Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 4, April 2019, pp. 1103-1111
DOI: 10.23940/ijpe.19.04.p5.11031111

Community Structure Division based on Immune Algorithm

Yuling Tian*

College of Information and Computer, Taiyuan University of Technology, Taiyuan, 030024, China

Abstract

Recently, the characteristics of complex networks and community structure have attracted attention from academia and society, and their research and applications have become increasingly important. Community structure division makes complex networks easy to understand. However, most community structure division methods often need the number of communities and have low efficiency. In this paper, an efficient method of community structure division in complex networks based on the immune algorithm is proposed. The method aims to find the core members of communities and classify other members according to core members. The individual evaluation of the core member is obtained by the affinity degree of the immune algorithm. In addition, the clone and mutation operation in the traditional immune algorithm is improved to be affected not only by the affinity but also by the iterative process. The improved immune algorithm can guarantee antibody diversity in the early stage of search and convergence in the later stage, and it then achieves faster convergence and higher precision of community structure division. Compared with traditional methods, the proposed method does not need the number of communities in advance, can get better results on real datasets, and has greater efficiency.

Keywords: complex network; community structure; community division; immune algorithm

(Submitted on November 10, 2018; Revised on December 13, 2018; Accepted on January 14, 2019)

1. Introduction

Community structure division helps researchers understand the organization of complex networks and plays an important role in network analysis. Lin et al. [1] investigated the influence of crops and hedged on soil microbial community structure and diversity. Zhang and Marbach [2] studied structures through mathematical modeling and analysis and formulated a generic model of a community in which each member decides how they allocate their time between content production and consumption with the objective of maximizing their individual reward. In reference [3], an optimal community structure that maximizes spreading dynamics was described, and a rich phase diagram with a triple point that separates the no-diffusion phase from the two diffusion phases was found.

In order to solve the problem of community division in complex networks, different measurement methods have been presented, and they can be divided into graph segmentation and hierarchical partitioning. The typical methods based on graph partition, such as the Kemighan-Lin algorithm and Spectral Bisection algorithm, usually need to specify the number and size of the community in advance, and they have limitations. By comparison, methods based on hierarchical partitioning do not depend on the number of communities but depend on the structure of complex networks, opening up a new way for the development of community division.

To measure the relationship strength between nodes, several methods based on hierarchical partitioning are presented from different views. They cluster nodes according to the relationship strength to obtain small communities and then cluster small communities to finally obtain large communities. Currently, there are many ways to measure the strength between nodes, for example in [4-8]. Atzmueller et al. [9] focused on description-oriented community detection using subgroup discovery. Moon et al. [10] developed two parallel versions of the GN algorithm to support large-scale networks. Xin et al. [11] designed the RWS (Random Walk Sampling) method to detect overlapping communities and found the closest friends for each node. Mirsaleh and Meybodi [12] proposed an evolutionary algorithm in which each chromosome represents a part

* Corresponding author.
E-mail address: tianyuling@tyut.edu.cn

of the solution and the whole population represents the solution. De et al. [13] presented a generative model and an efficient expectation-maximization algorithm. Krista and Borut [14] presented a multi-objective evolutionary algorithm and evaluated it using problem-specific genetic mutation, group crossover, and problem-specific initialization. Mensah and Soundarajan [15] presented a network sampling technique to crawl the community structure of dynamic networks. Zhou et al. [16] proposed a gain function with similarity and used the game theory to detect overlapping communities.

In order to measure the relationship strength between nodes in the network, different hierarchical partitioning algorithms seek different measurement standards; however, the traditional algorithms only add one node in each iteration, which leads to too many computation operations and reduces the partition efficiency.

In this paper, to increase the information that can be referred to in the community division of networks, three attributes that affect the strength relationship between nodes were considered. Then, the immune algorithm, with the abilities of inherent parallelism and rapid computing, was adopted to perform the traversal optimization of the whole complex network and find appropriate core nodes of communities. Finally, according to each core code, the remaining nodes with the highest affinity were found and classified into the corresponding community, and the fast community division was realized. The proposed algorithm does not need the number of communities and effectively avoids too many iterations in the process of division.

2. The Artificial Immune Algorithm

The artificial immune algorithm is an artificial intelligence method that mimics the biological immune system. In data processing, it has the ability of learning, memory, and parallel distributed processing.

Lizondo et al. [17] proposed a distributed demand-side management system in smart grids by using the artificial immune algorithm and adapted to tackle the Peak Load problem. Sharmila and Sakthi [18] enhanced the precision of the anomalous distinguishing pattern and framed an artificial immune system. In [19], immune memory enabled the design of a Case-Based Reasoning (CBR) System to provide a direct representation of knowledge about disturbances, and an immune inspired traffic signal control algorithm was proposed. In [20] a developed modified algorithm based on artificial immune system was presented, in which the Random Forest algorithm was used for data pre-processing and extraction of informative signs describing the behavior of a complex object of control.

In the artificial immune algorithm, antigens and antibodies are represented as L-dimensional vectors, corresponding to particular positions in the morphological space. Antigens and antibodies interact with each other in morphological space, and affinity reflects their interaction strength. Antibodies gain the right to survive through competition for learning. Antibodies with high affinity to antigens will be cloned by cloning selection and arouse activation and expansion, while antibodies with low affinity to antigens will be eliminated. In addition, recognition between similar antibodies leads to suppression, and similar antibodies will be eliminated. As antibodies evolve, memory cells eventually emerge. The memory cell matrix will be eventually output.

The memory cell matrix is a boundary weighted graph that is not fully connected. It consists of cell nodes, with the node pair connection as its boundary, and each connection corresponding to the assigned weight value. The example structure of the memory cells is shown in Figure 1.

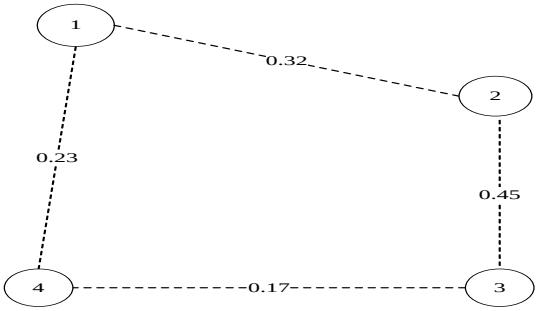

Figure 1. The example structure of the memory cells

There are four memory cells named node 1, node 2, node 3, and node 4, respectively. The affinity between memory cells is often between 0 and 1. The affinity between node 1 and node 2 is 0.32 and the affinity between node 1 and node 4 is 0.23, so node 1 is more similar to node 2.

3. The Community Division Method based on the Immune Algorithm

In this section, we describe the proposed algorithm based on the immune algorithm, the main idea of which is as follows:

(1) Divide the nodes into two parts: one part of the nodes is mapped to antigens, and the other part is mapped to antibodies. The relationship strength between two nodes in the network is mapped to the affinity of antibody to antigen. The core nodes of communities are mapped to memory cells evolved by antibodies.

(2) When antigens constantly enter, stimulate antibodies with high affinity to them. Through cloning, mutation, and competitive selection, antibodies with high affinity to antigens evolve into memory cells and are stored in memory cell banks. Dynamically update network parameters according to the evolutionary results.

(3) Search for optimal antibodies within the iteration time, and obtain the memory cell bank with excellent overall quality. Regard each memory cell in the memory cell bank as the core node of the community.

(4) Calculate the affinity of remaining nodes to each core node of the community. Select the node with highest affinity to the core node, and set them in the same community. If the affinity of the node and the core nodes are equal, the node is a community overlapping node.

3.1. Calculation of Affinity

To measure the relationship strength between nodes, that is, the affinity of antibodies to antigens, we consider the following three attributes synthetically based on the known node connections:

3.1.1. Shared Neighbor Number

In the process of community division, which community the node belongs to is often related to the belonging of the neighbor node. Considering that most nodes are not directly connected, the belonging of a node is often affected by the node that has most shared neighbors with it. In this paper, we measure the relationship strength by ratio of shared neighbor number between two nodes to the total number of their respective neighbors. The normalization is shown in Equation (1).

.

$$CN(i, j) = \frac{c(n(i), n(j))}{n(i) + n(j)} \tag{1}$$

Where $n(i)$ represents the neighbor number of node v_i and $c(n(i),n(j))$ represents the shared neighbor number between node v_i and node v_j.

3.1.2. Neighbor Similarity

Researchers found that the number of neighbor nodes is not a decisive factor in the belonging of nodes, but the number of connected subgraphs between neighbor nodes is. The higher the connection number between adjacent nodes, the closer the relationship strength between the two nodes. Therefore, we not only consider the number of shared neighbors but also introduce the neighbor similarity. The normalization is shown in Equation (2).

$$ES(i, j) = \frac{2e(n(i), n(j))}{n(i)n(j)} \tag{2}$$

Where $e(n(i), n(j))$ represents connection number between neighbors of node v_i and neighbors of node v_j.

3.1.3. Path Length

Nodes with shorter paths have a high probability of belonging to the same community. Therefore, in addition to measuring the number of shared neighbors and neighbor similarity, we measure the path length between nodes as well. In this paper, we define the path length between nodes as the number of edges contained in the shortest path connecting two nodes. The normalization is shown in Equation (3). If node v_i disconnects node v_j, set $D(v_i, v_j) = d$, where d represents the average path length of the network.

$$LS(i, j) = \frac{1}{D(i, j)+1} \tag{3}$$

Where $D(i, j)$ represents the path length between node v_i and node v_j.

In order to detect different structures of networks, in this paper, we define the affinity of node v_i to node v_j as $I(i, j)$. The normalization is shown in Equation (4).

$$I(i, j) = \frac{a \cdot CN(i, j) + b \cdot ES(i, j) + c \cdot LS(i, j)}{a + b + c} \tag{4}$$

Where a, b, and c represent weights corresponding to the above three attributes respectively. It is important to note that different combinations of weights (a, b, c) will affect the contribution of the corresponding attributes to the affinity $I(i, j)$, and then affect the division effects of the algorithm. Therefore, before realizing the community division of the network, we should analyze the influence of weights on the division effects under different values and obtain the best weight combination. In this paper, we adopt the standard modularity Q to measure the influence of weight on division effects. The greater the value of Q, the more ideal the division effects. The method of calculation is shown in Equation (5).

$$Q = \frac{1}{2m} \sum_{vw} [A_{vw} - \frac{k_v k_w}{2m}] \delta(c_v, c_w) \tag{5}$$

Where m represents the total number of edges in the network. If node v and node w are connected, $A_{vw} = 1$; otherwise, $A_{vw} = 0$. k_v represents the degree of the node v. If node v and node w are in the same community, $\delta(c_v, c_w) = 1$; otherwise, $\delta(c_v, c_w) = 0$.

3.2. Cloning and Mutation Operations

In the immune algorithm, cloning and mutation operations are used to realize fast searching and the whole optimization. The number of clones is directly proportional to the value of affinity, and the number of mutations is inversely proportional to the value of affinity. In this paper, we keep the condition that affinity degree affects cloning and mutation, and the number of clones and mutations also changes dynamically with the number of iterations of the algorithm. The equations for the number of clones and mutations are Equations (6) and (7), respectively. In order to avoid redundant optimization, we set the antibody node transformed into its neighbor node by directional mutation.

$$cnum(j) = (I(i, j) - V_i)\left(1 - \frac{It_i}{It_n} \times 0.01\right) \times 100 \tag{6}$$

$$vnum(j) = \frac{0.01}{I(i, j) - V_i}\left(1 - \frac{It_i}{It_n} \times 0.01\right) \tag{7}$$

Where $I(i, j)$ represents the affinity value of the node v_j and the node v_i and V_i represents the affinity threshold. It_i denotes the i^{th} iteration, and It_n denotes the total number of iterations.

3.3. Community Division Process

The implementation process of the proposed algorithm is as follows (suppose there are N nodes to be partitioned):

Step 1 Number the nodes to be partitioned, input the connection relationships of nodes, and select R nodes randomly as initial antibodies. Set $R = \sqrt{N}$, because the number of communities is generally not more than $\sqrt{N}$.

Step 2 Set the remaining N-R nodes as antigens, calculate the affinity of each antigen to antibodies according to Formula 4, and set the affinity as the stimulation value of the corresponding antibody.

- Immune cloning and mutation. For the i^{th} antigen Ag_i, according to Equation (7), clone the antibody whose affinity value is higher than the threshold V_i, and these antibodies conduct the set Ab_i. Mutate the remaining antibodies whose affinity values are lower than V_i, and these antibodies conduct the set Ab_i'.
- Recalculate the affinity of antibodies in Ab_i' to the antigen Ag_i, and the antibody whose affinity value is higher than V_i and antibodies in Ab_i are mutually inhibited, evolve into memory cells, and are then stored in memory cell set M_i. If the stimulation value of the memory cells in M_i is higher than V_i, update the best affinity as V_i.
- After all antigens enter, all the memory cell sets conduct the memory cell bank M. Count the cumulative stimulation of memory cells in M as *SUM*. The memory cells in M differentiate into the next iteration of the initial antibodies. In addition, add new antibodies randomly.

Step 3 Iterate step 2 until the *SUM* is no longer added or the number of iterations is reached. Obtain the best memory cell bank.

Step 4 Set memory cells in the memory cell bank as the core nodes of the communities, and calculate the affinity of the remaining nodes to each core node of communities. Select the node with the highest affinity to the core node, and set them in the same community. If the affinity of the node and the core nodes are equal, the node is a community overlapping node. The community structure division of the network is finished, and the number of core nodes is the number of communities of the network.

4. Experimental Results and Analysis

4.1. Experimental Datasets

In order to verify the rationality and validity of the proposed algorithm in this paper, we introduce the following three real datasets for experiments:

(1) Zachary Network: it is derived from the observation of friendships among members of an American university karate club and takes the members of the club as nodes, the friendship relationships between the members as edges, and consists of 34 nodes and 78 edges.

(2) Dolphin Network: it is derived from the observation of 62 dolphin populations in the Doubtful Sound Strait of New Zealand and takes the members of dolphins as nodes, frequent contacts between dolphins as edges, and consists of 62 nodes and 159 edges.

(3) College Football Network: it is derived from the American College Football League, takes the soccer teams as nodes, the games between teams as edges, and consists of 115 nodes and 613 edges.

4.2. Experimental Results

We mainly compare the proposed algorithm with the following two classical community division algorithms:

(1) Fast Newman: it was proposed by Newman based on the idea of greedy algorithm. By calculating the similarity of each pair of nodes, the nodes with high similarity are clustered to form the community structure. The time complexity of this algorithm is reduced while the division accuracy is also reduced.

(2) GN: it was proposed by Newman and Girvan based on edge mediums. By deleting the edges in the network, the community structure is detected. Both the division accuracy and the time complexity of this algorithm are high.

4.2.1. Community Division of Zachary Network

4.2.1.1. Parameters Selection

Set $N = 34$, $R = 6$, and the initial value of V as 0.05. Analyze the influence of the weight value of attributes on the modularity Q (shown in Figure 2) and obtain the optimal weight combination, where $a = 0.40$, $b = 0.80$, and $c = 0.90$.

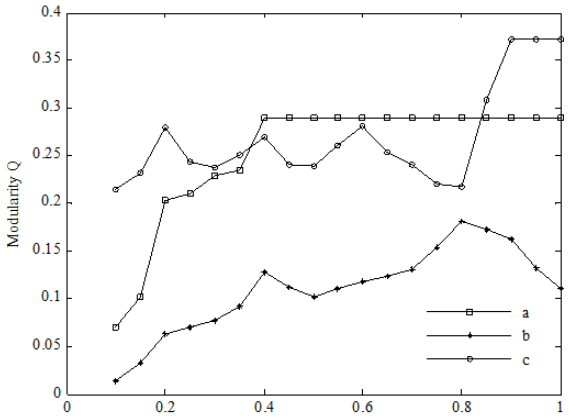

Figure 2. The change of modularity *Q* under different values of *a*, *b*, and *c*

4.2.1.2. Experimental Results of the Proposed Algorithm

Through the proposed algorithm, we obtain node 1 and node 34 as the community core nodes, classify remaining nodes, and obtain the community structure shown in Table 1.

Table 1. Community structure of Zachary

Community number	Community members number
Community 1	1,2,3,4,5,6,7,8,11,12,13,14,17,18,20,22
Community 2	9,10,15,16,19,21,23,24,25,26,27,28,29,30,31,32,33,34

4.2.1.3. Comparison with Classical Algorithms

The community structure detected by the proposed algorithm is consistent with the actual structure of the Zachary network. However, both the fast Newman algorithm and the GN algorithm fail to detect one node. Compared with the fast Newman algorithm and GN algorithm, the proposed algorithm increases the division accuracy rate by 3%.

4.2.2. Community Division of Dolphin Network

4.2.2.1 Parameters Selection

Set $N = 62$, $R = 8$, and the initial value of V as 0.05. Analyze the influence of the weight value of attributes on the modularity Q (shown in Figure 3) and obtain the optimal weight combination, where $a = 0.50$, $b = 0.90$, and $c = 0.60$.

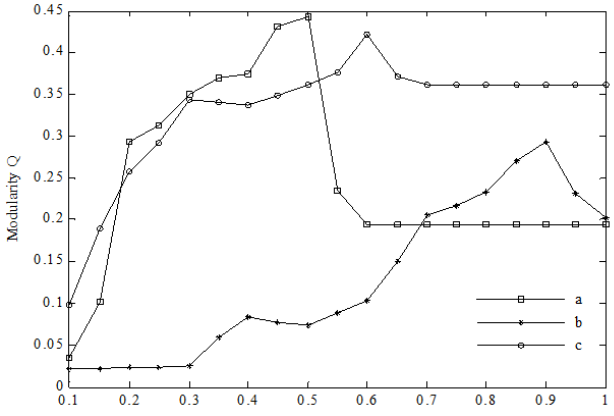

Figure 3. The change of modularity *Q* under different values of *a*, *b*, and *c*

4.2.2.2. Experimental Results of the Proposed Algorithm

Through the proposed algorithm, we obtain node 15 and node 58 as the community core nodes, classify remaining nodes, and obtain the community structure shown in Table 2.

Table 2. Community structure of Dolphin

Community number	Community members number
Community 1	2,6,7,8,10,14,18,20,23,26,27,28,32,33,40,42,49,55,57,58,61
Community 2	1,3,4,5,9,11,12,13,15,16,17,19,21,22,24,25,29,30,31,34,35,36,37, 38,39,41,43,44,45,46,47,48,50,51,52,53,54,56,59,60,62

4.2.2.3. Comparison with Classical Algorithms

The community structure detected by the proposed algorithm is consistent with the actual community structure of the Dolphin network. However, both the fast Newman algorithm and the GN algorithm fail to detect three nodes and one node, respectively. Compared with the fast Newman algorithm and GN algorithm, the proposed algorithm increases the division accuracy rate by 5% and 2%, respectively.

4.2.3. Community Division of College Football Network

4.2.3.1. Parameters Selection

Set N = 115, R = 11, and the initial value of V as 0.05. Analyze the influence of the weight value of attributes on the modularity Q (shown in Figure 4) and obtain the optimal weight combination, where a = 0.5, b = 0.9, and c = 0.2.

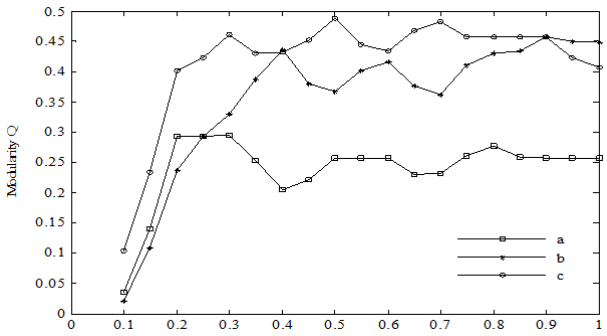

Figure 4. The change of modularity Q under different values of a, b, and c

4.2.3.2. Experimental Results of the Proposed Algorithm

Through the proposed algorithm, we obtain the community core nodes as nodes 2, 6, 4, 8, 9, 16, 62, 70, 83, 88, and 89, classify remaining nodes, and obtain the community structure shown in Table 3.

Table 3. Community structure of College Football

Community number	Community members number
Community 1	2,26,34,38,46,64,90,104,106,110
Community 2	6,4,11,41,53,73,75,82,85,98,99,103,108
Community 3	8,9,22,23,52,69,78,79,109,112
Community 4	16,3,7,14,33,40,48,61,65,101,107
Community 5	62,13,15,19,27,32,35,39,43,44,55,72,86,100
Community 6	70,12,25,29,51,91,
Community 7	83,81
Community 8	88, 21,28,57,60,63,66,71,77,96,97,114
Community 9	89, 47,50,54,59,68,74,84,111,115
Community 10	93,45,49,58,67,76,87,92,113

4.2.3.3. Comparison with Classical Algorithms

According to the actual community structures of the College Football network, the proposed algorithm fails to detect seven nodes. However, the fast Newman algorithm and the GN algorithm fail to detect 12 nodes and 10 nodes, respectively. Compared with the fast Newman algorithm and GN algorithm, the proposed algorithm increases the division accuracy rate by 5% and 3%, respectively.

Through the above experiments, the comparison of the proposed algorithm with the fast Newman algorithm and GN algorithm in division accuracy and execution time are shown in Figures 5 and 6, respectively. The experimental results show that the proposed algorithm can achieve more accurate division results and improve community structure of different

networks. However, the operation time of the algorithm is higher than that of the fast Newman algorithm and lower than that of the GN algorithm. Therefore, further improvements should continue to be made to the proposed algorithm.

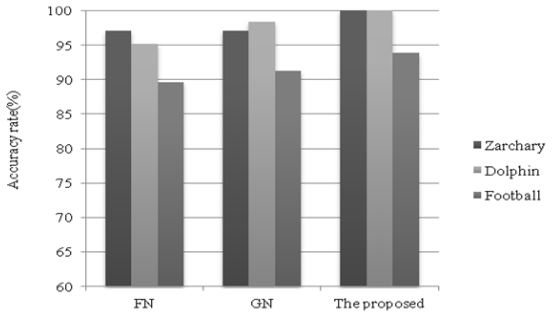

Figure 5. Comparison of division accuracy rates of different algorithms on three datasets

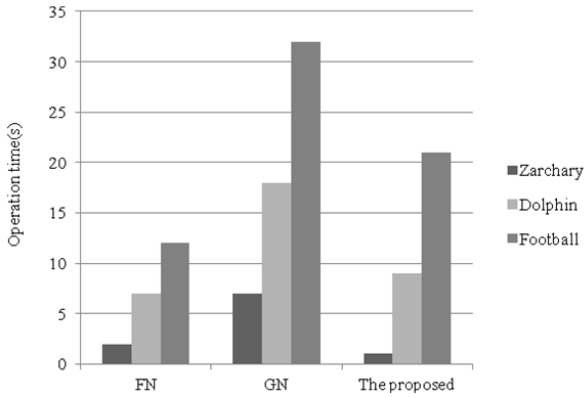

Figure 6. Comparison of operation time of different algorithms on three datasets

5. Conclusions

Aiming to improve the division accuracy and reduce operation time, in this paper, we proposed a new division algorithm. Based on the immune algorithm, the proposed algorithm measures the node relationship comprehensively and realizes the fast division of community structure.

The experimental results showed a relative improvement in comparison with other community division algorithms. In the future, we will further consider the factors affecting the accuracy of community division and improve the accuracy and performance of the algorithm on the directed networks.

Acknowledgments

This work is supported by the National Natural Science Foundation Project (No. 61472271).

References

1. X. Lin, M. Wang, and Z. Tian, “Influence of Crops and Hedges on Soil Microbial Community Structure and Diversity of the Sloping Agricultural Land,” *Safety and Environmental Engineering*, Vol. 2, pp. 244-249, February 2017
2. L. Y. Zhang and P. Marbach, “Stable and Efficient Structures for the Content Production and Consumption in Information Communities,” *Springer*, pp. 163-173, August 2018
3. Z. Su, W. Wang, and L. X. Li, “Optimal Community Structure for Social Contagions,” *New Journal of Physics*, Vol. 20, No. 5, pp. 1-10, May 2018
4. P. Tasgave and A. Dani, “Friend-Space: Cluster-Based Users Similar Post Friend Recommendation Technique in Social Networks,” in *Proceedings of the* 2015 *International Conference on Information Processing*, pp. 658-663, Pune, India, June 2016
5. S. Jarukasemratana, T. Murata, and X. Liu, “Community Detection Algorithm based on Centrality and Node Distance in Scale-Free Networks,” in *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pp. 258-262, Paris, France, May 2013
6. A. Anand, V. K. Sihag, and P. Svss, “Community Structure based on Node Traffic in Networks,” *International Journal of*

Computer Applications, Vol. 69, No. 13, pp. 15-20, May 2013
7. B. Cecile, C. David, M. Matteo, and M. Barbora, "Clustering Attributed Graphs: Models, Measures and Methods," *Network Science*, Vol. 3, No. 3, pp. 408-444, January 2015
8. D. Hric, T. Peixoto, and S. Fortunato, "Network Structure, Metadata and the Prediction of Missing Nodes and Annotations," ar Xiv: 1604.00255v1, September 2016
9. M. Atzmueller, S. Doerfel, and F. Mitzlaff, "Description-Oriented Community Detection using Exhaustive Subgroup Discovery," *Information Sciences*, Vol. 329, pp. 965-984, February 2016
10. S. Moon, J. G. Lee, and M. Kang, "Parallel Community Detection on Large Graphs with MapReduce and GraphChi," *Data and Knowledge Engineering*, Vol. 104, pp. 17-31, July 2016
11. Y. Xin, Z. Q. Xie, and J. Yang, "An Adaptive Random Walk Sampling Method on Dynamic Community Detection," *Expert Systems with Applications*, Vol. 58, pp. 10-19, October 2016
12. M. R. Mirsaleh and M. R. Meybodi, "A Michigan Memetic Algorithm for Solving the Community Detection Problem in Complex Network," *Neurocomputing*, Vol. 214, pp. 535-545, November 2016
13. C. De Bacco, E. A. Power, D. B. Larremore, and C. Moore, "Community Detection, Link Prediction, and Layer Interdependence in Multilayer Networks," *Physical Review E*, Vol. 95, pp. 042317, January 2017
14. R. Z. Krista and Z. Borut, "Multi-Objective Evolutionary Algorithm using Problem-Specific Genetic Operators for Community Detection in Networks," *Neural Computing and Applications*, Vol. 30, pp. 1-14, February 2017
15. H. Mensah and S. Soundarajan, "Sampling Community Structure in Dynamic Social Networks," *Springer*, pp. 114-126, May 2018
16. X. Zhou, X. H. Zhao, Y. H. Liu, and G. Sun, "A Game Theoretic Algorithm to Detect Overlapping Community Structure in Networks," *Physics Letters A*, Vol. 382, No. 13, pp. 872-879, April 2018
17. D. Lizondo, S. Rodriguez, and A. Will, "An Artificial Immune Network for Distributed Demand-Side Management in Smart Grids," *Information Sciences*, Vol. 438, pp. 32-45, April 2018
18. L. Sharmila and U. Sakthi, "An Artificial Immune System-based Algorithm for Abnormal Pattern in Medical Domain," *Journal of Supercomputing*, Vol. 4, pp. 1-15, April 2018
19. A. Louati, S. Darmoul, and S. Elkosantini, "An Artificial Immune Network to Control Interrupted Flow at a Signalized Intersection," *Information Sciences*, Vol. 433-434, pp. 70-95, April 2018
20. G. Samigulina and Z. I. Samigulina, "Modified Immune Network Algorithm based on The Random Forest Approach for The Complex Objects Control," *Artificial Intelligence Review*, Vol. 1, pp. 1-17, February 2018

Yuling Tian is a professor and master's tutor at Taiyuan University of Technology. Her research focuses on artificial intelligence and fault diagnosis.