method to measure the structural strength of network communities. It was first proposed by Mark NewMan [38].The value of modularity mainly depends on the community allocation of nodes in the network, that is, the community division of the network. It can be used to quantitatively measure the quality of network community division. The closer the value is to 1, the stronger the community structure divided by the network, and the better the quality of network division. Therefore, the optimal division of the network community can be obtained by maximizing the modularity Q, which is defined as:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(i,j) \tag{7}$$

where m represents the number of edges in the network, A_{ij} represents the number of edges between node i and node j, and k represents the degree of the node. $\delta(i,j) = 0$ means that node i and node j are not in the same community and $\delta(i,j) = 1$ means that node i and node j are in the same community.

4.1. Experimental results

Our experiments were conducted on a PC with an AMD Ryzen 7 1700 eight-core processor @ 3.65 GHz (CPU) and 32 GB RAM. Our programming environment is Python 3.65.

First, the P_c and P_m parameters of the CGN are tested using a multilevel community detection algorithm on karate dataset to determine the optimal P_c and P_m. The settings are $\beta = 2, Generation = 200, PopSize = 100$. Then according to the parameter configuration, fixed $P_m = 0.1$, 0.6, 0.7, 0.8, and 0.9, and four values are assigned to P_c. Then fixed $P_c = 0.7$, 0.01, 0.04, 0.07, and 0.1, and four values are assigned to P_m. The fitness curve of the experiment is shown in Fig. 5, and the optimal parameters of this case are $P_c = 0.7$ and $P_m = 0.01$. Later, the experimental parameters P_c and P_m were set as 0.7 and 0.01, respectively, *Generation* as 400, *PopSize* as 200, and β was set to different values according to different datasets.

The time complexity analysis. The time complexity of CGN is $O(Gen \cdot Pop \cdot \beta)$, where *Gen* is the maximum iteration number, *Pop* is the population size, and β is the budget value. The $GenandPop \ll |V|$. The β value range is $[0, |E_\beta|] \ll |E|$. The time complexity of Q-Attack is $O(Gen \cdot Pop \cdot \beta)$, where the β value range is $[0, |E|]$. The time complexity of RAT is $O(|E|)$. The time complexity of DICE is $O(|E_\beta|)$. The time complexity of SBA is $O(|C| + |EC|)$. The time complexity of NEURAL is $O(|V| + |E|)$.

Effectiveness analysis of the CGN algorithm. A natural inclination was to surmise that the higher the beta, the better the attack would be. Therefore, to test this idea, we validated the Karate and Miserables datasets, generated 10 adversarial networks for each β and recorded the mean values of NMI and BN. To meet the covert requirement of the attack strategy, namely, adding an invisible disturbance to the network, the maximum value of β should not exceed 5% of the number of edges in the original graph. Fig. 6 shows the performance of NMI and BN under different budgets, and it is clear that beta improves as it increases, which is sufficient to confirm the above hypothesis. What is interesting is the variation in BN, which suggests that there is a certain budget that balances the effects of attacks against the costs of attacks. Surprisingly, even the $\beta = 1$ CGN can give a very good attack effect. A very surprising scene appears in the visualized attack effect, as shown in

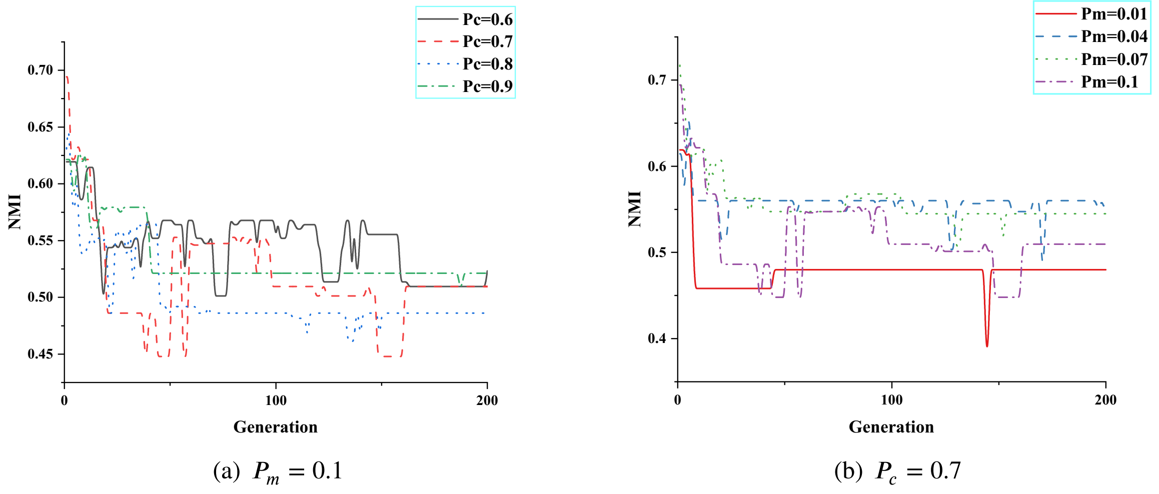


Fig. 5. The P_c and P_m parameters of CGN are tested using multilevel community detection algorithm on karate dataset to determine the optimal P_c and P_m.

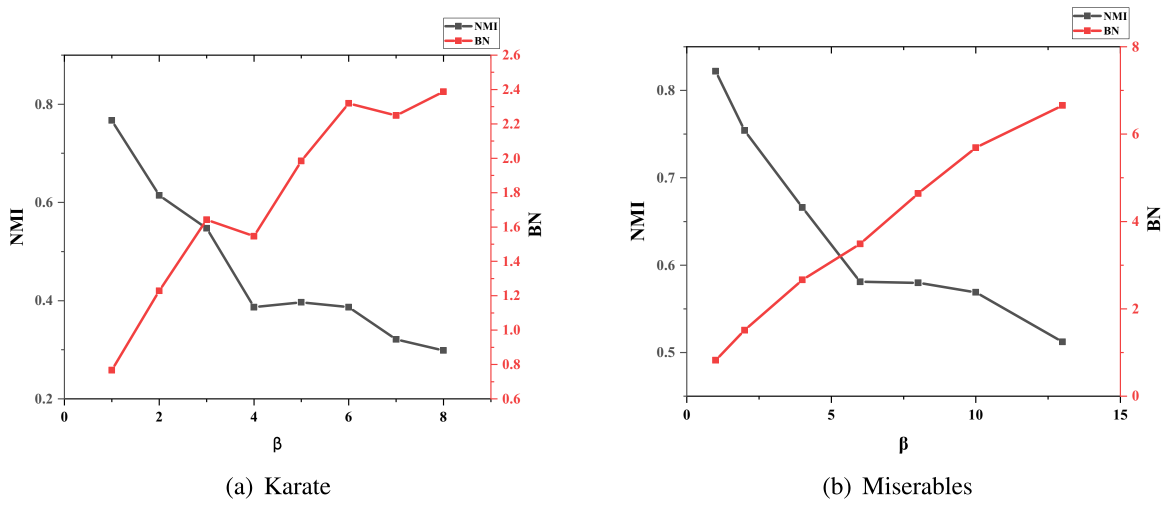
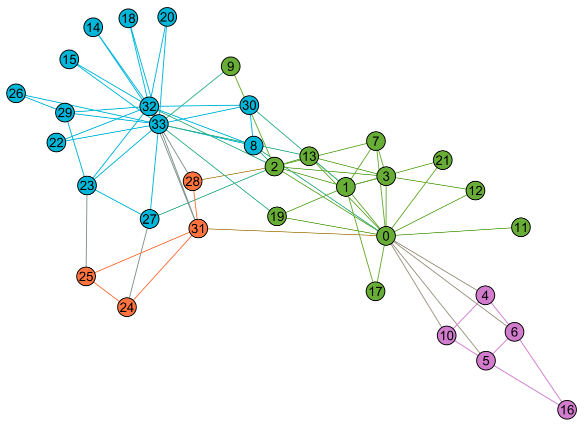


Fig. 6. Performance of NMI and BN under different budgets.

Fig. 7. When the CGN algorithm only adds one edge to the original graph, namely (16,17), the community where nodes 16 and 17 reside does not change much. However, at nodes 26 and 29, which are thousands of miles away, community affiliation has changed dramatically. This is very encouraging and demonstrates that the CGN algorithm is extremely stealthy.

The efficiency of CGN algorithm is improved by using gene pool with prior information and fitness function constructed by NMI. We validated multilevel as the attacked community detection algorithm on the Miserables dataset. The Eq. (2) and $f = 1 - Q$ were used as fitness functions respectively. β was set to 1%, 3%,5%, and 10%, respectively. The NMI targets corresponding to each β value are 0.85,0.75,0.65 and 0.55. When the algorithm reaches the preset NMI target value, the time taken by the algorithm is calculated. The experimental results are shown in Fig. 8(a). It can be seen that the algorithm for the fitness function based on NMI takes less time than the fitness function based on Q. This shows that the algorithm using NMI as fitness function has faster convergence speed and higher efficiency. The Eq. (2) was fixed as fitness function, and the gene pool with prior information(GP_P) was compared with the general gene pool(GP). The general gene pool is composed of existing edges and non-existing edges in the network, where the existing edges are marked as deletable edges and the non-existing edges are marked as additive edges. The time taken to calculate the algorithm when the algorithm reaches the preset NMI target value. Experimental results are shown in Fig. 8(b). The algorithm using a gene pool with prior information takes much less time than using a normal gene pool. It is proved that using gene pool with prior information can greatly shorten the convergence practice and improve the efficiency of the algorithm.


(a) Original network

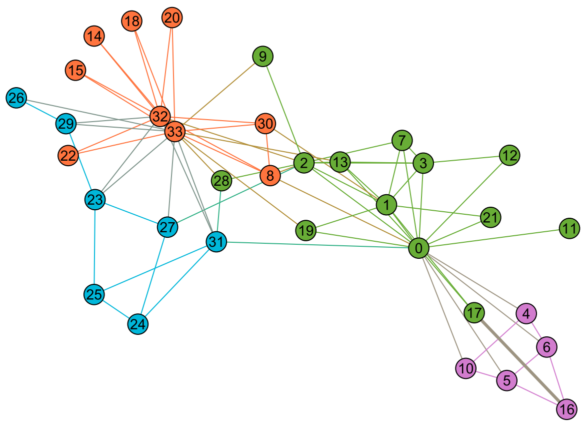
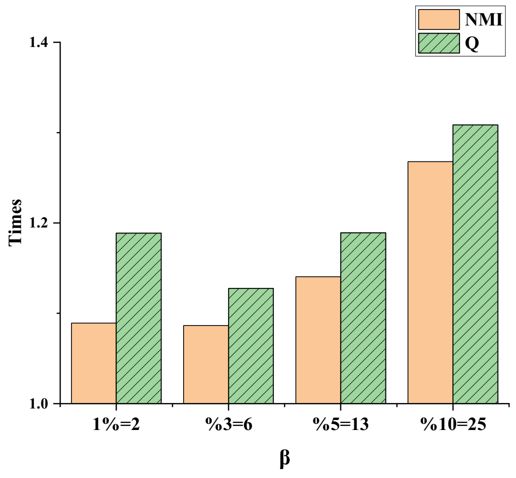
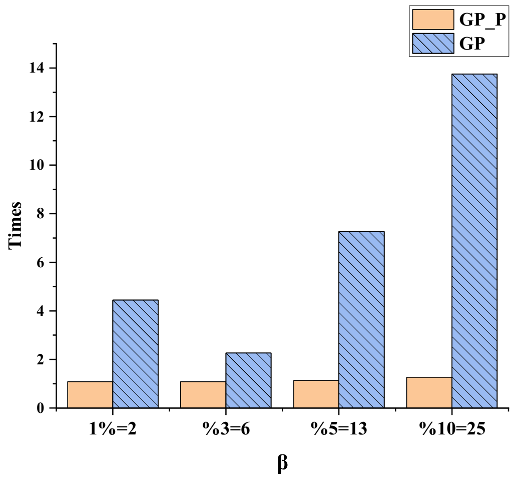

(b) Attacked network

Fig. 7. When $\beta = 1$, CGN attack karate network produced excellent attack effect.



(a) Comparison of different fitness functions



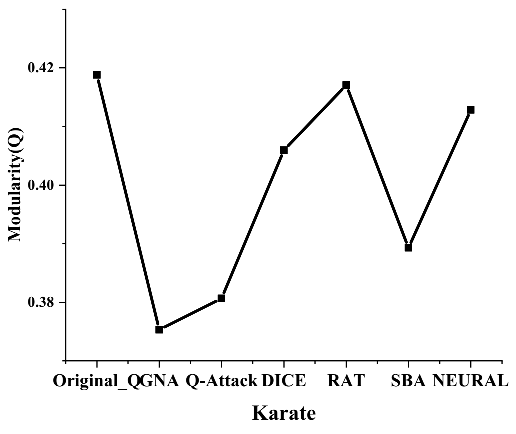
(b) Comparison of different gene pools

Fig. 8. The efficiency of CGN algorithm is improved by using gene pool with prior information and fitness function constructed by NMI.

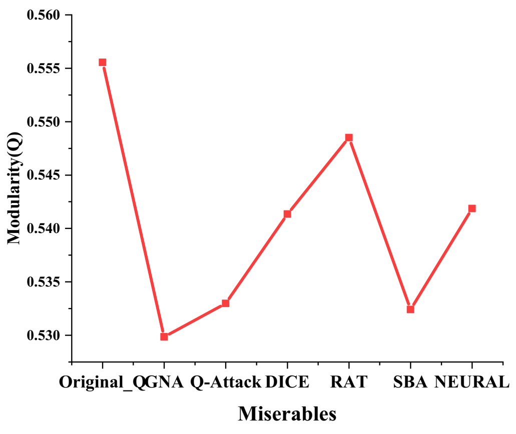
Validity analysis of different attack algorithms. This part compares the attack effects of the CGN and five advanced baseline algorithms against four community detection algorithms on four datasets. Before a detailed comparison, the calculation of the NMI value in this paper is explained first. In this paper, the NMI value is not calculated between the distribution

Table 2
NMI values of different attack algorithms.

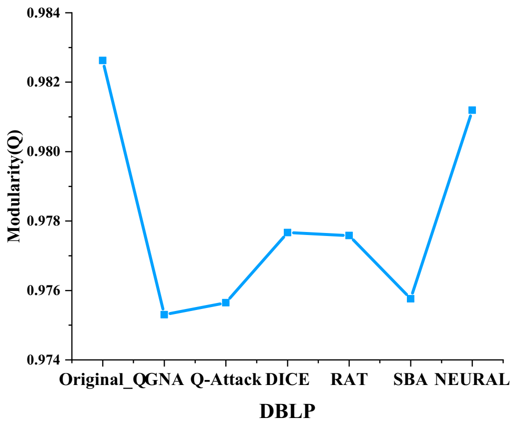
Network	β	Community detection	CGN	Q-Attack	DICE	RAT	SBA	NEURAL
Karate	5% = 4	Multilevel	**0.3867**	0.8949	0.9233	0.9321	0.9902	0.8485
		Fastgreedy	**0.3110**	0.8954	0.9199	0.9825	0.9053	0.6952
		Infomap	**0.5607**	0.8908	0.9103	0.9257	0.9112	0.9102
		Label_Propagation	**0.1124**	0.5722	0.6037	0.8755	0.5984	0.4032
Miserables	5% = 13	Multilevel	**0.5122**	0.8009	0.9252	0.9164	0.8967	0.8501
		Fastgreedy	**0.3688**	0.8211	0.9389	0.9921	0.9377	0.6937
		Infomap	**0.6879**	0.8550	0.9477	0.9825	0.9468	0.9037
		Label_Propagation	**0.1930**	0.7964	0.7963	0.9763	0.7835	0.6948
DBLP	1% = 142	Multilevel	**0.9606**	0.9821	0.9975	0.9905	0.9816	0.9936
		Fastgreedy	**0.9628**	0.9879	0.9896	0.9998	0.9783	0.9960
		Infomap	**0.9849**	0.9931	0.9987	0.9975	0.9954	0.9947
		Label_Propagation	**0.9393**	0.9648	0.9726	0.9825	0.9691	0.9519
Pubmed	1% = 443	Multilevel	**0.6983**	0.8814	0.9219	0.9266	0.8804	0.8683
		Fastgreedy	**0.5933**	0.8114	0.8496	0.9532	0.8211	0.8090
		Infomap	**0.9534**	0.9717	0.9865	0.9973	0.9723	0.9602
		Label_Propagation	**0.7645**	0.8587	0.8769	0.9012	0.8546	0.8425



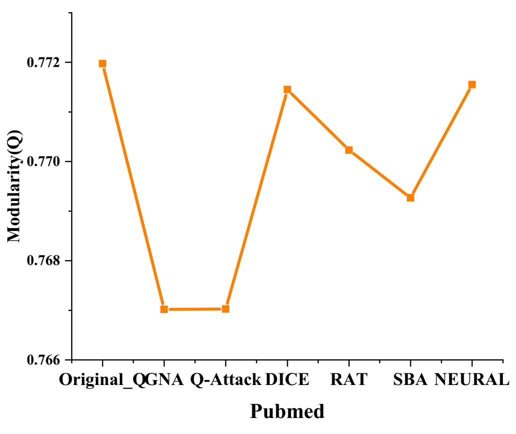
(a) Based on Multilevel algorithm



(b) Based on Multilevel algorithm



(c) Based on Multilevel algorithm



(d) Based on Multilevel algorithm

Fig. 9. Comparison of modularity values of six community hiding algorithms on different networks.

after the attack and the distribution of the original network. Instead, the community detection algorithm is used once on the original network, then the network attack is carried out, and finally the same community detection algorithm is used once. The NMI value is calculated based on the above results. This design makes the experimental results more objective and reasonable, because we aim to attack the community detection algorithm. The experimental results are shown in Table 2. The first one with the best attack effect is marked in bold. The attack effect of the CGN is completely superior to that of the baseline algorithm on all datasets and community detection algorithms. This compound expectation is not surprising because its purpose is to lower the NMI. The Q-Attack algorithm performs well on small datasets. The RAT algorithm is unstable, with occasional bright performance on small datasets but weak performance on large datasets. DICE and SBA also have some effect, and SBA has some competitiveness on big datasets, probably because it is also committed to breaking the community structure to set off a certain chain reaction. The NEURAL algorithm work well on many datasets, and the persistence losses they used are also aimed at breaking community structures.

Objectively verify the effectiveness of the CGN algorithm, we compare the hiding effect of the CGN algorithm with five advanced baseline algorithms based on the multilevel algorithm on four datasets. Before the detailed comparison, the calculation of the modularity Q value in this paper is explained first. The multilevel algorithm is first used on the original network to calculate the modularity at this time, and then the network attack is carried out, and the same community detection algorithm is used again to calculate the modularity at this time. The purpose of this design is to attack the community detection algorithm and make the experimental results more objective and reasonable. The experimental results are shown in Fig. 9. It can be seen that on all datasets, the attack effect of the CGN is better than that of the baseline algorithms. Even compared to the Q-Attack, an algorithm that focuses on reducing the value of modularity, CGN is no exception. The reason for this is that the gene pool with prior information introduced by CGN guarantees that the tightness within the community is destroyed. The RAT algorithm is unstable and occasionally performs well on small datasets but poorly on large datasets. DICE

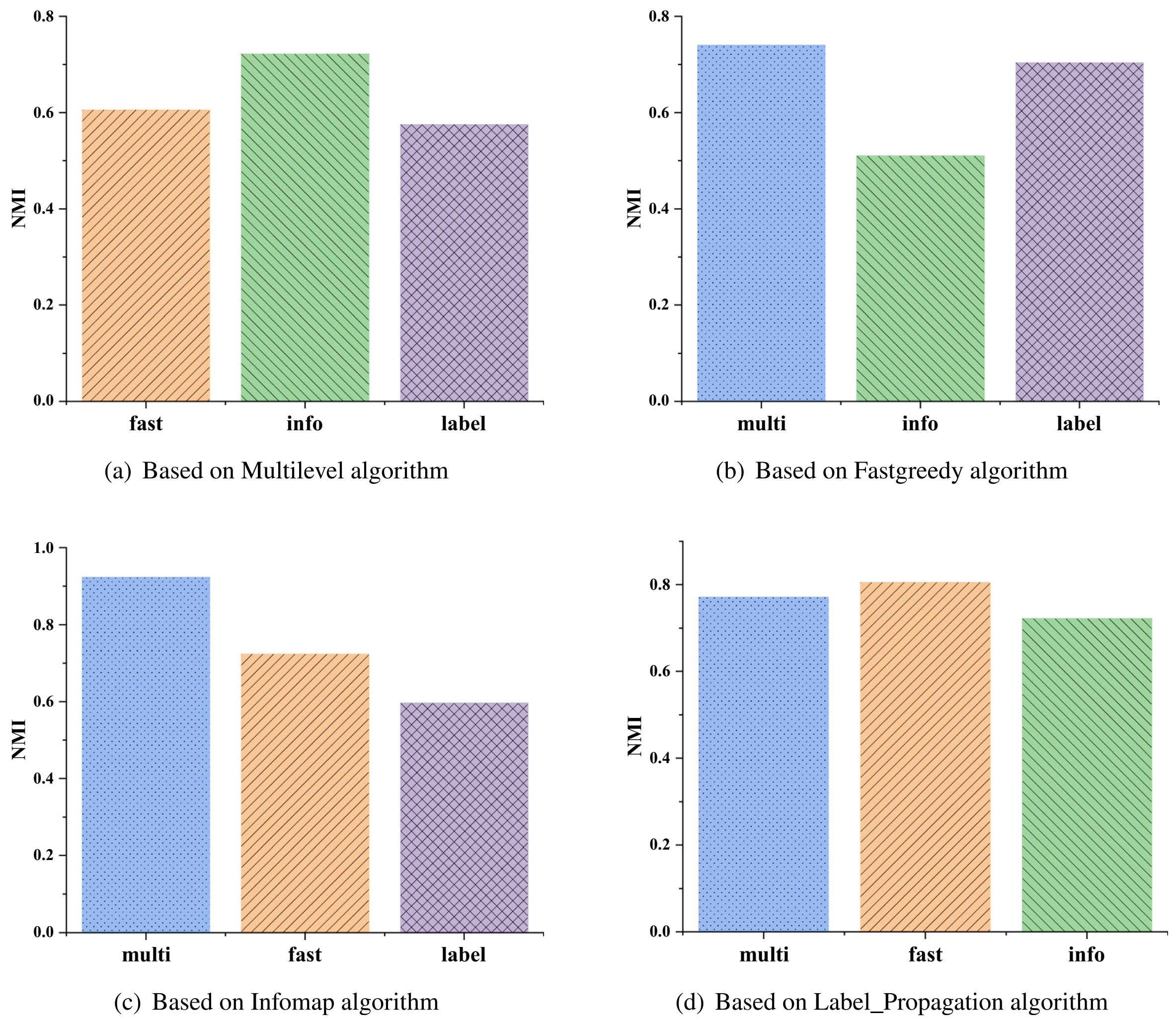


Fig. 10. Transferability experiments were performed on the Karate dataset using CGN.

also has a role to play, and SBA has a good competitive edge on big datasets, probably because it also works to break down community structures and set off certain ripple effects. NUERAL does not perform well for modularity reduction. The above experimental results show that the design of the CGN algorithm is very reasonable and effective.

Transferability analysis. Because NMI is a universal metric, CGN is applicable to all community detection algorithms. However, in the real world, it is not easy for an attacker to know which community detection algorithm is used by the network owner, so transferability experiments are necessary. We conducted cross-validation among the four community detection algorithms, as shown in Fig. 10, and found that the adversarial network obtained by using CGN on a specific community detection algorithm also had a good attack effect on other community detection algorithms. This shows that CGN can realize the original intention of the algorithm in both white-box attacks and black-box attacks, that is, adding invisible disturbances to the original network to destroy the overall community structure and realize the purpose of community hiding.

5. Conclusion and future work

In this paper, a community hiding algorithm based on a genetic algorithm with NMI as the fitness function was proposed, which effectively attacks the current mainstream community detection algorithms on multiple datasets by introducing gene pools with prior information as the basis of chromosome composition. The advantage of using NMI as a fitness function is that the minimum value of NMI indicates that the community structure of the network after attack is quite different from that of the original network. Therefore, the reduction of NMI must make the distribution of nodes as different as possible before and after the attack. CGN has achieved excellent results in experiments on different gene pools and fitness functions. CGN achieved the optimal results when compared with other advanced baseline algorithms. In the transplantation experiment, it is proven that CGN showed good performance in both white box attacks and black box attacks. These rich experiments showed that it is feasible to use the CGN algorithm to protect people's privacy from community detection algorithms in the real world.

With the deepening of people's concern about the current situation of personal privacy and the improvement of governments' awareness of the protection of citizens' privacy, increasing attention will be given to how to break down excessive information mining in social networks. Much remains to be studied in future work. The main type of community detection algorithm involved in this paper is nonoverlapping community detection, and determining how to expand the attack to the overlapping community detection algorithm is also a very worthy direction of research. A more speculative but interesting idea is to investigate how to implement attack algorithms against heterogeneous networks, dynamic networks, and networks containing attributes.

CRediT authorship contribution statement

Dong Liu: Conceptualization, Methodology, Funding acquisition, Writing – review & editing, Supervision, Resources, Investigation, Validation. **Zhengchao Chang:** Methodology, Software, Visualization, Writing – original draft, Formal analysis, Data curation. **Guoliang Yang:** Software. **Enhong Chen:** Writing – review & editing, Supervision, Resources, Investigation, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62072160), the key scientific and technical project of Henan Province (212102310381) and the Science Foundation of Distinguished Young Scholars of Henan Normal University(20210347).

References

[1] H.A. Simon, The architecture of complexity, in: Facets of systems science, Springer, 1991, pp. 457–476.
[2] P.K. Reddy, M. Kitsuregawa, P. Sreekanth, S.S. Rao, A graph based approach to extract a neighborhood customer community for collaborative filtering, in: International Workshop on Databases in Networked Information Systems, Springer, 2002, pp. 188–200.
[3] S. Fortunato, D. Hric, Community detection in networks: A user guide, Phys. Rep. 659 (2016) 1–44.
[4] D. Liu, D. Duan, S. Sui, G. Song, Effective semisupervised community detection using negative information, Math. Problems Eng. 2015 (2015).
[5] D. Liu, X. Liu, W. Wang, H. Bai, Semi-supervised community detection based on discrete potential theory, Phys. A 416 (2014) 173–182.
[6] D. Liu, H.-Y. Bai, H.-J. Li, W.-J. Wang, Semi-supervised community detection using label propagation, Int. J. Mod. Phys. B 28 (2014) 1450208.
[7] L. Fan, S. Xu, D. Liu, Y. Ru, Semi-supervised community detection based on distance dynamics, IEEE Access 6 (2018) 37261–37271.
[8] D. Liu, C. Wang, Y. Jing, Estimating the optimal number of communities by cluster analysis, Int. J. Mod. Phys. B 30 (2016) 1650037.
[9] X. Su, S. Xue, F. Liu, J. Wu, J. Yang, C. Zhou, W. Hu, C. Paris, S. Nepal, D. Jin, et al., A comprehensive survey on community detection with deep learning, IEEE Trans. Neural Networks Learn. Syst. (2022).
[10] Y. Yin, Y. Zhao, H. Li, X. Dong, Multi-objective evolutionary clustering for large-scale dynamic community detection, Inf. Sci. 549 (2021) 269–287.

[11] J. Sun, W. Zheng, Q. Zhang, Z. Xu, Graph neural network encoding for community detection in attribute networks, IEEE Trans. Cybern. (2021).
[12] X. Chen, Z. Jiang, H. Li, J. Ma, P.S. Yu, Community hiding by link perturbation in social networks, IEEE Trans. Comput. Soc. Syst. 8 (2021) 704–715.
[13] S. Mittal, D. Sengupta, T. Chakraborty, Hide and seek: outwitting community detection algorithms, IEEE Trans. Comput. Soc. Syst. 8 (2021) 799–808.
[14] X. Liu, L. Fu, X. Wang, J.E. Hopcroft, Prohico: A probabilistic framework to hide communities in large networks, in: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, IEEE, 2021, pp. 1–10.
[15] S. Nagaraja, The impact of unlinkability on adversarial community detection: effects and countermeasures, in: International Symposium on Privacy Enhancing Technologies Symposium, Springer, 2010, pp. 253–272.
[16] M. Waniek, T.P. Michalak, M.J. Wooldridge, T. Rahwan, Hiding individuals and communities in a social network, Nat. Human Behav. 2 (2018) 139–147.
[17] V. Fionda, G. Pirro, Community deception or: How to stop fearing community detection algorithms, IEEE Trans. Knowl. Data Eng. 30 (2017) 660–673.
[18] J. Chen, L. Chen, Y. Chen, M. Zhao, S. Yu, Q. Xuan, X. Yang, Ga-based q-attack on community detection, IEEE Trans. Comput. Soc. Syst. 6 (2019) 491–503.
[19] Y. Liu, J. Liu, Z. Zhang, L. Zhu, A. Li, Rem: From structural entropy to community structure deception, Adv. Neural Inf. Process. Syst. 32 (2019) 12938–12948.
[20] J. Chen, Y. Chen, L. Chen, M. Zhao, Q. Xuan, Multiscale evolutionary perturbation attack on community detection, IEEE Trans. Comput. Soc. Syst. (2020).
[21] J. Li, H. Zhang, Z. Han, Y. Rong, H. Cheng, J. Huang, Adversarial attack on community detection by hiding individuals, Proceedings of The Web Conference 2020 (2020) 917–927.
[22] J.C.-W. Lin, Y. Djenouri, G. Srivastava, U. Yun, P. Fournier-Viger, A predictive ga-based model for closed high-utility itemset mining, Appl Soft Comput 108 (2021) 107422.
[23] O.A. Arqub, Z. Abo-Hammour, Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm, Inf. Sci. 279 (2014) 396–415.
[24] Z. Abo-Hammour, O. Alsmadi, S. Momani, O. Abu Arqub, A genetic algorithm approach for prediction of linear dynamical systems, Math. Problems Eng. (2013).
[25] Z. Abo-Hammour, O. Abu Arqub, S. Momani, N. Shawagfeh, Optimization solution of troesch's and bratu's problems of ordinary type using novel continuous genetic algorithm, Discrete Dynamics in Nature and Society 2014 (2014).
[26] O. Abu Arqub, Z. Abo-Hammour, S. Momani, N. Shawagfeh, Solving singular two-point boundary value problems using continuous genetic algorithm, in: Abstract and applied analysis, vol. 2012, Hindawi, 2012.
[27] M. Tasgin, A. Herdagdelen, H. Bingol, Community detection in complex networks using genetic algorithms, arXiv preprint arXiv:0711.0491 (2007).
[28] H. Liu, X.-B. Hu, S. Yang, K. Zhang, E. Di Paolo, Application of complex network theory and genetic algorithm in airline route networks, Transp. Res. Record 2214 (2011) 50–58.
[29] R. Shang, J. Bai, L. Jiao, C. Jin, Community detection based on modularity and an improved genetic algorithm, Phys. A 392 (2013) 1215–1231.
[30] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, J. Stat. Mech.: Theory Exp. 2005 (2005) P09008.
[31] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (2002) 7821–7826.
[32] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech.: Theory Exp. 2008 (2008) P10008.
[33] A. Clauset, M.E. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (2004) 066111.
[34] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure, Proc. Natl. Acad. Sci. 105 (2008) 1118–1123.
[35] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (2007) 036106.
[36] M.E. Newman, Detecting community structure in networks, Eur. Phys. J. B 38 (2004) 321–330.
[37] M.J. Rattigan, M. Maier, D. Jensen, Graph clustering with network structure indices, in: Proceedings of the 24th international conference on Machine learning, 2007, pp. 783–790.
[38] M.E. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. 103 (2006) 8577–8582.
[39] M. Okuda, S. Satoh, Y. Sato, Y. Kidawara, Community detection using restrained random-walk similarity, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2019) 89–103.
[40] S. Lim, J. Kim, J.-G. Lee, Blackhole: Robust community detection inspired by graph drawing, in: 2016 IEEE 32nd International Conference on Data Engineering (ICDE), IEEE, 2016, pp. 25–36.
[41] V. Moscato, A. Picariello, G. Sperlí, Community detection based on game theory, Eng. Appl. Artif. Intell. 85 (2019) 773–782.
[42] A. Goldenberg, A.X. Zheng, S.E. Fienberg, E.M. Airoldi, A survey of statistical network models, Found. Trends, Mach. Learn. 2 (2009) 129–233.
[43] B. Karrer, M.E. Newman, Stochastic blockmodels and community structure in networks, Phys. Rev. E 83 (2011) 016107.
[44] J.H. Holland et al, Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, MIT Press (1992).
[45] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[46] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE, 1995, pp. 39–43.