

Research Article

Heuristic Artificial Bee Colony Algorithm for Uncovering Community in Complex Networks

Yuquan Guo,¹ Xiongfei Li,¹ Yufei Tang,² and Jun Li^{3,4}

¹College of Computer Science and Technology, Jilin University, Changchun 130012, China

²Computer & Electrical Engineering and Computer Science Department, Florida Atlantic University, Boca Raton, FL 33431, USA

³School of Management Science and Information Engineering, Jilin University of Finance and Economics, Changchun 130117, China

⁴Department of Applied Mathematics, Changchun University of Science and Technology, Changchun 130022, China

Correspondence should be addressed to Xiongfei Li; xiongfei@jlu.edu.cn

Received 21 July 2016; Revised 23 October 2016; Accepted 17 November 2016; Published 18 January 2017

Academic Editor: Mauro Gaggero

Copyright © 2017 Yuquan Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Community structure is important for us to understand the functions and structure of the complex networks. In this paper, Heuristic Artificial Bee Colony (HABC) algorithm based on swarm intelligence is proposed for uncovering community. The proposed HABC includes initialization, employed bee searching, onlooker searching, and scout bee searching. In initialization stage, the nectar sources with simple community structure are generated through network dynamic algorithm associated with complete subgraph. In employed bee searching and onlooker searching stages, the searching function is redefined to address the community problem. The efficiency of searching progress can be improved by a heuristic function which is an average agglomerate probability of two neighbor communities. Experiments are carried out on artificial and real world networks, and the results demonstrate that HABC will have better performance in terms of comparing with the state-of-the-art algorithms.

1. Introduction

A community is thought of as a set of nodes where nodes are densely interconnected and sparsely linked to other parts of the networks, producing a community structure of complex networks like that sketched in Figure 1. In the case, there are three communities which are denoted by the dashed circles. A community is a set of nodes with a high density of internal links, whereas links between communities have a comparatively lower density.

In real world, many complex systems can be represented in terms of networks. For example, social networks are widely observed in our lives, because people naturally are clustered to form communities, within their work environment, family, and friends [1]. Another example is the proteins network [2], where they are interacted very frequently with each other, as they belong to metastatic cells, which have a high motility and invasiveness with respect to normal cells. Communities of proteins networks correspond to functional groups. The web is a network of HTML pages interconnected by hyperlinks. Communities of the web networks [3] are groups of pages

having topical similarities. Internet is described by service and client as nodes and the connection between them by TCP or UDP. Communities of the Internet are groups of applications with similar function. The research of community structure in complex networks has positive theoretical and practical effects on analyzing topology structure of complex networks, understanding the functions of complex networks, detecting potential patterns of complex networks, and forecasting behaviors of complex networks.

The rest of this paper is organized as follows: the related works are discussed in detail in Section 2. Section 3 describes HABC, including overall framework of HABC, colony and nectar sources initialization, and heuristic searching progress. In Section 4, the experiments are carried out to verify the proposed method, and then in Section 5, conclusions are drawn.

2. Related Work

Many methods have been proposed recently to uncover the community structure of complex networks. For reviews

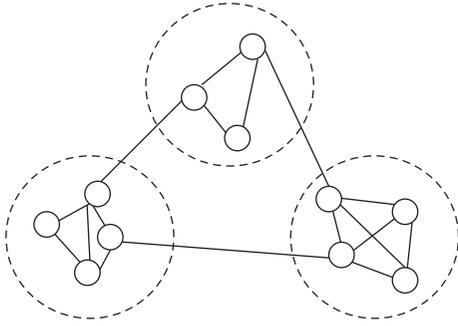


FIGURE 1: A community structure of small networks.

and comparison, one can refer to references [4–8]. The community detection problem is challenging in part because it is not very well posed [9]. The essence of the community detection problem is a node-clustering problem of complex networks. To resolve it, various approaches have been developed in recent years, including hierarchical clustering, dynamic algorithm, and optimization. Most variants of the graph partitioning problem are NP-hard; that is, it is unlikely that the solution can be computed in a time growing as a power of the graph size [4]. For example, network community structure detection can be modeled as modularity maximization problems, but unfortunately, it has been proved that modularity optimization is a NP-hard problem, so it is probably impossible to find the solution in a time growing polynomially with the size of the graph.

Hierarchical clustering is the first kind approach to detect community structure. Hierarchical clustering algorithms create a dendrogram according to the similarity between two groups of nodes and then uncover the community structure through cutting the dendrogram into subdendrogram. Hierarchical clustering algorithms can be classified into two categories: agglomerative algorithms and divisive algorithms. Agglomerative algorithms mainly include EAGLE [10], CPM [11], MCC [12], and CDHC [13]. Divisive algorithms mainly include GN [14] by Girvan and Newman, BCA [15] by Tyler et al., and local algorithm [16] by Radicchi et al. Hierarchical clustering has the advantage that it does not require a preliminary knowledge of community structure. However, the efficiency of these algorithms is poor in the procedure of cutting the dendrogram.

Dynamic algorithms employ processes running on the networks. Random walk [17–19] and label propagation are often applied to detect communities in dynamic algorithms. A drawback of random walks algorithm is the fact that the final partition is strongly dependent on the choice of the parameter. Therefore the most meaningful partitions cannot be obtained. Label propagation includes LPA [20], CNP [21], k -rank [22], COPRA [23], and BMLPA [24]. Label propagation is near linear time algorithm. However, the result of label propagation algorithm is not stable.

The methods in the third category produce communities by optimization [25]. In the community problem, the number of communities and the size of each community are unknown. For example, in a biological network, community

structure is not observed. In order to evaluate the quality of a partition, many graph clustering indexes have been proposed [26–28]. Clustering indexes optimization becomes a very effective method to detect communities. Modularity proposed by Newman and Girvan is the most popular in these indexes [29]. Optimization of modularity includes hierarchical optimization of modularity by Blondel et al. [30], extremal-optimization-based algorithm that optimizes the modularity [31], and optimization modularity via mean field annealing [32]. However, Brandes found that the unknown complexity status of modularity maximization [33–35] is NP-complete in the strong sense [36]. Therefore, the study of modularity optimization has attracted attention recently. GA is applied to discover community [37–40]. The advantage of GA is the rapidity of convergence and universality. However, premature convergence is the shortcoming of GA, and the speed of convergence is slow in later iteration process. Swarm intelligence based computation is an important approach of address optimization problem, and it mainly focuses on the collective behavior of decentralized, self-organized systems [41]. Ant colony optimization algorithms [42–44] are applied to detect community in complex networks. Our method which is named Heuristic Artificial Bee Colony (HABC) algorithm is this kind. The global optimization process of HABC is accelerated by positive feedback of bee colony optimizing.

In this paper, we apply Artificial Bee Colony optimization algorithm to detect community structure. The algorithm is named HABC. The original Artificial Bee Colony (ABC) uses random searching [45], so it will lead to a time consuming search process for community detection problems. Therefore, based on ABC, the proposed HABC introduces a new heuristic function to the search process, in order to find the local optimal component according to neighbor communities. The HABC includes initialization and three searching processes. In initialization phase, every nectar source gets simply community structure through using network dynamic algorithm. Employed bees and onlookers use a heuristic function to guide the searching process.

The key contributions of this paper can be summarized as follows: (1) proposing a Heuristic Artificial Bee Colony based community detection algorithm; (2) redefining a searching progress of Artificial Bee Colony algorithm to address community problem; (3) using an agglomerate probability of two neighbor communities as heuristic function of searching progress.

3. Heuristic Artificial Bee Colony Algorithm

Before we introduce the proposed HABC, we would like to briefly introduce some basic concepts in network. A network can be modeled as a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. A is the adjacency matrix of G . A_{ij} is 1 if there is an edge from node i to node j and 0 for otherwise. A community in a network is a group of nodes where nodes are densely interconnected and sparsely linked to other parts of the networks. Community structure can be represented by a set $P = \{C_1, \dots, C_k\}$, where k is the number of communities and C_i is a community which is a set of nodes

```

Input: a complex network  $G = (V, E)$ ;
Output: the set of communities  $C$ ;
(1) Initial parameters: NNS, MCN, limit;
(2) NS, EB, OB  $\leftarrow$  Initialization();
(3) For  $i = 1$  to MCN {
    For each employed bee of EB{
        EmployedBeeSearch();
    }
    Calculate the probability value of the nectar sources;
    For each onlooker bee of OE{
        OnlookerBeeSearch();
    }
    If there is an abandoned solution for the scout then
        NS  $\leftarrow$  NS  $\cup$  ScoutBeeSearch();
        C  $\leftarrow$  the best solution so far;
}
(4) return C;

```

ALGORITHM 1: Framework of HABC.

with the same community ID and P is regarded as a partition of the network.

3.1. Framework of HABC. Artificial Bee Colony (ABC) algorithm [45] simulates the behavior of a honey bee swarm for solving optimization problems. The model consists of two essential components: nectar source and bee colony. A nectar source represents a possible solution to the optimization problem. The nectar amount of a nectar source corresponds to the quality of the solution. The colony of bees consists of three groups of bees: employed bees, onlookers, and scouts. Employed bees are associated with a particular nectar source which is exploited by employed bees. Onlookers search nectar source which is shared by employed bees. Scouts search new nectar source when a nectar source is abandoned. The search process of ABC is directed by cooperation and conversion of bee colony. ABC is shown as a flowchart on Figure 2.

Based on ABC, the proposed HABC introduces a new heuristic function to the search process. The framework of HABC algorithm is shown in Algorithm 1. HABC algorithm consists of three processes including setting parameters, initialization, and searching processes. Firstly, the following parameters need to be initialized: the number of nectar source (NNS), the maximum cycle number (MCN), and the *limit* for abandoning nectar source. Secondly, initialization of HABC generates nectar source set NS, employed bee set EB, and onlooker bee set OB. A nectar source represents a solution of community problem. The number of employed bees is equal to the number of nectar sources. Thirdly, the search processes of the employed bees, the onlooker bees, and the scout bees are repeated until MCN is reached.

3.2. Colony and Nectar Source Initialization. In initialization period, HABC generates a nectar source set NS of NSN solutions, where a solution represents a partition of complex networks. Each solution $S_i = \{CID_{i1}, CID_{i2}, \dots, CID_{ij}, \dots, CID_{in}\}$, where $i = 1, 2, \dots$, NNS is an N -dimensional vector.

n is the number of nodes of complex network G . The component CID_{ij} is community ID of node j .

The initialization has two stages. In the first stage, all complete subgraphs of complex networks are extracted by *Bron-Kerbosch* algorithm [46]. The nodes in the same complete subgraph are assigned unique community ID. Other nodes of complex networks are randomly assigned unique community ID. We can get a nectar source with simple community structure. In the second stage, the nectar sources are generated though repeatedly running LPA [20] on the nectar source which is got in the first stage. Each nectar source gets simple community structure through initialization, and the nectar source set is diversity.

3.3. Heuristic Searching Progress. The nectar amount of nectar source corresponds to the quality (fitness) of the associated community solution. In HABC, the fitness function is defined by the following equation:

$$\text{fitness}(P) = \frac{1}{1 + C(P)}, \quad (1)$$

where P is a partition of the network and $C(P)$ is the conductance [17] of the network. The conductance of a network, which reflects how easily the diffusion occurs among different communities, is obtained by the following:

$$C(P) = \frac{1}{k} \sum_{i=1}^k \left(1 - \frac{\text{in_vol}(C_i)}{\text{vol}(C_i)} \right), \quad (2)$$

where $\text{vol}(C_i)$ is volume of community C_i and $\text{in_vol}(C_i)$ is referred to as the inward volume of C_i . $\text{vol}(C_i)$ is defined as (3) and $\text{in_vol}(C_i)$ is defined as (4):

$$\text{vol}(C_i) = \sum_{x \in C_i, y} A_{xy} \quad (3)$$

$$\text{in_vol}(C_i) = \sum_{x \in C_i, y \in C_i} A_{xy}. \quad (4)$$

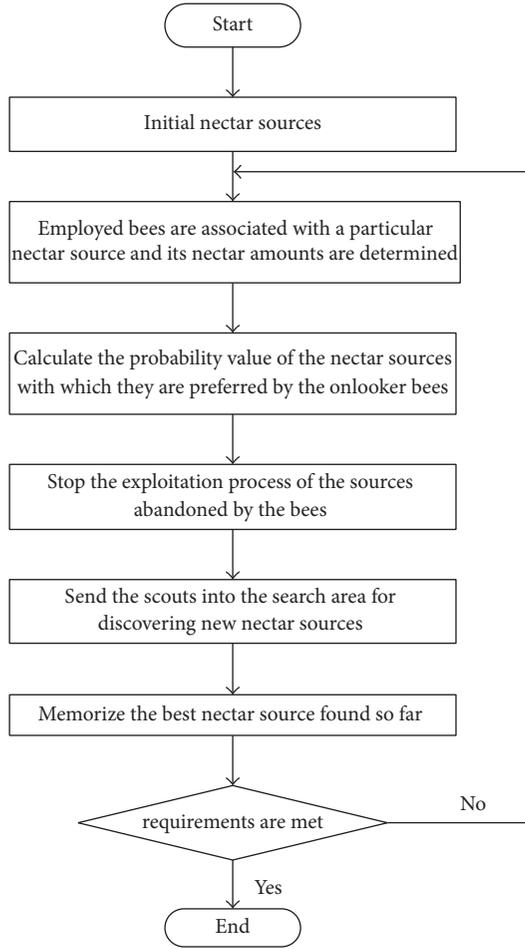


FIGURE 2: Flowchart of the ABC algorithm.

The conductance of network is the average departure probability of the whole community. It reflects the diffusion capacity among different communities. The value of conductance is in the range from 0 to 1. High value of conductance indicates a strong community structure.

The standard ABC algorithm uses (5) to generate a new nectar source position from the old one in the searching progress of bee colony, where k is a random nectar source index. k is different from i , and U_i , X_i , X_k are vectors which represent nectar sources. X_i represents current nectar source. U_i is new nectar source. X_k is a nectar source nearby X_i . The searching progress is accomplished by the component optimization of nectar source vector.

$$U_i = X_i + \text{rand}(-1, 1)(X_i - X_k). \quad (5)$$

In HABC, the component CID_{ij} of nectar source S_i is community ID. The searching progress of employed bee and onlooker is the components optimization of a nectar source according to neighbor community ID of components. So the searching function of HABC is adapted as (6), where NCI_1, \dots, NCI_m are neighbor community ID of node j . In

the searching progress of HABC, each component is updated according neighbor community ID.

$$CID_{ij} = \text{rand}(NCI_1, \dots, NCI_m). \quad (6)$$

The conductance of networks, which is the objective function of HABC, is average departure probability of all communities. So we can use the average agglomerate probability of two adjacent communities as the heuristic function of searching progress which is defined as follows:

$$h(C_i, C_j) = \frac{1}{2} \left(\frac{\text{in_vol}(C_i)}{\text{vol}(C_i)} + \frac{\text{in_vol}(C_j)}{\text{vol}(C_j)} \right). \quad (7)$$

Equation (7) can convert agglomerate probability of a node and a community as (8), where n is a node of networks. $C_{(n)}$ is the community in which n is a node. Equation (8) is applied as heuristic function to searching progress of employed bee, (6). A new search function is proposed as (9). The neighbor community ID set of node i is computed by $Ln(i)$. C_L is a community with community ID L .

$$h(n, C_j) = \frac{1}{2} \left(\frac{\text{in_vol}(C_{(n)})}{\text{vol}(C_{(n)})} + \frac{\text{in_vol}(C_j)}{\text{vol}(C_j)} \right) \quad (8)$$

$$CDI_i = \arg \max_L \{h(i, C_L), L \in Ln(i)\}. \quad (9)$$

For every nectar source, there is only one employed bee. The employed bee is sent into the nectar source with which it is associated. Each component of nectar source is optimized in the employed bee searching progress. Employed bee searching algorithm is proposed as in Algorithm 3. A component of the nectar source is randomly selected and is updated according to (9). The employed bee searching progress is repeated until each component is updated.

Onlooker searching progress is similar to the employed bee searching progress. But onlookers select nectar source according the probability which is calculated as (10). The probability is information about nectar source which is shared by employed bee. Searching progress of onlooker is important to distinguish HABC from other evolutionary algorithms. That is a greed searching algorithm which makes HABC fast convergence. Onlooker bee searching is proposed as Algorithm 4.

$$\text{prob}(i) = \frac{C(P_i)}{\sum_{i=1}^{\text{NNS}} C(P_i)}. \quad (10)$$

The employed bee whose nectar source has been exhausted by the bees becomes a scout. The scout searches the area surrounding hive for new nectar source. In line 4 of Algorithm 2, the vector S is created. The scout bee searching runs LPA on S PN times. The scout bee searching progress is proposed as Algorithm 5.

4. Experiment and Analysis

In order to test performance of the proposed HABC, we do different experiments on both real and artificial synthetic networks. EAGLE is a classic hierarchical clustering

Input: a complex networks $G = (V, E)$, the number of nectar source NNS, the number of label propagation NLP;
Output: the set of nectar source NS, the set of employed bee EB, the set of onlooker bee OB;

- (1) create all complete subgraphs of complex networks G ;
- (2) the nodes in same subgraph are assigned unique community ID;
- (3) the other nodes are assigned different community ID;
- (4) the vector $S = \{CID_1, CID_2, \dots, CID_n\}$ is generated where CID_i is community ID of node i ;
- (5) $NS = \emptyset, EB = \emptyset, OB = \emptyset$;
- (6) For $i = 1$ to NNS{
 - $S_i \leftarrow \text{LPA}(S, \text{NLP})$;
 - $\text{EmployBee} \leftarrow \text{CreateEmployedBee}(S_i)$;
 - $EB \leftarrow EB \cup \{\text{EmployBee}\}$;
 - $\text{OnlookerBee} \leftarrow \text{CreateOnlooker}(S_i)$;
 - $OB \leftarrow OB \cup \{\text{OnlookerBee}\}$;
 - $NS \leftarrow NS \cup \{S_i\}$;
- (7) return NS, EB, OB;

ALGORITHM 2: Initialization of HABC.

Input: employed bee ID EBID;
Output: the nectar Source NS

- (1) For $j = 1$ to $N\{$
 - A node n which is not labeled is randomly selected.
 - The neighbor community set Ln of n is calculated.
 - $NS_{[\text{EBID}].x_n} = \arg \max_L \{h(i, C_L), L \in Ln(i)\}$;
 - The node n is labeled;
- (2) return NS;

ALGORITHM 3: Employed bee search.

community detection algorithm [10]. COPRA is an available method in the class of dynamic algorithms [23]. GCE is a local optimization algorithm [25] and HABC belongs to local optimization associated with globe target. Therefore, EAGLE, COPRA, and GCE are selected as compared methods. We use NMI [26] to evaluate the difference between the community structure uncovered by HABC method and the true community structure. NMI is defined as (11). It is based on defining a confusion matrix N , where the rows correspond to the “real” communities and the columns correspond to the “detected” communities. N_{ij} is simply the number of nodes in the real community i that appear in the found community j . The number of real communities is denoted by C_A and the number of found communities is denoted by C_B . The sum over row i of matrix N_{ij} is denoted by $N_{i.}$, and the sum over column j is denoted by $N_{.j}$. The value of NMI is between 0 and 1. A higher NMI indicates that the detected communities more perfectly match the true communities.

$$\text{NMI} = \frac{-2 \sum_{i=1}^{C_A} \sum_{j=1}^{C_B} C_{ij} \log \left(\frac{N_{ij} N}{N_{i.} N_{.j}} \right)}{\sum_{i=1}^{C_A} N_{i.} \log \left(\frac{N_{i.}}{N} \right) + \sum_{j=1}^{C_B} N_{.j} \log \left(\frac{N_{.j}}{N} \right)}. \quad (11)$$

4.1. Artificial Networks. We apply HABC to the LFR benchmark [47] which is networks with various distributions of vertex degree and community size. There are many parameters to control the created networks: the number of nodes N , the average vertex degree k , the maximum vertex degree \max_k , the mixing ratio μ , the exponent of the power law distribution of vertexes, the exponent of the power law distribution of community size, the minimum community size \min_c , and the maximum community size \max_c . In our experiment, we generate six network sets. The parameters of six network sets are shown in Table 1. The default configuration of LFR benchmark is $\langle k \rangle = 15$ and $\max_k = 50$.

As shown in Figure 3, each graph shows a comparative result of four methods on one network set which is generated by adjusting values of mixing ratio μ ($0 \leq \mu \leq 1$). When the community structure is evident ($0.1 \leq \mu \leq 0.3$), both HABC and other three methods can accurately detect the community structure. But the higher NMI value for HABC clearly illustrates that it is superior to other three algorithms when $0.1 \leq \mu \leq 0.3$. As can be seen in Figure 3(d), the NMI value of HABC is higher than other algorithms when $0.4 \leq \mu \leq 0.5$. But the effect of CPM, EAGLE, and GCE declines obviously when $\mu > 0.4$.

In order to explore the accuracy of different algorithms, the number of standard communities in the benchmark is compared with the number of detected communities by four algorithms. As shown in Figure 4, the cyan represents the number of standard communities each network sets. The black, red, green, and blue represent the number of uncovered correct communities by EAGLE, CPM, GCE, and HABC, respectively. We found that when μ (Mix) increases, the accuracy of these four algorithms declines. As can be seen from Figures 1 and 2, we can find inherent relationship between the accuracy of four algorithms and the number of uncovered communities. The closer the number of uncovered communities is to standard communities, the higher the NMI value is.

```

Input: the nectar source set NS;
Output: the updated set of nectar source NS;
(1) calculate  $prob()$  of every nectar source;
(2) select a nectar source which  $prob()$  is maximum, and get EBID of the nectar source;
(3) For  $i = 1$  to  $N$ {
    Node  $n$  which is not labeled is randomly selected.
    The neighbor community set NCI of  $n$  is calculated.
     $NS_{[EBID]}.x_n = \arg \max_L \{h(i, C_L), L \in Ln(i)\}$ ;
    The node  $n$  is labeled;
}
(4) return NS;

```

ALGORITHM 4: Onlooker bee search.

TABLE 1: Parameters for artificial networks.

Network set	Nodes N	Mixing ratio μ	Community size range	
			min_c	max_c
N1	1000	0.1, 0.2, 0.3, 0.4, 0.5	10	50
N2	1000	0.1, 0.2, 0.3, 0.4, 0.5	30	100
N3	3000	0.1, 0.2, 0.3, 0.4, 0.5	10	50
N4	3000	0.1, 0.2, 0.3, 0.4, 0.5	30	100
N5	5000	0.1, 0.2, 0.3, 0.4, 0.5	10	50
N6	5000	0.1, 0.2, 0.3, 0.4, 0.5	30	100

```

Input: S, PN;
Output: the set of communities C
(1)  $P_i \leftarrow LPA(S, PN)$ ;
(2)  $EmployBee \leftarrow CreateEmployedBee(P_i)$ ;
(3)  $EB \leftarrow EB \cup \{EmployBee\}$ ;
(4) return  $\{P_i\}$ ;

```

ALGORITHM 5: Scout bee search.

In order to inspect the influences of community size range on different algorithms, the following experiment is conducted. The performance of each algorithm on network set N1 ($\mu = 0.3$, $min_c = 10$, $max_c = 30$) and N2 ($\mu = 0.3$, $min_c = 30$, $max_c = 100$) is compared. The performance of each algorithm on networks N3 ($\mu = 0.3$, $min_c = 10$, $max_c = 30$) and N4 ($\mu = 0.3$, $min_c = 30$, $max_c = 100$) is compared. The performance of each algorithm on networks N5 ($\mu = 0.3$, $min_c = 10$, $max_c = 30$) and N6 ($\mu = 0.3$, $min_c = 30$, $max_c = 100$) is compared. To investigate the effects of network size on different algorithms, we compare the performance of each method on network sets N1 ($\mu = 0.3$, $min_c = 10$, $max_c = 30$), N3 ($\mu = 0.3$, $min_c = 10$, $max_c = 30$), and N5 ($\mu = 0.3$, $min_c = 10$, $max_c = 30$). As can be seen from Figures 5 and 6, HABC has a better performance for different network size and different community range size compared with CPM, GCE, and EAGLE.

TABLE 2: Summary of real world network.

Network	Nodes	Edges	Community
Karate	34	78	2
Dolphins	62	159	2
Football	115	613	12
Polbooks	105	441	3

4.2. Real World Networks. HABC is further tested on many real world networks which are shown in Table 2. We select Zachary's karate club network [48], dolphin network [49], American college football network [14], and books on US political network (Polbooks) [50]. For these networks, we cannot use NMI to evaluate performance of four algorithms, because the true community structures of these networks are unknown. Table 3 shows the comparison results of Q values of HABC, EAGLE, GCE, and CPM. The performance of HABC outperforms EAGLE, GCE, and CPM.

5. Conclusion

In this paper, Heuristic Artificial Bee Colony (HABC) algorithm is proposed for community detection. The proposed HABC is inspired by the intelligent behaviors of the bee swarm and contains four important sections, which are initialization, employed bee searching, onlooker searching, and scout searching stages. Combining LPA and Bron-Kerbosch algorithm, the initialization stage generates nectar

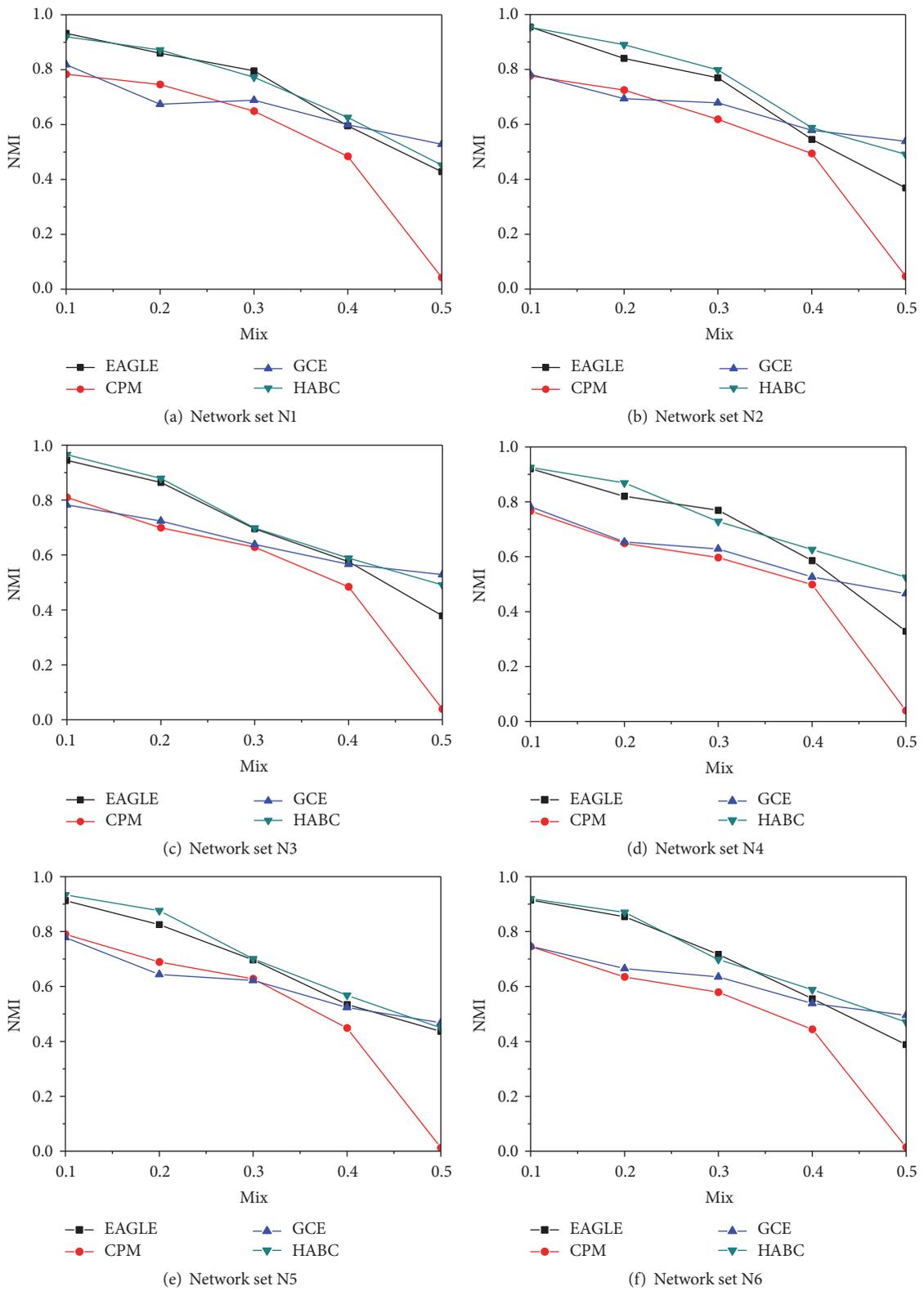


FIGURE 3: Comparative NMI value for different algorithms on six network sets.

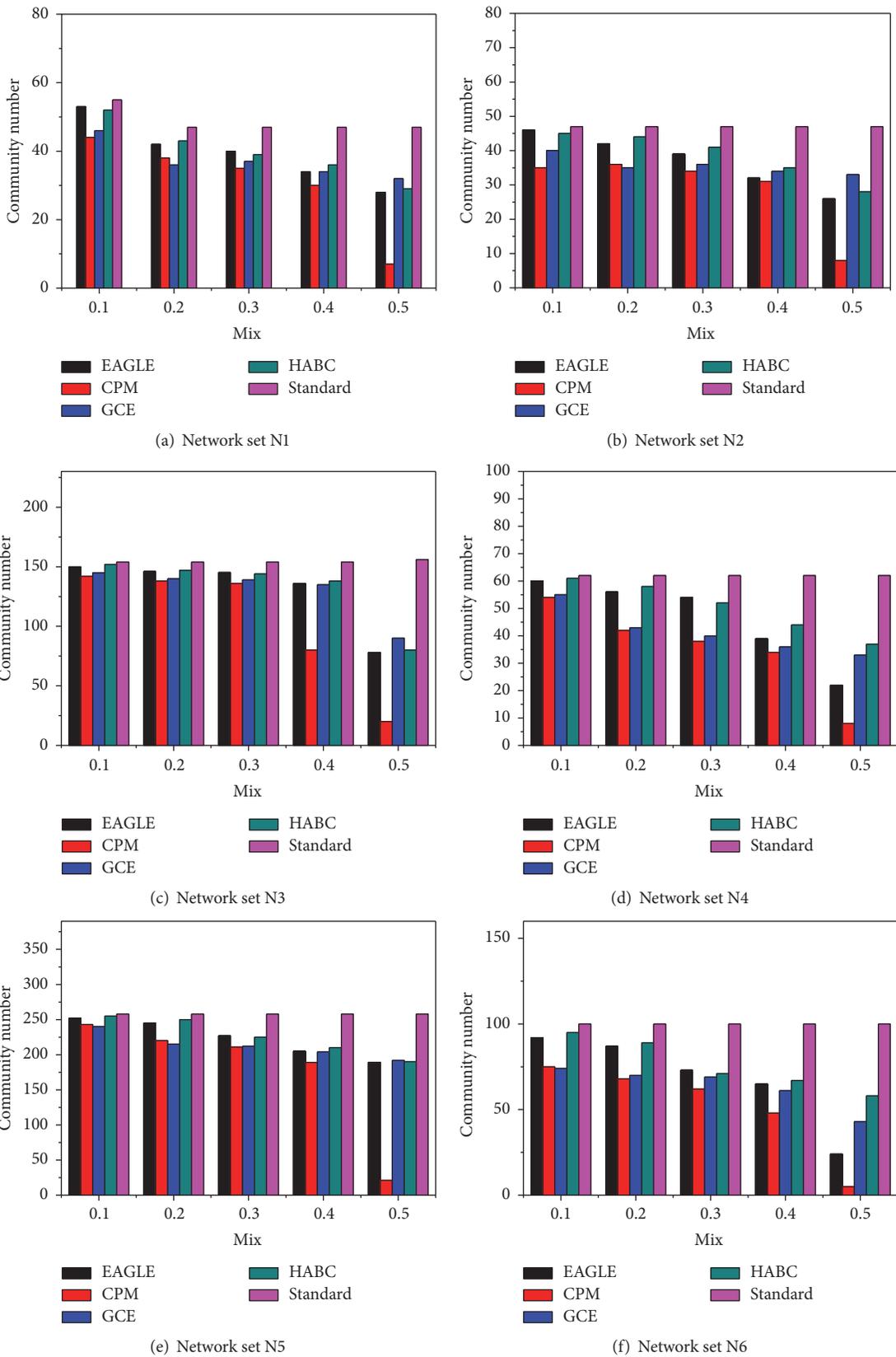


FIGURE 4: The number of uncovered communities by four algorithms on six networks.

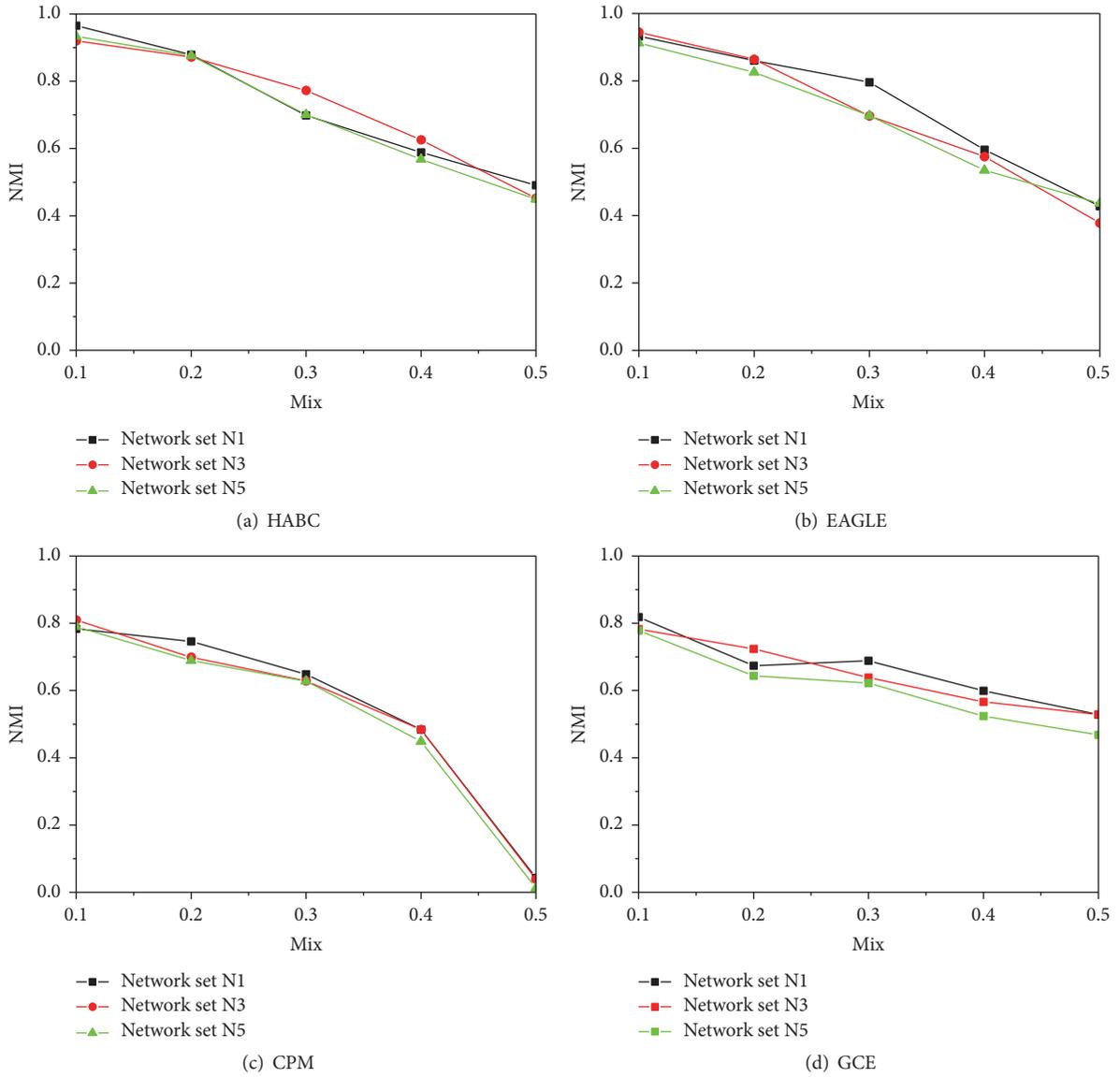


FIGURE 5: The effects of network size on four algorithms.

TABLE 3: Comparisons of Q value of different algorithms on real world networks.

Network	EAGLE	CPM	GCE	HABC
Karate	0.4019	0.3723	0.3824	0.4104
Dolphins	0.5197	0.4708	0.4896	0.5298
Football	0.5994	0.5764	0.5769	0.6081
Polbooks	0.5098	0.4478	0.4687	0.5103

sources to which an employed bee is assigned. Each nectar source represents a solution of community. Then the nectar source is explored by the employed bee and the onlooker. In exploring progress, a newly defined heuristic function is applied to guide searching nectar source. A nectar source will be abandoned when it is exhausted, and the scout is sent for discovering new nectar sources. Experiments are carried out on artificial and real networks, and the performance of the HABC has been analyzed in detail. Simulation results

on six artificial network sets demonstrate that HABC has better performance at community detection quality with much higher NMI value than that of EAGLE, CPM, and GCE. The experimental results demonstrate that HABC has better performance at community detection quality aspect.

Competing Interests

The authors declare that they have no competing interests.

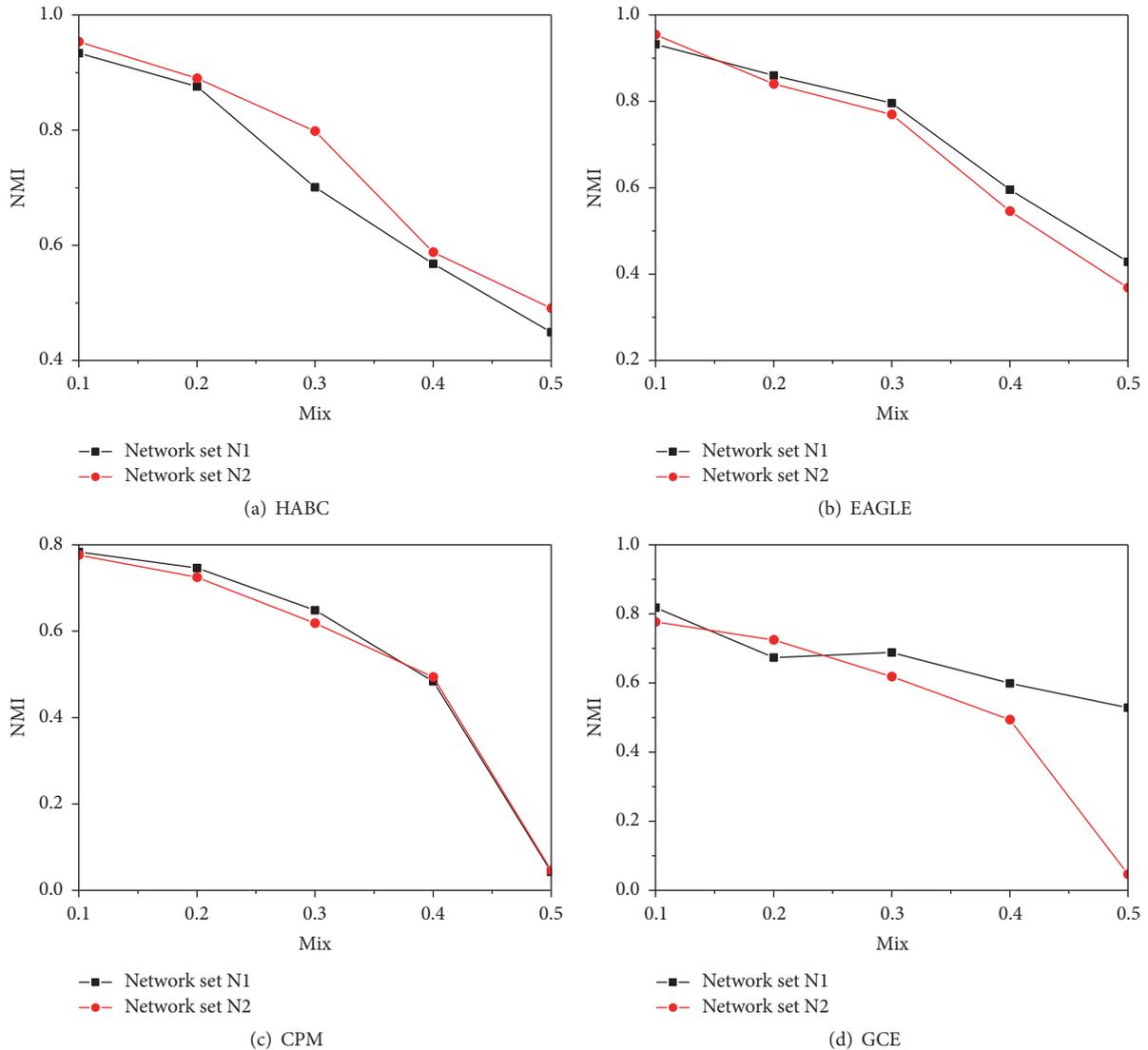


FIGURE 6: The influences of community size range on four algorithms.

Acknowledgments

The work was supported by the National Science & Technology Pillar Program, China (Grant no. 2012BAH48F02), National Science Foundation of China (Grant no. 61272209), Technology Development Plan of Jilin Province (Grant no. 201105017), and Agreement of Science & Technology Development Project, Jilin Province (Grant no. 20150101014JC), Science and Technology Development Program of Jilin Province of China (Natural Science Foundation of Jilin Province, Grants nos. 20140101199JC and 20150101053JC).

References

- [1] D. J. Watts, P. S. Dodds, and M. E. J. Newman, "Identity and search in social networks," *Science*, vol. 296, no. 5571, pp. 1302–1305, 2002.
- [2] L. Royer, M. Reimann, A. F. Stewart, and M. Schroeder, "Network compression as a quality measure for protein interaction networks," *PLoS ONE*, vol. 7, no. 6, Article ID e35729, 2012.
- [3] D. Gibson, J. Kleinberg, and P. Raghavan, "Inferring web communities from link topology," in *Proceedings of the 9th ACM Conference On Hypertext And Hypermedia: Links, Objects, Time and Space—Structure in Hypermedia Systems: Links, Objects, Time and Space—Structure in Hypermedia Systems*, pp. 225–234, Pittsburgh, Pa, USA, 1998.
- [4] S. Fortunato and C. Castellano, "Community structure in graphs," in *Computational Complexity*, pp. 490–512, Springer, New York, NY, USA, 2012.
- [5] G. K. Orman, V. Labatut, and H. Cherifi, "Comparative evaluation of community detection algorithms: a topological approach," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2012, no. 8, Article ID P08001, 2012.
- [6] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," *Physical Review E*, vol. 80, no. 5, Article ID 056117, 2009.

- [7] B. Yang, D.-Y. Liu, J. Liu, D. Jin, and H.-B. Ma, "Complex network clustering algorithms," *Journal of Software*, vol. 20, no. 1, pp. 54–66, 2009.
- [8] Q. Cai, L. Ma, M. Gong, and D. Tian, "A survey on network community detection based on evolutionary computation," *International Journal of Bio-Inspired Computation*, vol. 8, no. 2, pp. 84–98, 2016.
- [9] M. E. J. Newman, "Communities, modules and large-scale structure in networks," *Nature Physics*, vol. 8, no. 1, pp. 25–31, 2012.
- [10] H. W. Shen, X. Q. Cheng, K. Cai, and M.-B. Hu, "Detect overlapping and hierarchical community structure in networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 388, no. 8, pp. 1706–1712, 2009.
- [11] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.
- [12] M.-S. Shang, D.-B. Chen, and T. Zhou, "Detecting overlapping communities based on community cores in complex networks," *Chinese Physics Letters*, vol. 27, no. 5, Article ID 058901, 2010.
- [13] C. Yin, S. Zhu, H. Chen, B. Zhang, and B. David, "A method for community detection of complex networks based on hierarchical clustering," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 849140, 9 pages, 2015.
- [14] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [15] J. R. Tyler, D. M. Wilkinson, and B. A. Huberman, "E-mail as spectroscopy: automated discovery of community structure within organizations," *The Information Society*, vol. 21, no. 2, pp. 143–153, 2005.
- [16] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Paris, "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, pp. 2658–2663, 2004.
- [17] X.-Q. Cheng and H.-W. Shen, "Uncovering the community structure associated with the diffusion dynamics on networks," *Journal of Statistical Mechanics-Theory and Experiment*, vol. 2010, no. 4, Article ID P04024, 2010.
- [18] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 4, pp. 1118–1123, 2008.
- [19] D. Jin, B. Yang, C. Baquero, D. Y. Liu, D. X. He, and J. Liu, "A Markov random walk under constraint for discovering overlapping communities in complex networks," *Journal of Statistical Mechanics-Theory and Experiment*, vol. 2011, no. 5, Article ID P05031, 2011.
- [20] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, Article ID 036106, 2007.
- [21] H. Lou, S. Li, and Y. Zhao, "Detecting community structure using label propagation with weighted coherent neighborhood propinquity," *Physica A: Statistical Mechanics and Its Applications*, vol. 392, no. 14, pp. 3095–3105, 2013.
- [22] Y. Jiang, C. Jia, and J. Yu, "An efficient community detection method based on rank centrality," *Physica A: Statistical Mechanics and Its Applications*, vol. 392, no. 9, pp. 2182–2194, 2013.
- [23] S. Gregory, "Finding overlapping communities in networks by label propagation," *New Journal of Physics*, vol. 12, Article ID 103018, 2010.
- [24] Z.-H. Wu, Y.-F. Lin, S. Gregory, H.-Y. Wan, and S.-F. Tian, "Balanced multi-label propagation for overlapping community detection in social networks," *Journal of Computer Science and Technology*, vol. 27, no. 3, pp. 468–479, 2012.
- [25] C. Lee, F. Reid, A. McDaid, and N. Hurley, "Detecting highly overlapping community structure by greedy clique expansion," <https://arxiv.org/abs/1002.1827>.
- [26] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 9, Article ID P09008, 2005.
- [27] K. Steinhaeuser and N. V. Chawla, "Identifying and evaluating community structure in complex networks," *Pattern Recognition Letters*, vol. 31, no. 5, pp. 413–421, 2010.
- [28] H.-W. Shen, X.-Q. Cheng, and J.-F. Guo, "Quantifying and identifying the overlapping community structure in networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2009, no. 7, Article ID P07042, 2009.
- [29] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, Article ID 026113, 2004.
- [30] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, Article ID P10008, 2008.
- [31] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical Review E*, vol. 72, no. 2, Article ID 027104, 2005.
- [32] S. Lehmann and L. K. Hansen, "Deterministic modularity optimization," *The European Physical Journal B*, vol. 60, no. 1, pp. 83–88, 2007.
- [33] U. Brandes, D. Delling, M. Gaertler et al., "On finding graph clusterings with maximum modularity," in *Proceedings of the 33rd International Conference on Graph-Theoretic Concepts in Computer Science (WG '07)*, Lecture Notes in Computer Science, pp. 121–132, Dornburg, Germany, 2007.
- [34] U. Brandes, D. Delling, M. Gaertler et al., "On modularity clustering," *IEEE Transactions on Knowledge & Data Engineering*, vol. 20, no. 2, pp. 172–188, 2008.
- [35] M. E. J. Newman, "Community detection in networks: modularity optimization and maximum likelihood are equivalent," *Physical Review E*, vol. 94, no. 5, Article ID 052315, 2016.
- [36] U. Brandes, D. Delling, M. Gaertler et al., "On modularity clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 2, pp. 172–188, 2008.
- [37] R. Shang, J. Bai, L. Jiao, and C. Jin, "Community detection based on modularity and an improved genetic algorithm," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 5, pp. 1215–1231, 2013.
- [38] Z. Wu, Y. Lin, H. Wan, S. Tian, and K. Hu, "Efficient overlapping community detection in huge real-world networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 391, no. 7, pp. 2475–2490, 2012.
- [39] C. Pizzuti, "A multiobjective genetic algorithm to find communities in complex networks," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 3, pp. 418–430, 2012.
- [40] Y.-Q. Guo, X.-F. Li, and X. Liu, "Heuristic genetic algorithm associated with spectral analysis uncovering multi-scale

- community of complex networks,” *Journal of Jilin University (Engineering and Technology Edition)*, vol. 45, no. 5, pp. 1592–1600, 2015.
- [41] H. Duan and Q. Luo, “New progresses in swarm intelligence-based computation,” *International Journal of Bio-Inspired Computation*, vol. 7, no. 1, pp. 26–35, 2015.
- [42] X. Zhou, Y. Liu, J. Zhang, T. Liu, and D. Zhang, “An ant colony based algorithm for overlapping community detection in complex networks,” *Physica A: Statistical Mechanics and Its Applications*, vol. 427, pp. 289–301, 2015.
- [43] J. Ji, X. Song, C. Liu, and X. Zhang, “Ant colony clustering with fitness perception and pheromone diffusion for community detection in complex networks,” *Physica A: Statistical Mechanics and Its Applications*, vol. 392, no. 15, pp. 3260–3272, 2013.
- [44] L. Ben Romdhane, Y. Chaabani, and H. Zardi, “A robust ant colony optimization-based algorithm for community mining in large scale oriented social graphs,” *Expert Systems with Applications*, vol. 40, no. 14, pp. 5709–5718, 2013.
- [45] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Tech. Rep. tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [46] C. Bron and J. Kerbosch, “Algorithm 457: finding all cliques of an undirected graph,” *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.
- [47] A. Lancichinetti, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms,” *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 78, no. 4, Article ID 046110, 2008.
- [48] 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, 1977.
- [49] D. Lusseau, “The emergent properties of a dolphin social network,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 270, no. 2, pp. S186–S188, 2003.
- [50] V. Krebs, 2014, <http://www.orgnet.com>.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

