Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 7498605, 8 pages
https://doi.org/10.1155/2020/7498605

Research Article

Complex Network Filtering and Compression Algorithm Based on Triangle-Subgraph

Shuxia Ren, Tao Wu, and Shubo Zhang

Department of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin 300387, China

Correspondence should be addressed to Shuxia Ren; t_rsx@126.com

Received 21 March 2020; Accepted 9 May 2020; Published 1 June 2020

Guest Editor: Hao Shen

Compressing the data of a complex network is important for visualization. Based on the triangle-subgraph structure in complex networks, complex network filtering compression algorithm based on the triangle-subgraph is proposed. The algorithm starts from the edge, lists nodes of the edge and their common node sets to form a triangle-subgraph set, parses the triangle-subgraph set, and constructs new complex network to complete compression. Before calculating the set of triangle-subgraph, node importance ranking algorithm is proposed to extract high- and low-importance nodes and filter them to reduce computational scale of complex networks. Experimental results show that filtering compression algorithm can not only improve the compression rate but also retain information of the original network at the same time; sorting result analysis and SIR model analysis show that the sorting result of node importance sorting algorithm has accuracy and rationality.

1. Introduction

Some networks contain millions or even billions of nodes and edges bringing new challenges to understand and analyse complex networks. Without compression, points and edges presented will be very dense, and it is difficult for people to obtain useful information from the network.

Scholars began to pay attention to the compression field of complex networks [1] and designed many methods to compress complex networks. Gilbert [2] proposed an algorithm for node compression based on the node importance evaluation index, which achieves the purpose of compressing the network by deleting noncritical nodes and edges. The disadvantage is the use of keep-one and keep-all strategies to supplement important nodes and edges, which costs much time. Yan and Zhang[3] and others are inspired by centrality that can stress important nodes. Five compression schemes are proposed from five aspects: random centrality, degree centrality, relative node importance, PageRank, and intermediate centrality. The disadvantage is that node and edge are deleted and new network is not considered to supplement, resulting in the loss of network information. Zhang et al. [4] proposed a bound_tri algorithm based on the triangle structure. The algorithm starts from nodes and compresses the network by constructing triangle set. The disadvantage is that the need to access the adjacency matrix and adjacency list at same time leads to high-time complexity, and bound_tri algorithm takes degree as the selection criterion, which does not conform to the actual situation of the network.

The abovementioned algorithm mainly starts from node and uses ways of deleting and merging nodes to realize the compression of a complex network. Based on edge, the paper proposes complex network filtering and compression algorithm based on the triangle-subgraph. In order to shorten compression time and improve compression efficiency, before calculating the triangle-subgraph set, the paper proposes node influence as a global importance measure and proposes node importance sorting algorithm NRSA (Node Rank Select) based on LeaderRank algorithm [5] to extract high- and low-importance nodes and filter them to reduce size of complex networks that need to be calculated. The experimental results show that important node ranking results of NRSA algorithm are better than other sorting algorithms, and compression algorithm improves compression efficiency, and retains most of the information of the original network.

2. NRSA Algorithm

In LeaderRank algorithm, the probability of jumping to an adjacent node is calculated by the degree of each node. The way of calculation only considers partial importance of the node, ignoring global importance in the entire network.

It can be seen from Figure 1 that node 4 has more neighboring nodes than node 7, and the LR value assigned to node 4 with each of the iterations by LeaderRank algorithm is more than node 7. Therefore, the result of the LeaderRank algorithm is that node 4 is more important than node 7. However, from the structure of the network, node 7 acts as an intermediate node connecting three different node groups, which should be the most important in the entire network structure. In order to comprehensively consider partial importance and global importance of nodes, adjacent-degree is used to construct influence of nodes as the global importance.

Definition 1. Adjacent-degree: v_i is the node in the network G; adjacent-degree is the sum of the degrees of all adjacent nodes, recorded as AdjDe.

$$\mathrm{AdjDe}(v_i) = \sum \mathrm{De}(v_j). \tag{1}$$

Definition 2. Node influence: the node influence of v_i is sum of the adjacent-degree of v_i and degree of v_i. It is recorded as Node_In (v_i).

$$\mathrm{Node\_In}(v_i) = \mathrm{AdjDe}(v_i) + \mathrm{De}(v_i). \tag{2}$$

Definition 3. Node rank: the formula for calculating the NR_i of node v_i is as follows:

$$\mathrm{NR}_i = \mathrm{LR}_i \sum_{j\in a_i} \frac{\mathrm{Node\_In}(v_j)}{\mathrm{Node\_In}(v_i)} \mathrm{LR}_j. \tag{3}$$

Equation 3 shows (1) a_i is the adjacent node connected to node v_i. The more the number of adjacent nodes, the higher the NR value of node v_i, which satisfies intuitive judgment. (2) The greater the node influence of node v_j, the more the LR value of v_j obtained by v_i.v_i is more important in the network structure.

The steps of NRSA algorithm are as follows:

(1) The LR value of the complex network's node is calculated using LeaderRank algorithm

(2) The adjacent-degree of node v_i is calculated using equations (1) and (2), and the node influence Node_Ef (v_i) is obtained

(3) Using equation (3), the LR_i value and Node_Ef (v_i) are calculated to obtain NR_i, and the result is mapped to the [0, 1] interval using deviation standardization

3. Complex Network Filtering and Compression Algorithm Based on Triangle-Subgraph

3.1. Triangle-Subgraph. Triangle-subgraph [6] is a special 3-connected subgraph in a graph data-connected subgraph. In a complex network $\mathrm{G} = (\mathrm{V}, E)$, $T_\triangle = (v_\triangle, e_\triangle)$ can represent a triangle-subgraph with three nodes and three edges:

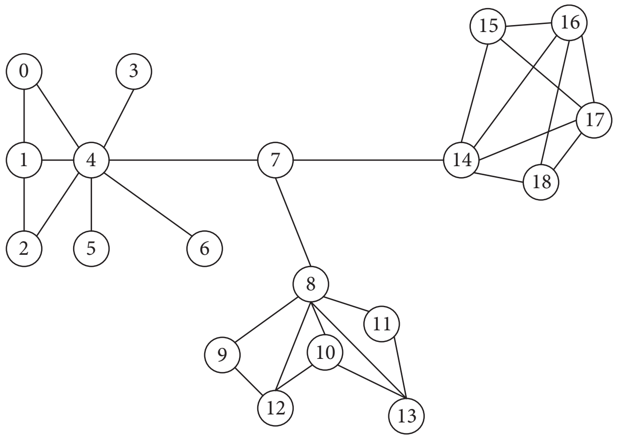


Figure 1: Simple undirected network.

$$\begin{aligned} T_\triangle &= (v_\triangle, e_\triangle) = T_\triangle < a, b, c \geq \{v_\triangle = \{a, b, c\} \subset v, \\ e_\triangle &= \{(a, b), (a, c), (b, c) \subset e\}\}. \end{aligned} \tag{4}$$

For G with e edges, time required to calculate the triangle-subgraph is $o(e^{3/2})$ [7], indicating that the triangle-subgraph can be calculated within certain time.

3.2. List Triangle-Subgraph. The traditional method of listing a triangle-subgraph is node compression algorithm [8]. The algorithm proposed in paper is an edge compression algorithm. First, an edge is randomly selected from the complex network, and then the adjacency list of two nodes connected to the edge is searched to check whether they have a common adjacent node, and finally they are combined with a common adjacent node to form a triangle-subgraph set. For example, the selected edge is (a, b), and the adjacency list of nodes connected to the edge is $\mathrm{Adj}(a) = \{w, h, m, n\}$ and $\mathrm{Adj}(b) = \{w, h, m, n, l, d\}$. The common adjacent nodes of node a and b are $\mathrm{Adj}(a) \cap \mathrm{Adj}(b) = \{w, h, m, n\}$. The set of all triangle-subgraphs for edges (a, b) is $<\mathrm{a, w, b}>$, $< a, h, b >$, $< a, m, b >$, $< a, n, b >$.

3.3. Triangle-Subgraph Compression Algorithm with Filtering Property. From the complex network real dataset Polblogs and Youtube's node importance value statistics chart, Figure 2 shows that two graphs have an important feature, that is, the value distribution of node importance in complex network is very uneven. Moreover, there are very few common adjacent nodes that can be contained between high-importance and low-importance nodes. But, looking for a triangle-subgraph between high-importance and low-importance nodes consumes much computing resources. Therefore, before calculating the triangle-subgraph, the method of filtering out high- and low-importance nodes is used to reduce the data size in the complex network.

We propose and define the triangle-subgraph compression algorithm with filtering properties as the NIIET (node importance in edge triangle) algorithm. NIIET

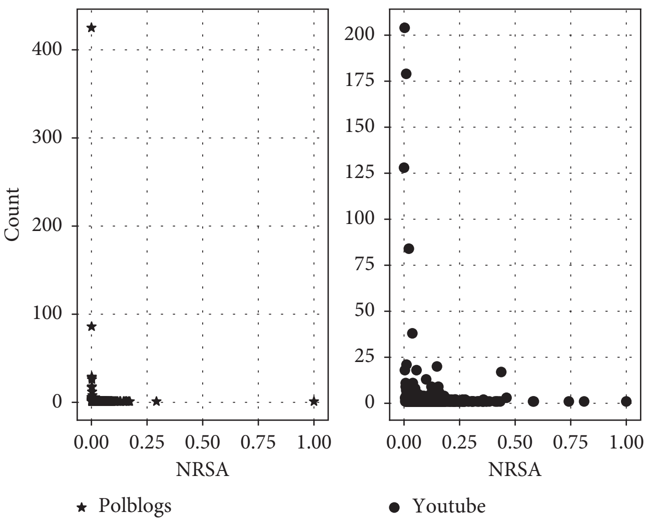


Figure 2: Node importance statistics of real dataset.

algorithm only needs to access the adjacency list of the complex network when compressing complex networks. The adjacency list contains the directionality of edge, so it can be applied to compression of a directed graph and undirected graph.

Figure 3 is a simple undirected network diagram containing 16 edges and 8 nodes. The results obtained by the edge compression algorithm are shown in Table 1. It can get a triangle-subgraph set with 42 triangles, which will result in 27 edge redundancy. If storing edge of triangle-subgraph requires 2 units, then 54 units of redundancy will store. Therefore, by filtering out high- and low-importance nodes, redundancy of the triangle subgraph set and compression efficiency are reduced.

First, the importance of the nodes calculated by the NRSA algorithm is shown in Table 2. Next, set the low importance node standard low_percent = 15% and the high importance node standard high_percent = 85%. The low importance node is node 7, and the high importance node is node 4. Finally, when listing a triangle-subgraph, the information containing node 7 and node 4 is filtered from the adjacency list of other nodes. The network after filtering the high- and low-importance nodes is shown in Figure 4. The calculation results are shown in Table 3. There are only 15 triangle-subgraphs, and the triangle subgraph set contains only 7 redundant edges. The results show that a complex network with a high compression rate can be obtained by filtering the low- and high-importance nodes.

3.4. The Steps of NIIET Algorithm

Input: Complex network $G(V, E)$, Adjacency list AdjG.

Output: Complex network $G'(V', E')$, triangle-subgraph set *trangle_list*.

(1) Input complex network $G(V, E)$, and calculate value of node by using NRSA algorithm;

(2) Set percentage of low_percent and high_percent;

(3) According to percentage, the low importance node set low_nodelist and the high importance node set high_nodelist are obtained;

(4) Traversing the edge E. If node v_i and v_j at both ends of the edge are located in low_nodelist or high_nodelist, the information of two nodes is filtered out from the adjacency list of v_i and v_j connected nodes;

(5) Traversing edge E. If AdjG(v_i) and AdjG(v_i) intersect, triangle-subgraph set trangle_list consists of nodes v_i and v_j and their common node set;

(6) Parse the triangle-subgraph set trangle_list, construct new complex network $G'(V', E')$;

(7) Output $G'(V', E')$, triangle-subgraph set trangle_list.

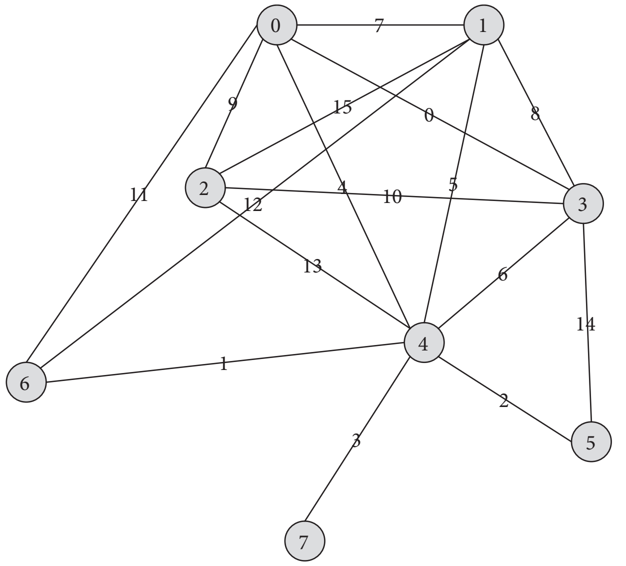


Figure 3: The original network.

Table 1: Number of triangle-subgraph sets in the original network.

E	Num	E	Num	E	Num	E	Num
0	3	4	4	8	3	12	2
1	2	5	4	9	3	13	3
2	1	6	4	10	3	14	1
3	0	7	4	11	2	15	3

Table 2: Node importance table of the original network.

Node	NRSA	Node	NRSA
4	1.0	2	0.63003303870
0	0.77232674038	6	0.42442432758
1	0.77232674038	5	0.19582840067
3	0.76363255124	7	0.0

4. Experimental Analysis

The paper analyses the node importance experiment and the compression experiment, respectively, and selects Zachary

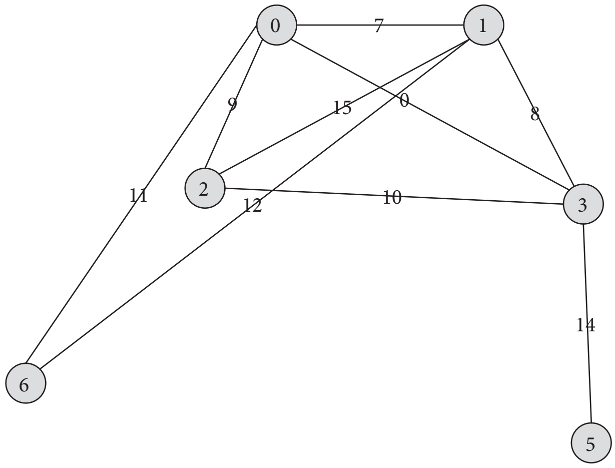


Figure 4: Filter network after high- and low-importance nodes.

Table 3: Number of triangle-subgraph sets of the complex network after filtering.

E	Num	E	Num	E	Num	E	Num
0	2	4	0	8	2	12	1
1	0	5	0	9	2	13	0
2	0	6	0	10	2	14	0
3	0	7	3	11	1	15	2

[9], Football [10], Neural [10], Netscience [11], Polblogs [12], and Youtube [13] to conduct experiments.

4.1. Node Importance Analysis. To further prove the rationality of the NRSA algorithm, the SIR model [14] is used to propagate the node importance ranking results of PageRank, LeaderRank, and NRSA algorithms in the neural network. The change of lin was mainly observed as the ratio of the number of nodes in the I(infected) state in the SIR model to the total number of nodes in the network. Top 10% nodes and top 20% of NRSA algorithm, LeaderRank, and PageRank algorithm are selected as infected nodes for propagation. The results of infection experiments of the neural network are shown in Figure 5.

As can be seen from Figures 5(a)~5(b), the highest value of lin of NRSA algorithm exceeds the LeaderRank algorithm, and both are close to 0.8. It is indicated that the node selected by the NRSA algorithm propagates in the same time step to higher depth than LeaderRank. From the time step mentioned above, the lin value of the two algorithms does not differ much between the 5 and 40 time steps. However, it can be seen from the 5–10 period that the slope of the NRSA algorithm is higher than that of the latter, indicating that the node selected by the NRSA algorithm has faster propagation speed than the LeaderRank algorithm. In general, the nodes selected by the NRSA algorithm are more reasonable than those selected by LeaderRank. Similarly, from the comparison of NRSA and PageRank algorithms in Figures 5(c)~5(d), the NRSA algorithm is better than the PageRank algorithm.

4.2. Compression Experiment Analysis. The compression experiment analysis is mainly carried out from the two parts of node selection and compression. Since NIIET algorithm uses filtering to reduce the computational scale, the selection of nodes with low and high importance will affect compression results of complex networks. Therefore, it is necessary to analyse the influence of high- and low-importance node selection criteria on compression results. At the same time, NIIET algorithm and bound_tri [4], edge_iterator_hash [15], and node_iterator [8] are used to analyse compression efficiency from the aspects of compression rate [16], compression time, and rate of information retention [17].

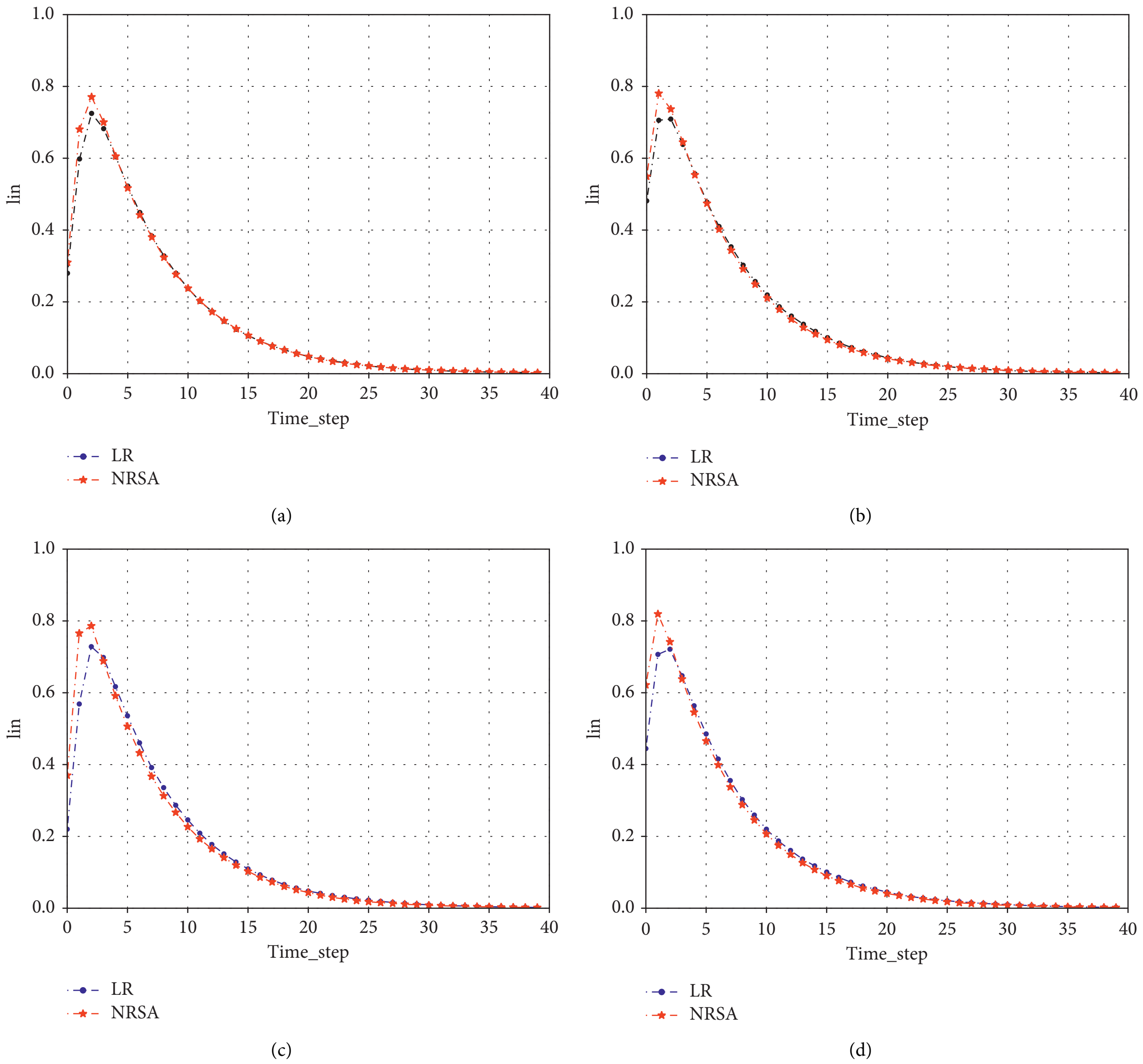
4.2.1. Analysis of Influence of Node Selection Criteria on Compression Results

(1) Low-Importance Node Selection Criteria Impact. Equation (5) is used in experiment to represent the node compression ratio. The specific formula is as follows:

$$\text{Rate\_}V = 1 - \frac{|v'|}{|v|}. \tag{5}$$

In the abovementioned formula, $|v|$ and $|v'|$ indicate the number of nodes before and after compression.

In this experiment, the range of low importance node selection criterion is set to low_percent = [10%, 30%], and the relationship between the point compression ratio and low-importance node selection criterion is shown in Figure 6. It can be seen that the influence of the criterion on



(a)

(b)

(c)

(d)

FIGURE 5: Comparison of neural network propagation results. (a) Top 10% comparison, (b) top 20% comparison, (c) top 10% comparison, and (d) top 20% comparison.

the point set compression rate is very small because the node of the low importance node's connection relationship with other nodes is small and cannot participate in the calculation of the triangle-subgraph. Therefore, the influence of low-importance node selection criteria on the compression ratio of complex networks can be neglected.

(2) High-Importance Node Selection Criteria Impact. As shown in Figure 7(b), the number of triangle-subgraph calculated by a complex network is also increased. However, the more the nodes and edges that need to be calculated for too many triangle-subgraphs, the longer the compression time will be. We will consider setting the filtering criterion to improve the compression efficiency and time of complex networks. Figures 7(a) and 7(b) show that the range of filter criteria is not very large when the range of the filter criteria is [75%, 85%]. At the same time, the range corresponding to Figure 7(a) is [1, 2], the compression ratios in the range are not much different, and the compression ratio is relatively stable, so that the balanced set of triangle-subgraph can be obtained.

4.2.2. Compression Efficiency Comparison Result Analysis. For the quantitative estimation of the compressed complex network, the paper uses the retention rate of the network information as the evaluation standard. At the same time, the high-importance node selection criterion of the experiment is set to high_percent = 85% and low-importance node selection criterion is low_percent = 10%. The compression effect comparison chart is shown in Figure 8.

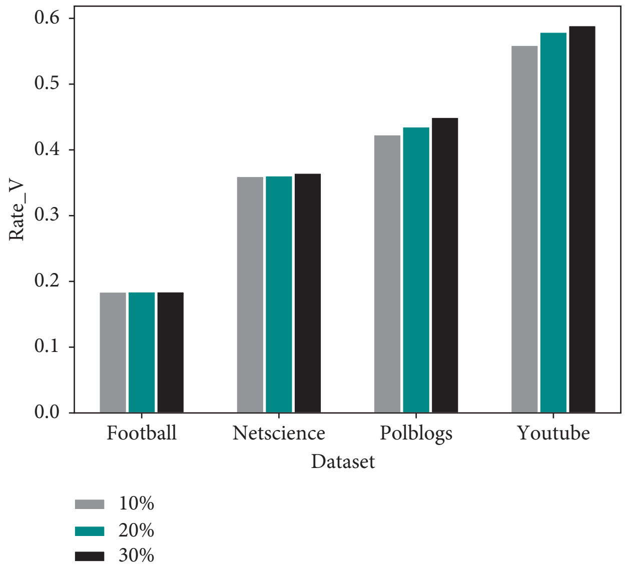
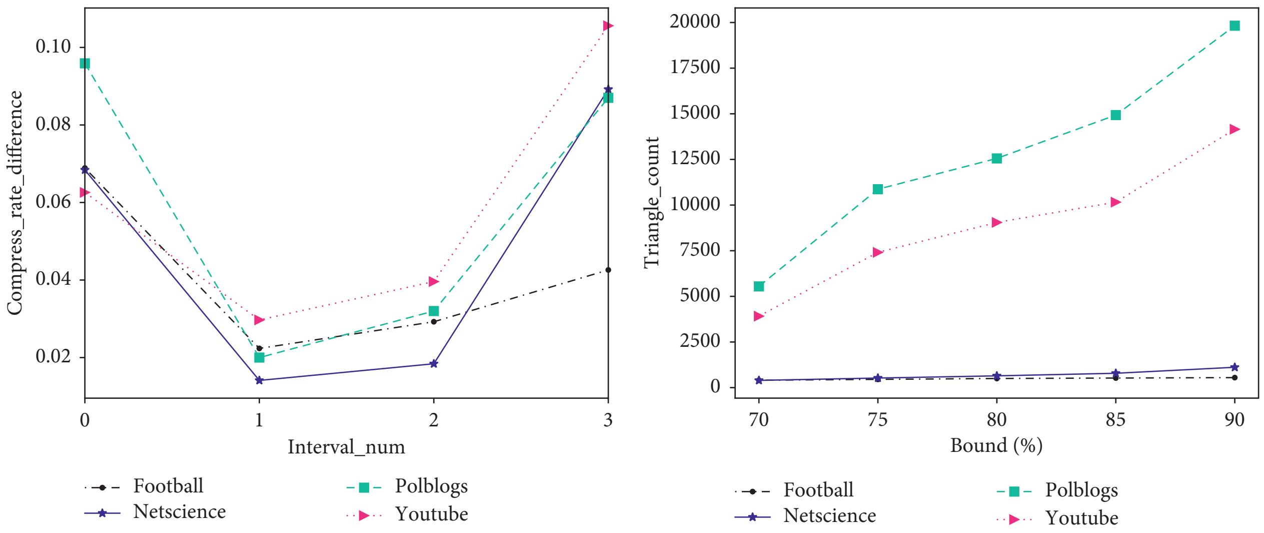


Figure 6: Different dataset points set compression ratio.



(a)

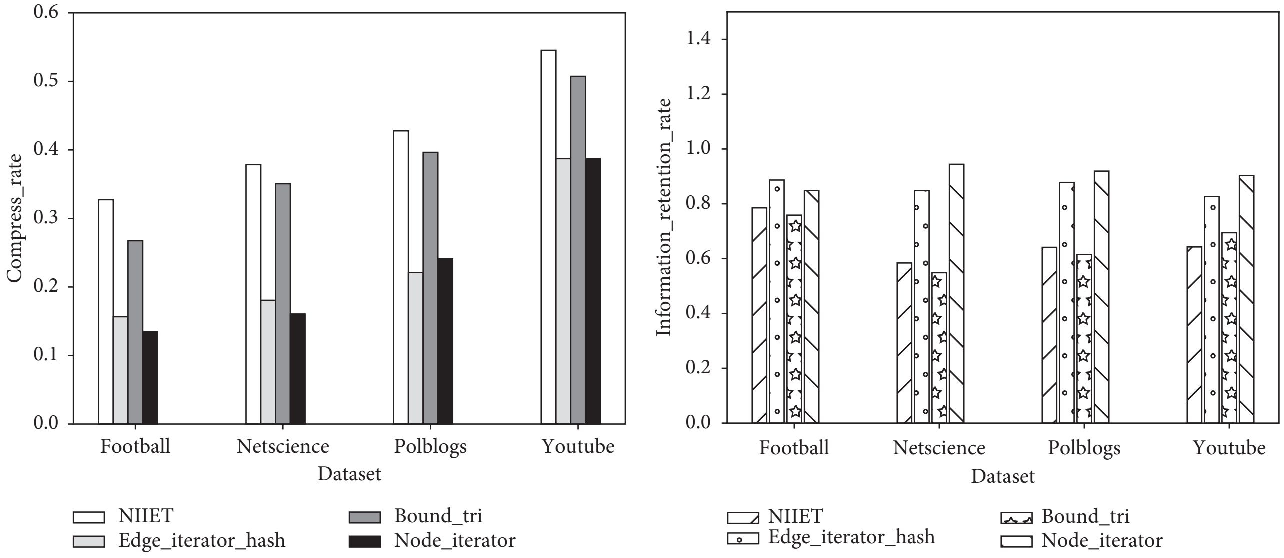
(b)

Figure 7: High-importance node selection criteria impact. (a) The difference of compression rate. (b) Number of triangle-subgraphs.

By observing Figure 8(a), from the perspective of compression ratio, as the two compression algorithms node_iterator and edge_iterator_hash require all nodes and edges in a complex network to participate in the calculation of the triangle-subgraph, compared with the first two compression algorithms, they can obtain more triangle-subgraphs, but the compression ratio is the lowest.

At the same time, NIIET algorithm is superior to the bound_tri algorithm in the compression ratio. Since bound_tri algorithm needs to access adjacency matrix of the complex network before compression to determine the edge relationship between nodes and then access and modify the adjacency list, the compression method greatly increases time complexity. Moreover, the adjacency matrix repeatedly confirms the two connected nodes, resulting in a repeated triangle-subgraph in the calculated triangle-subgraph set, which reduces the compression ratio of bound_tri algorithm.

Finally, it can be seen from Figure 8(b) that NIIET algorithm filters high- and low-importance nodes to reduce compression time and increases the compression rate, but the compressed network can still maintain 50%–70% of the network information. The quantity indicates that NIIET algorithm maintains a good effect on the amount of network information, and the compression result is reasonable and credible.



(a)

(b)

Figure 8: Comparison of the compression effect. (a) Compression rate. (b) Rate of information retention.

5. Conclusion

The paper proposes NIIET algorithm, which uses edges as iteration objects to compress complex networks by listing the triangle-subgraph. Before calculating the triangle-subgraph set, the paper proposes NRSA algorithm to calculate high- and low-importance nodes. The size of the adjacency list is reduced by filtering high-importance and low-importance nodes. The experimental results show that NRSA algorithm is reasonable. NIIET algorithm outperforms other algorithms in terms of compression rate, and the compressed network can still maintain a high amount of information, which is enough to show that compressed network retains most of structure of complex network.

In next stage of work, the paper will proceed from two aspects: (1) from the perspective of the location of the node, the influence of the node on the network function [18] to explore a new ordering method of node importance. (2) Based on the triangle-subgraph structure and other properties of complex networks, new complex network compression algorithm will proposed.

Data Availability

Data will be provided by the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Yu Zhang, Y.-B. Liu, G. Xiong et al., “Survey on succinct representation of graph data,” *Journal of Software*, vol. 9, pp. 1937–1952, 2014.

[2] A. C. Gilbert and K. Levchenko, “Compressing network graphs,” in *Proceedings of the LinkKDD Workshop at the 10th ACM Conference on KDD*, Suzhou, China, 2004.

[3] J. Yan and Z. Zhang, “Compressing big graph data: a relative node importance approach,” in *Proceedings of the 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, Nanjing, China, October 2017.

[4] L. Zhang, C. Xu, W. Qian, and A. Zhou, “Common neighbor query-friendly triangulation-based large-scale graph compression,” in *Web Information Systems Engineering—WISE 2014*, pp. 234–243, Springer, Berlin, Germany, 2014.

[5] L. Wang and R. Rao, “Multi-label propagation algorithm for overlapping community detection based on LeaderRank and node similarity,” *Computer Systems Applications*, vol. 27, no. 6, pp. 146–150, 2018.

[6] Z.-M. Han, C. Yan, Li Meng-Qi, L. Wen, and W.-J. Yang, “An efficient node influence metric based on triangle in complex networks,” *Acta Physica Sinica*, vol. 65, no. 16, 2016.

[7] M. Latapy, “Main-memory triangle computations for very large (sparse (power-law)) graphs,” *Theoretical Computer Science*, vol. 407, no. 1-3, pp. 458–473, 2008.

[8] A. Itai and M. Rodeh, “Finding a minimum circuit in a graph,” *SIAM Journal on Computing*, vol. 7, no. 4, pp. 413–423, 1978.

[9] Z. Zhu, S. Lin, and K. Cui, “Network topology layout algorithm based on community detection of complex networks,” *Journal of Computer-Aided Design & Computer Graphics*, vol. 23, no. 11, pp. 1808–1815, 2011.

[10] X. Zhou, L. Xiao, and H. Tinglei, “Time-varying complex network layout algorithm based on node centrality,” *Systems Engineering and Electronics*, vol. 39, no. 10, pp. 2346–2352, 2017.

[11] X. Zhu, X. Zhao, and M. Liu, “Analysis of influential node based on community structure,” *Application Research of Computers*, vol. 34, no. 9, pp. 2582–2585, 2017.

[12] A. Lada and G. Natalie, “The political blogosphere and the 2004 US Election,” in *Proceedings of the 3rd International Workshop on Link Discovery*, pp. 36–43, New York, NY, USA, 2005.

[13] L. Tang and H. Liu, "Uncovering cross-dimension group structures in multi-dimensional networks," in *SDM Workshop on Analysis Of Dynamic Networks*, pp. 568–575, Sparks, NV, USA, 2009.

[14] Lu Wang, Q. Guo, and J. Liu, "Measuring node importance based on weighted nonlinear method," *Application Research of Computers*, vol. 5, 2018.

[15] T. Schank and D. Wagner, "Finding, counting and listing all triangles in large graphs, an experimental study," *Experimental and Efficient Algorithms*, vol. 3503, pp. 606–609, 2005.

[16] H. Li, J. Zhang, J. Yang, J. Bai, Y. Chu, and L. Zhang, "Social network compression based on the importance of the community nodes," *Acta Scientiarum Naturalium Universitatis Pekinensis*, vol. 49, no. 1, pp. 117–125, 2013.

[17] H. Chou, *Research on Complex Network Visualization Technology Based on Compression and Cluster Analysis*, Jiangsu University, Zhenjiang, China, pp. 23-24, 2017.

[18] R. Xiaolong and L. Linyuan, "Review of ranking nodes in complex networks," *Chinese Science Bulletin*, vol. 59, no. 13, p. 1175, 2014.